Design, Implementation and Monitoring of the Firewall System for a DNS Server Protection

Filip Hock, Peter Kortiš
University of Zilina, Faculty of electrical engineering, Department of Telecommunications and Multimedia, Zilina, Slovakia
filip.hock@fel.uniza.sk, peter.kortis@fel.uniza.sk

***Abstract*—Today DNS servers run on many different applications and operating systems what means there are many options how to protect DNS server. Each regular application has implemented security mechanisms that protect the system from standard attacks. DNS service works on application layer, however it is possible to prevent many threats already on lower layers. This paper deals about DNS security mechanisms applicable on transport a network layer. The proposed protection technique is based on traffic shaping, flow filtering and prioritization. The presented experiments were performed in subarea of real campus network that is used by students and university staff. Because of implemented security mechanisms the performance of DNS service for internal users has not been affected.**

Keywords—domain; DNS; server; firewall;

I. INTRODUCTION

The firewall systems are generally known as a system for filtering traffic according to an IPv4 or IPv6 address, MAC address or port number. The firewalls are usually used if there is a need to restrict access to a specific part of a network or a host, with the rules based just on the parameters mentioned above. However, firewall systems are more sophisticated and more effective if use specific policy on specific place in a network topology according to protected protocol or network service. The sophisticated approach means that firewall uses more features like traffic shaping, more TCP parameters than only the port number, limitation based on number of connections, logging etc.

Domain Name Systems use both protocols on the transport layer (TCP and UDP). Users send Query messages and get Replies on UDP port 53, however servers between themselves usually use TCP on the same port number, for example during the zone exchange. Because of this, is able to suppose communication on UDP for caching DNS server and mostly use of the TCP protocol for authoritative DNS server. Difference between authoritative and caching DNS server is described in RFC documents [1].

We can divide attacks on DNS into two big categories. First category (Denial of Service attacks) tries to make the service unusable. The second category makes attacks where is not directly victim a DNS server, but serving as an amplifier of the performed attack. What means, the product DNS server turns into an attacker. Both types of attacks are undesirable, but first category is more dangerous. Small sub category of the second category is the use of the DNS server within the man-in-the-middle attack

II. POPULAR ATTACKS ON A DNS SERVICE

A. *Cache poisoning*

Attacker tries to add his own DNS record into the cache of the server, so DNS will point on the fake IP address. Very often used by attackers instead of old phishing methods where the user can see the suspicious URL. Attacker creates fake bank's website and into provider's caching DNS imports record pointing on the IP address of his fake site. Client of the victim bank has no chance to recognize the difference.

In a past, attacks used just a vulnerability of DNS applications, today attacker has to firstly disable authoritative DNS server of victim URL, or least respond before him. When a DNS caching server obtains a query of the domain which it does not have in cache will ask an authoritative server and waits for its reply. One of the method how to avoid this problem is using the DNSSEC [2]. Cache poisoning belongs to the application layer and the firewall is here inefficient.

B. *Amplifying DNS attack*

These attacks are a good example why use the firewall to protect yours DNS servers. Amplifying attacks uses one big disadvantage of a DNS server. If DNS server receives a Query that has couple of bytes, responses are several times bigger. As you can see on a Figure 1, captured by Wireshark, legitimate communication (for type A record) was amplified 4.4 times.

Protocol	Length	Info
DNS	70	Standard query 0x0004 A google.com
DNS	310	Standard query response 0x0004 A googl
DNS	70	Standard query 0x0005 AAAA google.com
DNS	98	Standard query response 0x0005 AAAA go

Figure 1: Sample of a DNS Query and DNS Reply size

Value of the Length shows the size of the Ethernet frame in the byte units. However, these are standard Queries (types A and AAAA) which carry only necessary information for typical user. The problem arises, when the user requests for the information of the whole domain with the record type ANY. On the Figure 2 user requested all records for the domain google.com and get response that contains only few records. The amplification factor in this example only 7.7, because the google servers are perfectly administered and secured.

Badly configured DNS server should amplify data flow 40 - 60 times.

Protocol	Length	Info
DNS	82	Standard query 0x0009 ANY google.com
DNS	629	Standard query response 0x0009 ANY google.com

Figure 2: Sample of DNS Query and Reply of the ANY type

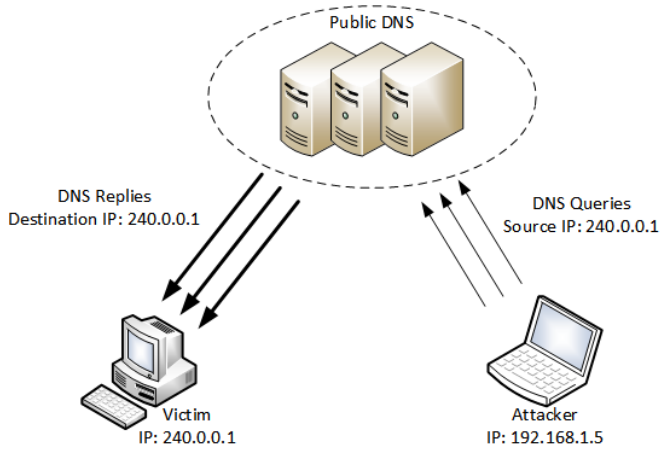

Figure 3: Amplifying DNS attack scheme

C. TCP SYN Flood

Principle of the TCP SYN Flood attack is shown on the Figure 4. It is possible limit number of the connections from the one source by the firewall. However, this protection is effective while the attacker makes only DoS. In a case of the DDoS are the sources different. In this case is possible to implement policies that prioritize home users or lower timer after which server decides close the TCP connection.

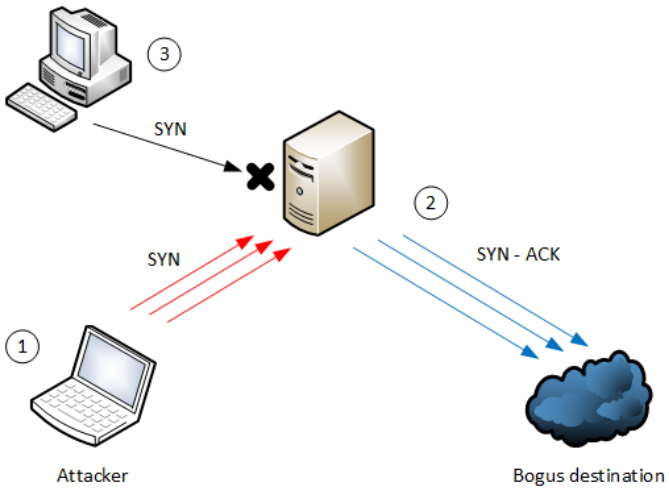

Figure 4: TCP SYN Floods attack scheme

D. DNS Hijacking

Similar attack to Cache poisoning but the attacker changes records on an authoritative DNS server. Usually only option how to perform such a difficult attack is to break into the operating system, which runs the server, and then edit records directly in the service daemon. Attackers usually exploit operating system holes.

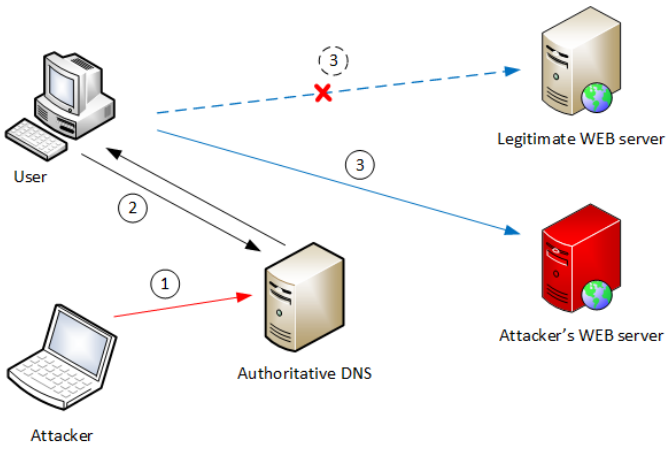

Figure 5: DNS Hijacking attack scheme

After successful edit of the record in DNS server (1), users ask DNS server for the IP address of the WEB server (2). Because the attacker changes the record, DNS server gives the IP address of the Attackers WEB server instead of the legitimate WEB server's IP address. User contacts fake WEB server (3) instead the proper one.

III. Firewall Features

Usually firewall has many additional features; it does not only check the traffic by IP address and TCP or UDP port, or both. This paper shows advantages of traffic shaping and QoS, also described in papers earlier [6, 7].

A. Traffic shaping

One important function is traffic shaping. This feature ensures that traffic which overloads specific bandwidth will be dropped (Figure 6) or buffered (Figure 7).

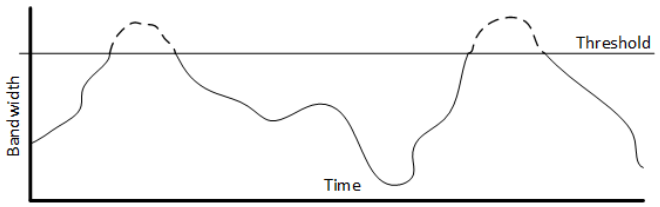

Figure 6: Traffic shaping - dropping overload

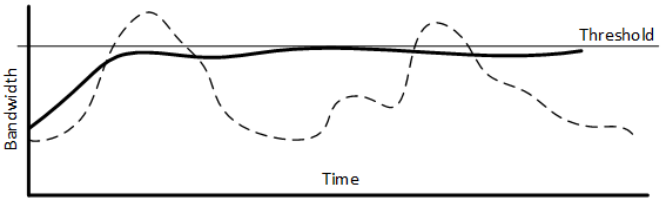

Figure 7: Traffic shaping - buffering overload

As seen on the Figure 6 border, which divides whole bandwidth on allowed band and prohibited area is called threshold. Traffic passing threshold is drawn by dashed line and is dropped.

The principle of real traffic shaping is shown on Figure 7, where original traffic drawn by dashed line is shaped. The shaped traffic is drawn by thick line. Overload traffic is stored in a buffer memory where waits until less traffic flows through the line and then stored traffic is sent.

B. *Quality of Service*

By marking specific traffic with label is possible assign to this traffic priority. The traffic with higher priority is prioritized.

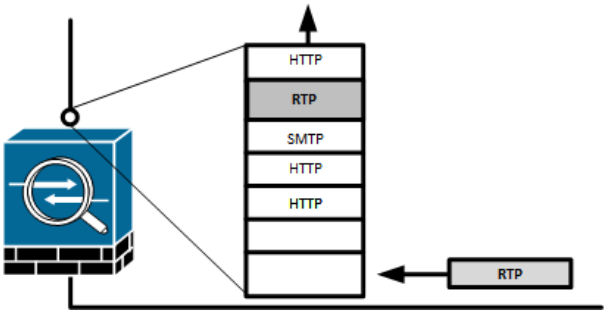

Figure 8: Prioritizing the traffic

The RTP[1] traffic that enters into an egress queue skips other traffic and is processed immediately (Figure 8).

However, it is possible to prioritize any kind of the traffic specified by the parameters of the network and transport layers.

IV. Firewall Design

As mentioned earlier, firewall policy depends on the specific type of the network or service and it is necessary choose proper rules to ensure good and secure traffic flow. Usually creating good policy requires a necessary level of knowledge and some time for testing the group of the specified rules.

A. *Advanced firewall features*

The design of the firewall for DNS servers uses advantages of the traffic shaping with dropping overload and QoS priority. For the line with 1Mbps bandwidth were created three pipes each for specific type of the network. Less important network is whole Internet, than follows LAN where the DNS server is placed and most important is one subnet from the LAN.

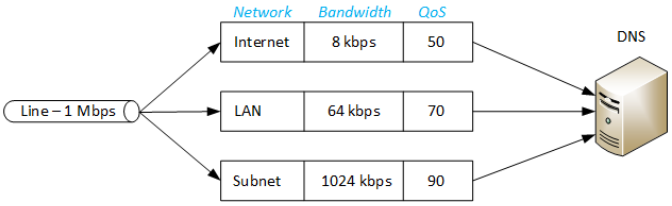

Figure 9: Design scheme for DNS traffic prioritization

Each type of the network has assigned bandwidth for DNS traffic. The traffic from each pipe has assigned tag of priority to ensure which traffic will be processed first.

Firstly, DNS obtains traffic from the subnet, then from the whole LAN and on the end from the Internet. Advantage of this approach is that the DNS attacks from the Internet can never exhaust whole bandwidth and server is always accessible at least from the LAN and subnet for example also for the management purposes.

In this case, model works only for the traffic related to port 53, which is DNS traffic. It is possible to adapt traffic shaping and QoS for specific port range or whole line.

It is possible reserve some bandwidth for large trusted recursive DNS servers, like Google, and ensure that large part of the Internet will have access to the records always. This kind of the approach cannot ensure 100% protection from the DoS and DDoS, but can ensure that there will not be 100% blackout.

B. *Prohibited networks*

Also firewall should have typical elements like deny all private IP described in RFC 1918 [3], deny all packets with the loopback source address, DHCP auto-configuration address or multicasts (class D) and experimental addresses (class E), etc.

C. *TCP relation*

Despite the fact, that DNS servers use primary UDP protocol to provide DNS service, the TCP protocol is used by another application protocols and services. For example, the administration of server is usually performed via SSH protocol. Therefore, it is necessary to avoid the TCP connection table exhaustion. This goal should be achieved by setting the TCP connection timeout to proper (low) value, so the incoming TCP sessions, that are limited by traffic shaping, are not able to exhaust the TCP connection table. The opened, but inactive sessions in TCP connection table expire faster as the new TCP SYN packets are coming.

Another more sophisticated security methods that prevent TCP SYN Flood should be used. A good option how to resist this flood is use of a TCP SYN Cookies method [9].

V. Firewall Configuration

Almost every firewall uses same basic concept in configuration, the differences are only in syntax. In this case is firewall configured under the FreeBSD operating system using IPWF package, which is included directly in kernel of the OS.

```
ipv4="158.193.227.150"
ipv6="2001:4118:300:301:219:dbff:fe5a:2a51"
uniza4="158.193.0.0/16"
uniza6="2001:4118:300::/48"
if1="vr0"
ktam4="158.193.227.0/24, 158.193.227.214/0"
ktam6="2001:4118:300:301::/64"

####  DNS Traffic Shaping rules  ####
ipfw pipe 10 config bw 1024Kbit/s
ipfw pipe 20 config bw 64Kbit/s
ipfw pipe 30 config bw 8Kbit/s
##  Priority specification
ipfw queue 1 config pipe 10 weight 90
ipfw queue 2 config pipe 20 weight 70
ipfw queue 3 config pipe 30 weight 50
```

1. RTP = Real Time Protocol

```
## Rules for DNSv4
$cmd 400 queue 1 ip from $ktam4 to $ipv4 53
$cmd 401 queue 2 ip from $uniza4 to $ipv4 53
$cmd 402 queue 3 log ip from any to $ipv4 53
## Rules for DNSv6
$cmd 403 queue 1 ip6 from $ktam6 to $ipv6 53
$cmd 404 queue 2 ip6 from $uniza6 to $ipv6 53
$cmd 405 queue 3 ip6 from any to $ipv6 53
```

Figure 10: Firewall configuration example

Firstly, three pipes are created, each with specific throughput in kbps. Next are assigned priority values to these pipes. Finally, is each pipe assigned to specific network type, firstly for IPv4 and secondly for IPv6 traffic. Networks are represented with variables.

VI. Traffic Monitoring

It is a good approach to monitor a traffic passing by the firewall or generally traffic aiming to the DNS. If the system administrator is able to monitor actual load in each pipe he will be informed about traffic jam in some of them and he can investigate the reasons.

The traffic shaping design helps to monitor the services remotely by means of SNMP or another protocol even during the DNS DoS attack, because, there is free link capacity for non DNS traffic. Therefore, the administrator is able to connect to the server during the attack and administrate the system remotely.

However, when administrator reserves some bandwidth for a management traffic he can still get detailed information which part or service does not work and still connect to a device with remote access.

If is monitor just traffic flows to the DNS at least is able to have statistical information about average traffic. Sometime is possible to see some traffic bursts as peaks in a graph. This means that high traffic passes in short time, e.g. new DNS record queries arriving. If it is possible observe high traffic for a long time, it will need to consider fact that some attack occurred (Figure 11). With empirical study of traffic is possible to identify normal behavior and unusual behavior. However, it is important realize that every unusual traffic behavior is potential attack which occurs or service failure.

Anyway, it is really necessary to monitor passing traffic load as well as any other parameters, like service status, especially when server runs more than one service. Because the traffic can flow no matter the one service is running or not. Also you can monitor server resources, because some attacks are more sophisticated, which is able to see right on resource graphs and traffic graph shows normal output. For these purposes is possible to use many open-source services (RRDTOOL) as well as a commercial service.

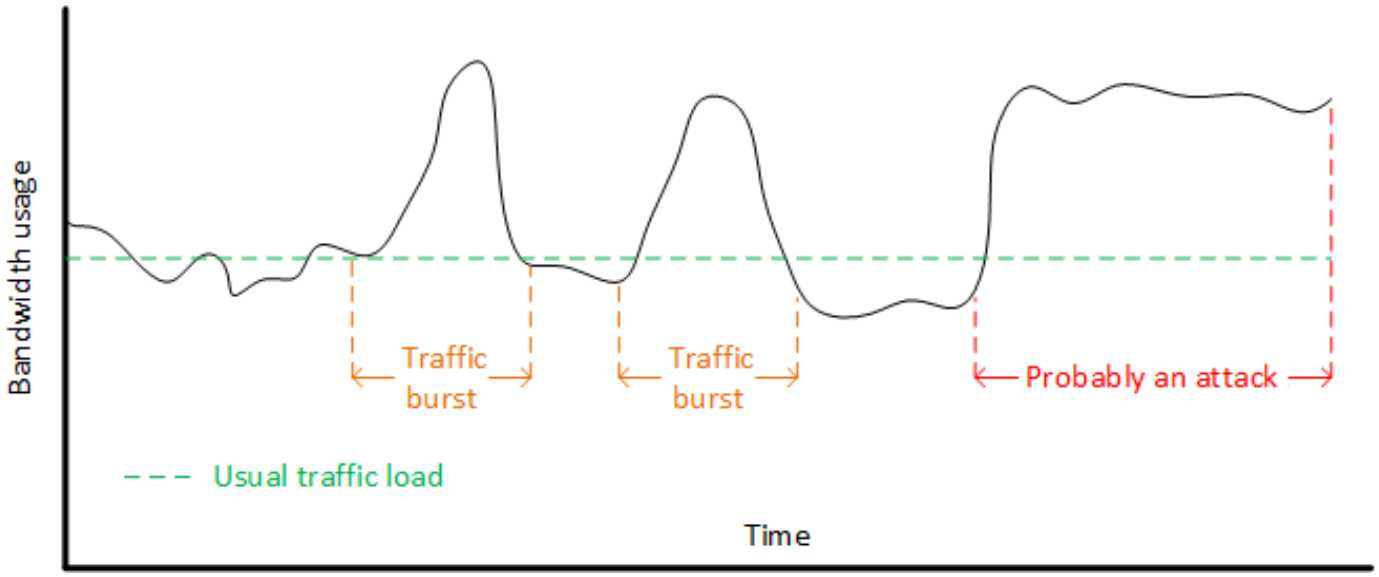

Figure 11: Graph about traffic statistics

VII. Firewall Placement

During the design of the network protected by firewall, it is important to define the appropriate position of the firewall in the network topology and the rules as well. Firstly, is a need to realize if the firewall protects just one host or whole subnet.

One of the good approaches is to put the firewall for the host as close as possible to the host and firewall for the subnet as close as it possible put it on the perimeter of the protected subnet. The collection of filtering rules strongly depends on the firewall placement.

VIII. Firewall Design Testing

For testing purposes were used two production DNS servers each with its own firewall. Master DNS uses traffic shaping and QoS while Slave DNS uses strict firewall rules where the port 53 is accessible only from LAN network. As an attack was used one of the described in the beginning – TCP SYN Flood.

A. *Attacker's Terminal*

Attacker uses common personal computer with Kali Linux [6] operating system installed. This Linux distribution is very popular for penetrating tests. To perform the TCP SYN Floods, attacker uses specially adapted console, called **msfconsole**, what is metasploit console. In the console is executed script for performing attack, **msf > use auxiliary/dos/tcp/synflood**, next was configured parameters as an IP address and TCP port number of the victim. On the picture is shown the attackers console performing the TCP SYN Flood attack (Figure 13).

```
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the int
   NUM                         no        Number of SYNs to s
   RHOST      158.193.227.150  yes       The target address
   RPORT      53               yes       The target port
   SHOST      158.193.214.11   no        The spoofable sour
   SNAPLEN    65535            yes       The number of bytes
   SPORT                       no        The source port (e
   TIMEOUT    500              yes       The number of seco

msf auxiliary(synflood) > exploit

[*] SYN flooding 158.193.227.150:53...
```

Figure 12: Attacker's console output (TCP SYN Flood attack)

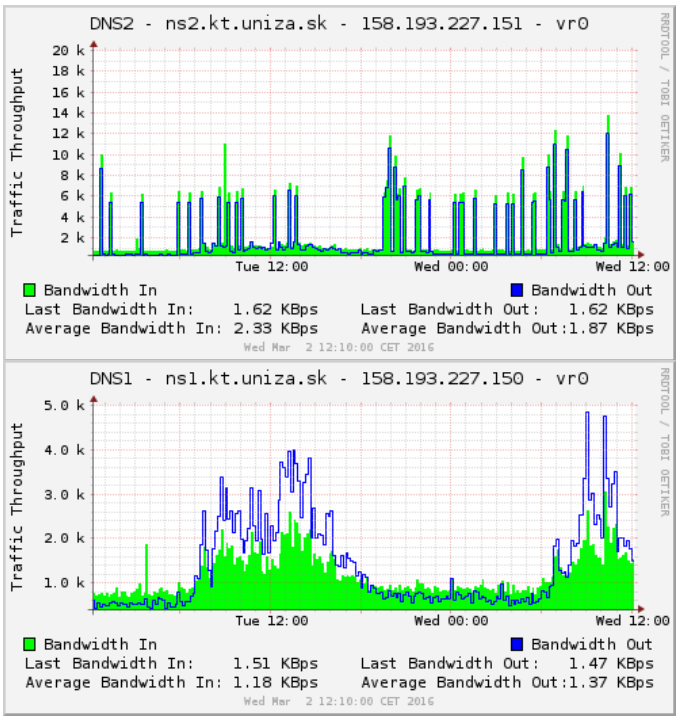

Figure 13: Normal traffic throughput measured on DNS server

a) Master DNS b) Slave DNS

B. *Normal operation*

During the normal operation of monitored DNS severs, was maximal outgoing traffic around 7 kbps and the operational traffic from 5 kbps to 1 kbps in dependence from the part of the day (Figure 13). Monitoring was measured with SNMP, performed by RRDTOOL to graphical output and backup measuring again with SNMP and MRTG.

The upper graph (Figure 13) shows higher output traffic than incoming during the work hours. This event occurs because the DNS replay is much bigger then the query as was explained earlier. Also during the work hours master server gets more DNS traffic than the other one (Figure 13). On the second graph are the peaks because the server runs also the monitoring and graphing service for both.

C. *DNS without the traffic shaping*

When the DNS is under attack from internal network, all functionalities and services are denied. Attack from the outside networks would be denied by the firewall. The consequences of DNS attack are visible as a gap between operational statuses.

The gap in the graph appears, because the RRDTOOL populates the database using SNMP messages encapsulated in the TCP segments (Figure 15). The SNMP messages were not delivered, because no TCP segment delivered to the server during the TCP SYN Flood attack. The attack also exhausted server's resources, so the server did not process any UDP segment carrying DNS query, as well (Figure 14).

```
# dig @158.193.227.151 kt.uniza.sk

; <<>> DiG 9.9.5-12.1-Debian <<>>
@158.193.227.150 kt.uniza.sk
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be
reached
```

Figure 14: Console output - DNS service denied

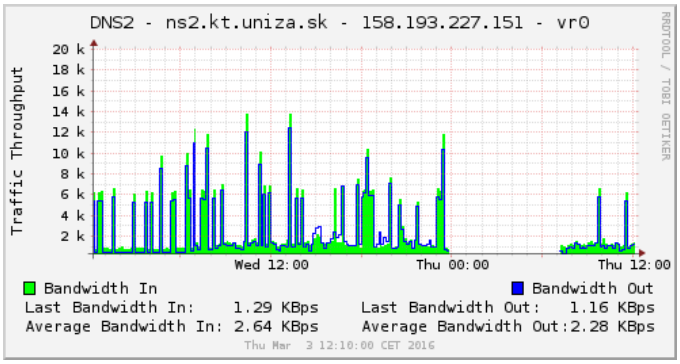

Figure 15: TCP SYN Floods on the Slave DNS

D. *DNS with the traffic shaping*

On the other hand, when traffic shaping with QoS is included in the firewall as was described earlier and the server is under the attack, all services are operational from the zones, which are prioritized. In this case the service is denied from

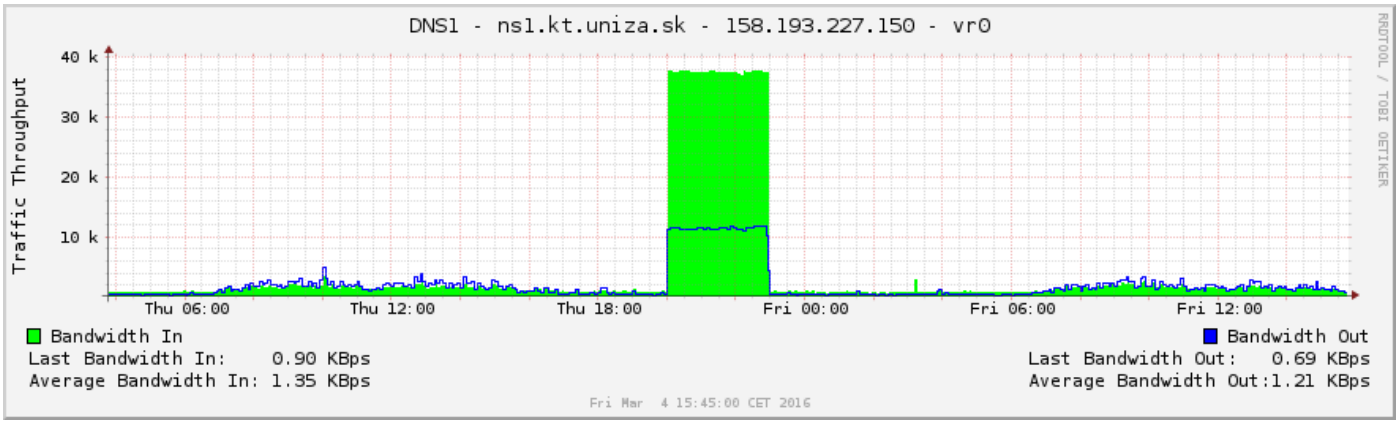

Figure 16: TCP SYN Floods on the Master DNS

the outside of the LAN but as it possible to see on the picture (Figure 16), server is processing at least queries which were not shaped. Incoming traffic is around 38 kBps, some packets were dropped and around 12 kBps was sent out. From outside of the network was response same as on the Figure 14. However, from the inside of the network the queries were served.

```
> server 158.193.227.150
Default Server:  [158.193.227.150]
Address:  158.193.227.150
> kt.uniza.sk
Server:  [158.193.227.150]
Address:  158.193.227.150
Name:    kt.uniza.sk
Address:  158.193.214.233
```

Figure 17: Console output - DNS service allowed

IX. Conclusion

There are many servers protected by firewalls, however rules in the firewalls are very strict or better say – static. We have showed how firewall can dynamically adjust the traffic to ensure at least some packets arrived when is the DNS service under the attack. In the paper, we present the instructions for creation of the rules for traffic shaping and prioritizing. The purpose is to ensure connectivity for the objects of high demand. All presented methods were tested in real environment and their advantage was proved. As was mentioned above, traffic shaping and QoS were explained and described many times earlier, however this paper show advantages of these features on the specific example on the real network environment where the DNS servers are bothered with attacks every day.

References

[1] P. Mockapetris, “Domain Names – Implementation and specification”, IETF: RFC 1035, 1987

[2] Yu Xi, Ch. Xiaochen, X. Fangqin, “Recovering and Protecting against DNS Cache Poisoning Attacks” IEEE, pp. 120-12, 2011 [2011 International Conference of Information Technology, Computer Engineering and Management Sciences].

[3] “Address Allocation for Private Internets”, RFC 1918, https://tools.ietf.org/html/rfc1918

[4] “DNS Best Practices, Network Protections, and Attack Identification”, http://www.cisco.com/c/en/us/about/security-center/dns-best-practices.html

[5] “IPFW Freebsd Man Pages”, https://www.freebsd.org/cgi/man.cgi?ipfw%288%29

[6] Daian Daniel-Simon, Giura Dan-Horia, “Traffic shaping and traffic policing impacts on aggregate traffic behaviour in high speed networks” IEEE, pp. 465-467, 2011 [6th IEE International Symposium on Applied Computational Inteligence and Informatics].

[7] Z. Jiang, I. Joe, “Efficient Bandwidth Utilization and Congestion Control Trough Network Traffic Analysis” IEEE, pp. 280-283.

[8] “Kali Linux”, https://www.kali.org/

[9] Bo Hang, Ruimin Hu, “A Novel SYN Cookie Method for TCP Layer DDoS Attack” IEEE, pp. 445-448, 2009, International Conference on Future BioMedical Engineering.