David Muñoz Sanchez
Department of Computer Science and Engineering

Report
5 December 2009

T-110.6100 Special Assignment in Datacommunications Software P 5 Cr.

Helsinki University Of Technology.

Passport-PK: Network High Level Source Authentication
Simulation and Design

David Muñoz Sanchez
Department of Computer Science and Engineering

Report
5 December 2009

T-110.6100 Special Assignment in Datacommunications Software P 5 Cr.

Helsinki University Of Technology.

Table of Contents

Table of Figures

David Muñoz Sanchez
Department of Computer Science and Engineering
Report
5 December 2009

T-110.6100 Special Assignment in Datacommunications Software P 5 Cr.

Helsinki University Of Technology.

Terminology:

In this article, we name as Upgraded router to the border routers in each AS which uses Passport-PK.

1. Abstract:

We present the report about the work made by David Muñoz Sanchez in November 2009.

At the beginning, I explored some related literature and documents, such as Segment Layer Authentication[1], Passport[4] and Packet Level Authentication[3]. After a week we changed to Passport-PK[2] before continue working in SLA and try, if it was possible, to get some material related to PLA, which is the one SLA is based in, which will be helpful for our purposes.

Passport-PK, based in Passport, is a system which evaluates the source addresses in the network through a public key based mechanism. One border router of sender's AS signs the packet with its private key and each upgraded router checks the source of the packet through its public key and , if this checking fails, discard the packet. This checking is done only in a percentage of the packets which passes through the router to allow to optimize the system.

The main goal of this article is to show how to improve the performance of this protocol, try to demonstrate in some way that it is effective, show how will be the simulation of this protocol done, which could be the simulator to use and how will it be implemented.

2. Scopes:

Assumption:

We assume every network can prevent internal source spoofing problem through Ingress Filtering[6] or SAVA[7].

Threat model:

Prevent IP spoofing based attacks, specially reflector DDoS[5].

Mitigate on-path and off-path replay attacks.

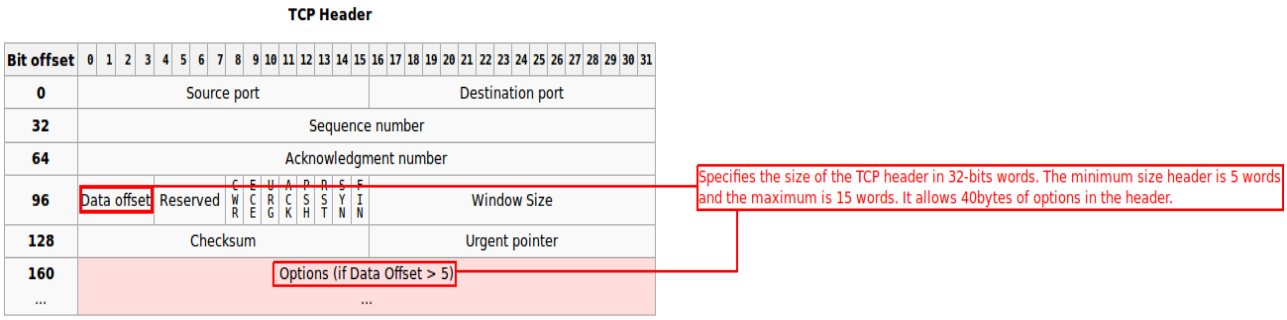

Figure 1: TCP Header
Image from: http://en.wikipedia.org/wiki/Transmission_Control_Protocol (Modified by David Muñoz Sanchez)

3. Passport-PK Designs.

3.1. Passport-PK header position:

Initially the Passport-PK header was though to be located in a TCP option or a shim layer between IP and transport layer. So The first work we did is to analyze these options and tried to get the best one for our purposes.

3.1.1. Analysis of TCP Header Options:

The Options field in the TCP Header could be used for this purpose[8]. The problem in this case is hat the maximum Length of the options header is 40 bytes or 10 words of 32-bits each. In Figure 1 we can see the TCP Header with its fields.

3.1.2. Analysis of IPv6 extension headers.

In our case this one will be the option chosen. IPv6[9] allows to encapsulate another protocol's header inside Ipv6's header and define which kind of extension header it is carrying. In our case the best option would be “Hop by Hop” Option which makes this option to be examined in each hop inside the path. In Figure 2 we can see the IPv6 Header and its fields.

Octet Offset		0								1								2								3							
	Bit Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
C	96																																
10	128																																
14	160																																
18	192	Destination Address																															
1C	224																																
20	256																																
24	288																																

Figure 2: IPv6 Header
Image from: http://en.wikipedia.org/wiki/IPv6 (modified by David Muñoz Sanchez)

3.2. Passport-PK header

It has three fields: Signature(326 bits), Network Prefix (Mask, 8 bits) and a Sequence Number(128 bits, used to prevent replay attacks).

Total length of Passport-PK header: 462 bits. This is the reason we choose IPv6 extension header and not the TCP option one.

3.3. Stamping and verification:

According to data obtained from PLA, one FPGA hardware accelerator could achieve 194,220 verifications per second. On the other hand, one FPGA could achieve 441.560 signatures per second.

4. Probability of checking.

4.1. Initial analysis of the threat.

In networks, every packet pass through a path which has lots of routers. In our case some of this routers could be upgraded and some could be legacy ones.

With this scenario it was useful to calculate a good end-to-end average to detect a spoofed packet. According to the path distribution used in Pi[10], we can calculate, if all routers contained in the path are upgraded, which probability do we need to guarantee that a high percentage of packets would be checked.

4.2. Methodology used to calculate this probability.

In this scenario, we consider the probability to check the packet of all routers a constant "p". And so routers act independently, not influencing the rest, making possible the packet to be checked more than one time through the path or maybe none. So here is the demonstration about how to calculate it:

Oposite event: The probability of one event to happen is equal to 1-Probability of the same event not to happen.
--> **P[E] == 1 - Not(P[E]).**

Intersection probability(independent events with the same probability) = product of probabilities.
--> **1- (1-p)*(1-p)...*(1-p) = 1-(1-p)^n**

p : probability of each router to check one packet

n : number of routers the packet pass through.

So X = 1-(1-p)^n.
X : End-to-End probability of each packet to be checked at least one time through the path.

In this case we want to calculate **'p' to be maximum**:
X= 1-(1-p)^n --> (1-p)^n = 1-X --> 1-p = (1-X)^(1/n)-->

Final formula used: p = 1-(1-X)^(1/n).

4.3. Calculations made according to demanded probability.

Figure 3 shows the results of calculating this probabilities in a graphical way. In Figure 4 we can analyze the data more exactly.

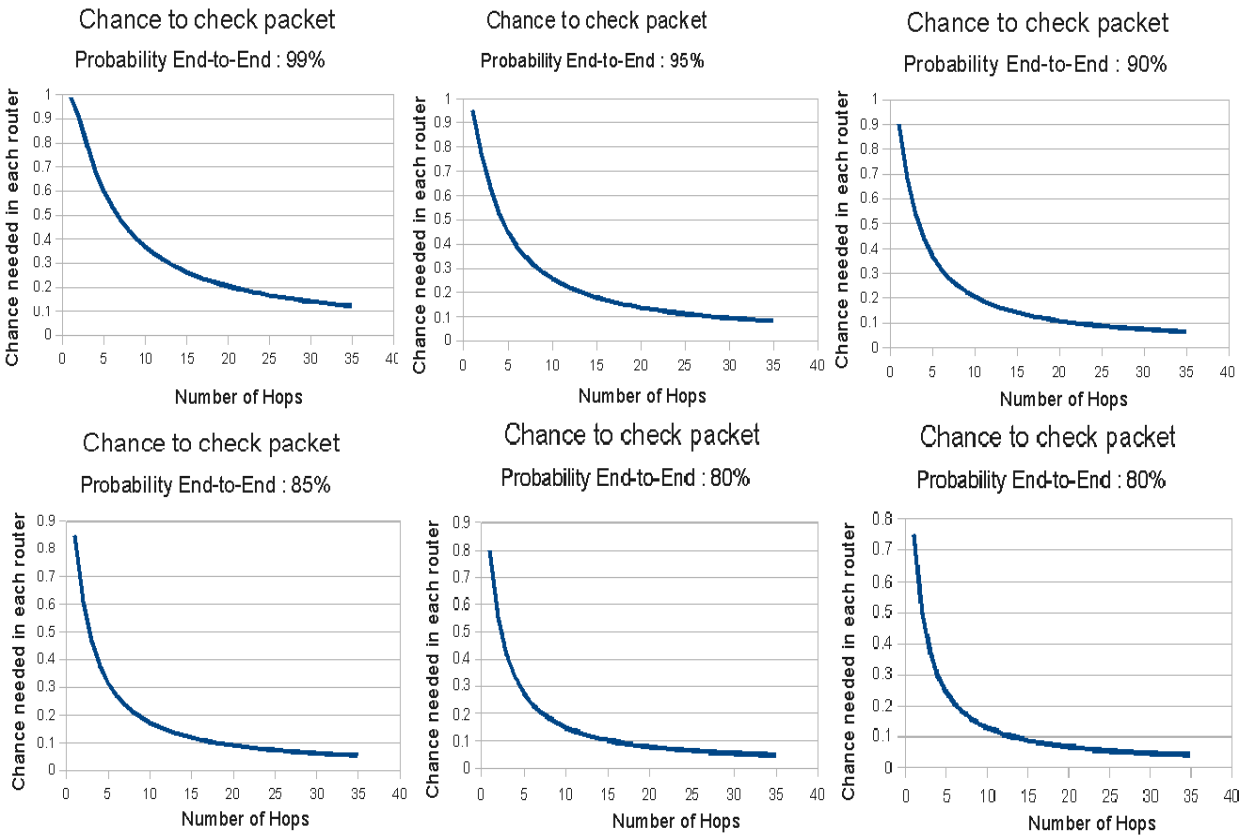

Figure 3: Graphical analysis of needed probability chance.

N_Hops	ProbCalc0.99	ProbCalc0.95	ProbCalc0.90	ProbCalc0.85	ProbCalc0.80	ProbCalc0.75
1	0.99	0.95	0.9	0.85	0.8	0.75
2	0.9	0.78	0.68	0.61	0.55	0.5
3	0.78	0.63	0.54	0.47	0.42	0.37
4	0.68	0.53	0.44	0.38	0.33	0.29
5	0.6	0.45	0.37	0.32	0.28	0.24
6	0.54	0.39	0.32	0.27	0.24	0.21
7	0.48	0.35	0.28	0.24	0.21	0.18
8	0.44	0.31	0.25	0.21	0.18	0.16
9	0.4	0.28	0.23	0.19	0.16	0.14
10	0.37	0.26	0.21	0.17	0.15	0.13
11	0.34	0.24	0.19	0.16	0.14	0.12
12	0.32	0.22	0.17	0.15	0.13	0.11
13	0.3	0.21	0.16	0.14	0.12	0.1

Figure 4: Numerical analysis of needed probability chance.

5. Simulation scenario and possible simulation environments:

5.1. Analysis of the scenario used:

In Figure 5 we can see the scenario used which is similar to the one used in Passport in order to compare the security between each other.

We emulate scenarios in which hosts attacker “attacker”, “attacker1” and “attacker2” which are in legacy ASes , spoof the source address of a victim in an upgraded AS to launch reflector attacks against “victim” through some reflector Hosts “rh1”,...,”rh8”. In this topology we can see legacy routers without upgrading “r1”,...,”r11” and upgraded routers which use Passport-PK “ur1”,…, “ur5”.

In order to compare the performance of Passport-PK and Passport, we also make the hosts rh1 to rh8 each send 100 files to the victim using TCP. These TCP traffic is used to measure how the reflector attacks affect the network performance. The size of each file is 20KB, and a file transfer aborts if it cannot finish in 10 seconds. We vary the attackers’ sending rate from 1% to 20% of the bottleneck link bandwidth and measure the file transfer time.

In order to see how effective is Passport-PK, we want to know how many spoofed packets can be detected if only 30% of AS border routers have been upgraded.

In future simulations we will use random generated topologies.

5.2. Simulation environment: OMNET++

After spending almost one week on analyzing OMNET++, NS2, GNS3 and NS3 we decided to use OMNET++ due to the high deployment which it has, the support given to IPv6 and because of its modular structure for simulating which could make easier to reuse modules programmed by other researchers and so will make future research about topics related to DDoS easier to simulate.

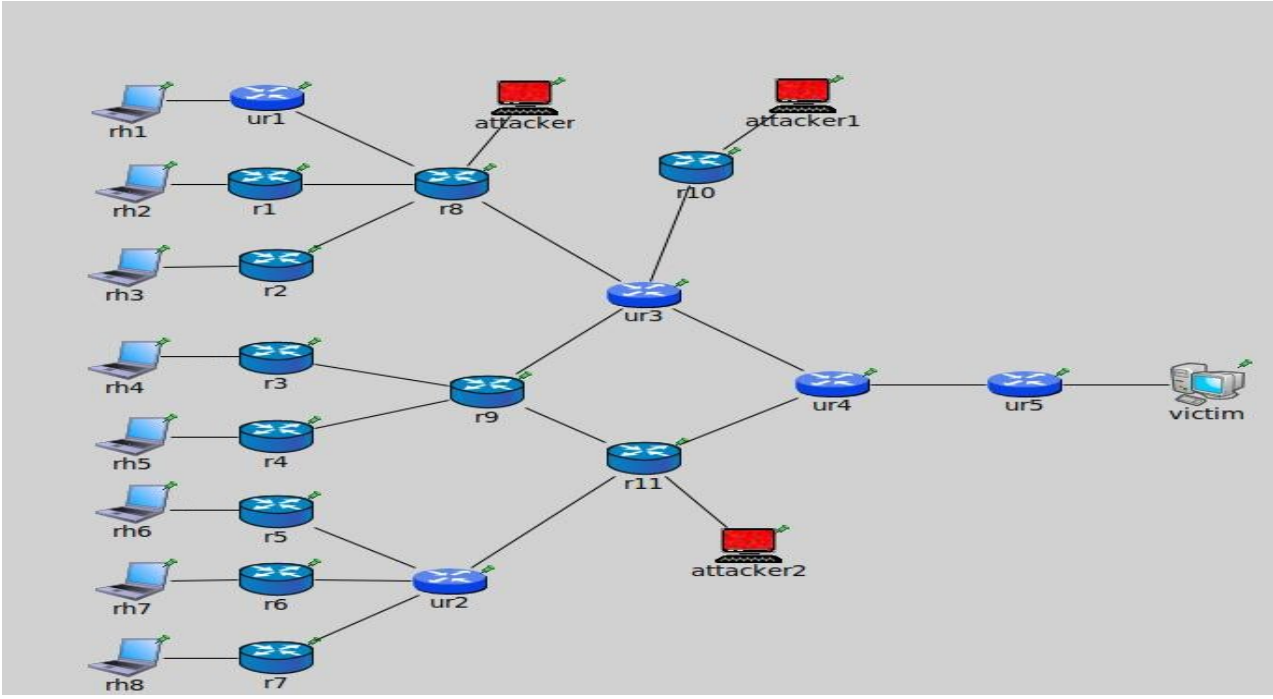

Figure 5: Initial topology used in simulation

5.3. Initial work with OMNET++

- **Module Attacker:** Will be the one sending UDP packets to the reflectorHost which will resend it to the "victim". We need to define a new Module to be used by this host to do this. The next work in this module will be to add a "mark" in the UDP packet(I still have to decide how will it be done) to simulate the signature of the protocol, it should be not very difficult but I preffer to test the network first without special features. Only simulating the DDdoS first and obtaining the results of this test and comparing it with the scenario with our routers. In this case it will be a standardHost6 of omnetpp which will use the module "UDPSpoof" which is defined as follows:
  - Will need a special parameter "srcAddress" defined in omnetpp.ini, UDPSpoof.ned and UDPSpoof.h which will define for the attacker, which address will it spoof.
  - Additionally we will define a new sendToUDP method(inherited from UDPAppBase) which accepts as argument a source Address. This method should be programmed in UDPSpoof.cc too.

- **Module reflectorHost**: Will be the one sending 100 files about 20KB each. This requires to use a special module which exist TCPConnectionApp(Note:Maybe we will need to modify this one to send "N" times the packet). In addition will need to "reflect" "N" times the UDP packets received. To simulate this we defined another module called UDPreflectNPackets which will receive the UDP Packet and answer it with the same packet "N" times, needing a special parameter "nTimes" defined in UDPreflectNPackets.ned, UDPreflectNPackets.h and in omnetpp.ini which will set how many times the packet will be sent(To make it more reusable and define different "N").

- **Module legacyRouters**: Will be the same router as Router6 in omnetpp.

- **Module upgradedRouter**: Initially will be the same router as Router6 given by omnetpp. After this, it will need to define the checking method for the input packets. To simulate it we plan to "wait" a certain amount of time according to which it needs to check one packet(defined in Passport-PK paper).

- **Module victim**: It must have some Sink to accept the TCP Files sent by reflectorHosts and the UDP files received due to the DDos.

- **FlatNetworkConfigurator6**: It is included in omnetpp and it configures the network with some IPs and configure static routetables.

- **Omnetpp.ini**: In it are defined all the relations between modules and parameters of each module.

6. Work being done in this moment.

- Define the timers which will be used to take the final statistics interesting for our work.
- Complete Attacker module to use between 1-20% of its bandwidth
- Complete reflectorHost module to send 100 files.
- When achieved these last objectives, do a initial simulation and take the results back to compare with the results will be taken from the simulation with upgradedRouters.

7. Next step work.

- Define the parameter will be used to simulate the Passport-PK Header(defining its traffic overhead and all other attributes). Maybe it could be useful to add a place in the message where, the FIRST Border router of each AS puts its network prefix(in our case each router should check a bit, if it is set, it has the network prefix, if it is not set, then the router sets it and add its network prefix). In this case for the simulation maybe it can be simplified to assume that "attackers" will not set this bit in the packet before sending and so assuming that this method is secure.
- Define the module upgradedRouter to "check" these packets with some probability.
- Take the results back and compare them with the results obtained without upgraded routers.
- Create a random large topology and compare the results with and without upgraded routers and make a final comparison between results obtained from the random large topology and the initial one.

8. Conclusions.

Taking into account analyzed data and theoretical probabilities calculations we expect to obtain a very high rate of detection of IP spoofed packets taking into account the big amount of packets normally sent and received by the reflector hosts and so, being easier to detect packets without having a very high probability parameter set in each router and so maintaining the performance of the network as similar to the original one as possible.

T-110.6100 Special Assignment in Datacommunications Software P 5 Cr.

Helsinki University Of Technology.

9. References

[1] Ming Li, Matti Siekkinen, Sasu Tarkoma, Antti Yla-Jaaski, “Segment Level Authentication: Combating TCP-based Misbehaving Traffic”.

[2] Ming Li, David Muñoz Sanchez, “Passport-PK : Network High Level Source Authentication”.

[3] Dimitrij Lagutin, “Redesigning Internet – The Packet Level Authentication architecture”.

[4] Xiu Liu, Ang Li, Xiaowei Yang, David Wetherall, “Passport: Secure and Adoptable Source Authentication”.

[5] Vern Paxson, “An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks”.

[6] Network Working Group, RFC2827 , “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoffing”.

[7] Network Working Group, RFC5210 , “A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience”.

[8] Information Sciences Institute University of Southern California, RFC293, “Transmission Control Protocol”.

[9] Network Working Group, RFC2460, “Internet Protocol, Version 6 (IPv6) Specification”.

[10] Abraham Yaar, Adrian Perrig, Dawn Song, “PI: A Path Identification Mechanism to Defend against DdoS Attacks”.