DNS Amplification Attack Revisited

Marios Anagnostopoulos*, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, Stefanos Gritzalis

Laboratory of Information and Communication Systems Security,
Department of Information and Communication Systems Engineering,
University of the Aegean, Samos GR-83200, Greece

Abstract

It is without doubt that the Domain Name System (DNS) is one of the most decisive elements of the Internet infrastructure; even a slight disruption to the normal operation of a DNS server could cause serious impairment to network services and thus hinder access to network resources. Hence, it is straightforward that DNS nameservers are constantly under the threat of Denial of Service (DoS) attacks. This paper presents a new, stealthy from the attacker's viewpoint, flavor of DNSSEC-powered amplification attack that takes advantage of the vast number of DNS forwarders out there. Specifically, for augmenting the amplification factor, the attacker utilizes only those forwarders that support DNSSEC-related resource records and advertize a large DNS size packet. The main benefits of the presented attack scenario as compared to that of the typical amplification attack are: (a) The revocation of the need of the aggressor to control a botnet, and (b) the elimination of virtually all traces that may be used toward disclosing the attacker's actions, true identity and geographical location. The conducted experiments taking into consideration three countries, namely Greece, Ireland and Portugal demonstrate that with a proper but simple planning and a reasonable amount of resources, a determined perpetrator is able to create a large torrent of bulky DNS packets towards its target. In the context of the present study this is translated to a maximum amplification factor of 44.

Keywords: Amplification attack; Reflection attack; DNS; DNSSEC; DoS; Cybersecurity.

1. Introduction

Denial of Service (DoS) attack is one of the most devastating types of attack, as it aims to disrupt the availability of services in any public or open network like the Internet. This attack becomes even more powerful when comes in the form of a Distributed DoS (DDoS), where many ill-motivated entities collude with the intention of paralyzing the victim. As it is known, DNS Name Servers (NS) provide the mapping of a domain name to its corresponding IP address. Nevertheless, this simple process constitutes the cornerstone of Internet due to the fact that it comes before any other transaction takes place. In this context, DNS amplification attack represents a more perilous

*Corresponding author. *Address*: Info-Sec-Lab Laboratory of Information and Communications Systems Security, Department of Information and Communication Systems Engineering, Samos, GR-83200, Greece. *Phone*:+302273082280 *Fax*:+302273082009

Email address: managn@aegean.gr (Marios Anagnostopoulos)

kind of (D)DoS, since it undermines the normal operation of DNS NS. In fact, the target can be an individual host, such as a DNS NS, or an entire NS infrastructure of a country-specific or generic top-level domain, like org, com, net etc.

A plethora of DNS amplification attack incidents have been reported over time against large corporations, banks, top ranked e-commerce sites, DNS infrastructure servers etc. Very recently, several hacker groups have threatened to blackout the Internet by launching a DNS amplification attack against the root servers [1]. Although for the moment such allegations are proven to be pretentious [2], no one can safely exclude the possibility this kind of attack to be used with the aim to tear down the operation of the NS of a valuable domain name. For instance, in May 2007, US-CERT has received a report that Estonia was experiencing a national DDoS attack [3]. According to the source, the attacks consisted of DNS flooding of Estonia's root level servers. By that time 2,521 unique IPs have been identified as part of the attacking botnets. More recently, in a DNS amplification incident that is characterized as the biggest cyber-attack of this kind in Internet's history, the network infrastructure of Spamhaus was targeted [4]. Spamhaus, which is an organization responsible for blacklisting spam-related sources, sustained for at least a week period a network flood that reached 300 Gbps at its peak. The attack was considered as an act of retaliation on the part of blacklisted operators.

Our Contribution: This paper introduces and assesses a new flavor of DNS amplification attack. Among others, the main advantage of our proposal compared with the standard version of the attack as described in the literature so far is that it does not disclose any illegal or suspicious activity during its execution. Namely, the network flow during the execution of the attack seems to be perfectly legitimate. Moreover, the attack is very hard to be traced back to the perpetrator who, as a result, enjoys the advantage of anonymity. Specifically, the attack scenario is separated into two parts. First off, one needs to perform a degree of reconnaissance to identify the devices in a specific geographical area which operate as (open) DNS forwarders. Second, they need to repeatedly send spoofed queries for DNS Security Extensions (DNSSEC) [5, 6, 7] related Resource Records (RR) to this pool of forwarders. The forwarders consider the victim as the originator of the queries because the (spoofed) source address of the packet contains the victim's IP address. The results we obtained indicate that with proper planning and a relatively fair amount of resources, an attacker is capable of creating a large torrent of bulky DNS answers towards its target. Of course, as shown next, the power of the attack increases proportionally to the number of attacking nodes. It is also to be noted here that although there were some scarce and undocumented concerns that the gradual adoption of DNSSEC (due to its increased RR size) would facilitate aspiring aggressors to mount improved DNS amplification attacks, to authors' best knowledge this is the first study that involves DNSSEC-related RR in a (D)DoS attack. So, although DNSSEC is used among others to drastically confine, if not eliminate, the well-known cache poisoning vulnerability [8, 9], in the course of the current research it will become evident that it may be used as a vehicle for launching large-scale DDoS assaults.

The remainder of the paper is organized as follows. The next section gives basic background information about the DNS protocol and explains the way a standard DNS amplification attack unfolds. The related work is also included in the same part of the paper. Section 3 details on the proposed DNSSEC-driven amplification attack and discusses its impact. Possible ways of repelling and remediating this new type of amplification attack are also included in the same section. The last section draws a conclusion.

2. Background and Related work

DNS is based on the client-server architecture. The server side of the service constitutes a distributed database that utilizes a hierarchical tree (multi-tiered) structure to organize the domain name space into zones. For each zone, there is an authoritative NS responsible to provide answers to the incoming requests regarding the resources of the zone. For this reason, the authoritative NS contains a zone file with the answers in the form of RR. Each RR maps the resources of the zone to its corresponding domain name. Whenever an application running on a given host needs the IP address of a domain (or other related DNS RR), it dispatches a query to a predefined recursive NS. Note that typically the latter entity belongs to the corresponding ISP. In turn, the recursive undertakes to traverse DNS hierarchy and locate the appropriate answer. Moreover, for performance reasons, the recursive NS maintains a cache memory for storing the received RR in order to fulfill subsequent similar requests arriving from the same or another client of the internal network.

A DNS amplification attack utilizes recursive NS as reflector(s) to direct DNS network traffic towards a target. Essentially, an attacker sends spoofed DNS requests to recursive NS. Therefore, the corresponding answers are steered to the target instead of the initiator of the request. The effectiveness of a DNS amplification attack lies on the fact that a small DNS request could trigger a much larger response. The ratio of the size of the response versus the size of the initiating request is called the amplification factor, i.e., *Amplification Factor = size of (response) / size of (request)*. This factor is a direct indication of the impact of the attack; the bigger the amplification factor, the rapider the bandwidth and resource consumption on the victim's side. Especially, with the adaptation of EDNS0, an extension mechanism which allows a NS to create a response with size by far larger than the basic limit of 512 bytes, the amplification factor increases considerably.

For accomplishing a high amplification factor, the attacker usually places a large TXT resource record (about 4 KB) on an NS that controls [10]. This means that either the assailant owns the domain name for which the server is authoritative or they corrupt the zone file in the case they somehow manage to compromise it (Fig. 1: step 1). Moreover, the attacker should control a preferably large botnet. This could be achieved by the distribution of a malware that compromises devices connected to the Internet. After that, the herder of the botnet is able to command the infected computers (bots) to carry out whatever pernicious action they desire.

As depicted in Fig. 1, a typical amplification attack begins by instructing the controlled bots to continually issue DNS requests for one or more maliciously placed large TXT records (Fig. 1: step 2). These requests are directed to (preferably) multiple NS with the capability of serving recursive queries, e.g., open recursive NS or recursive ones that belong to the ISP the bots are connected to (Fig. 1: steps 3-5). Nevertheless, the request is crafted in such a way that its source IP address is that of the victim (e.g., the DNS server of interest). As the DNS protocol is based on UDP, such fabrication is straightforward. Therefore, the recursive NSs which receive the requests, are deceived into directing their responses towards the sufferer (Fig. 1: step 6).

Summarizing, for a typical DNS amplification attack, the perpetrator needs to perform the following actions. First, it is required to place in the DNS hierarchy at least one large resource TXT record, either by compromising an NS or by registering a domain zone. Second, they need to recruit a botnet by distributing a malware. The final step is that of locating a pool of open recursive NS. However, it is conceivable that the first two steps are bound to leave traces that might put the attacker at risk of being detected. In the following section we shall show that the attack under the focus of this paper grants the privilege of almost full anonymity to the attacker.

The first (and only) study that documents DNS amplification by analyzing data stemming from real attacks is given in [10]. This work shows that the analyzed attacks were able to create a flood of 2.8 Gbps in average, while in some cases the ingress traffic could reach an amount of 10 Gbps. The authors make an estimation that the assailants were able to employ a maximum of about 140,000 open recursive NS as their reflector. Typically, to mitigate this threat, the network administrators are advised to disable open recursion [11]. Substantially, they should restrict the services of DNS recursive NS only to the users of the internal network. However, as described in [12], to circumvent this restriction, the attacker utilizes a query for the NS of the root ('.'). The corresponding answer contains the list of the 13 root servers and their IP addresses. Its size is about 500 bytes, which provide an amplification factor of 8. By doing so, NS that support only non-recursive requests are forced to be involved in the attack.

As already mentioned in the previous section, so far in the literature, there exist some worries that DNSSEC could be used as vehicle for launching DNS amplification attacks. The authors in [13] were the first to notice that the increased RR size that DNSEC brings along might augment the amplification factor. However, apart from a single remark, no explanation or description about how this can be done is given. Moving a step forward, the authors in [14] identify this particular shortcoming and argue that DNSSEC-enabled systems would be alluring for aggressors involved in DNS amplification attacks. This is because the increased bandwidth requirements of DNSSEC impose significant upgrades in network infrastructure to sustain the same level of DDoS barricade. To support this standpoint they present some preliminary findings acquired via the use of basic DNS lookups. That is, they recorded the replies produced by DNSSEC-enabled servers to a standard DNS query. Based on the results, they theoretically estimate the amplification factor (43.5 in the best case) and conclude that with DNSSEC, attackers will need substantially fewer DNS amplifiers to perform DDoS. Another interesting point that the authors observed in their experiments is that some servers do not implement DNSSEC by the book, so the responses received may significantly vary in size. This, however, is of considerable value as it significantly affects the amplification factor achieved during an attack. Similar to the previous work, the authors in [15] attempt to measure the computational and bandwidth load generated on a DNSSEC-aware recursive resolver. They notice that most DNSSEC response packets carry a load of $400 \sim 800$ bytes due to the digital signatures contained, whereas a typical DNS response packet has a payload less than 300 bytes. After bombarding their recursive resolver with a query message rate of 1300 packets per second they observed that its response rate drops to 20%. This is a direct indication that DNSSEC resolver is certainly more vulnerable to DDoS attacks.

The work given in [16] analyzes traces from .org in an attempt to estimate the size of the DNSSEC validating resolver population. The authors point out that some of the traffic volumes recorded may also be due to DDoS attacks that used the .org DNSKEY RRset as amplifier. Some other noteworthy work that touches upon the same problem but in an indirect way is given in [17]. Specifically, the authors recognize that their method - used to calculate the Maximum Transmission Unit (MTU) path between a resolver and an NS through a large TXT RR - could possibly be exploited in a DDoS attack. In fact, the described attack scenario does not utilize DNSSEC-related RR (as in our case) but rather a large TXT RR (as in a typical DNS amplification attack) of a DNSSEC-enabled NS.

From the above it can be argued that until now no work in the literature provides an analytical perspective and concrete evidences on DNSSEC-powered amplification attacks. It is also self-evident that none of them has considered DNS forwarders as the major players in the

context of such an attack. Lastly, it has to be stressed that all the previous contributions but [15] have been carried out during the early stages of DNSSEC deployment. This is in contrast to our work where DNSSEC has obtained a certain level of maturity.

For the remediation of DNS amplification attack several security advisories and guidelines have been issued [11, 18, 19]. Still as already pointed out, numerous amplification attacks with slight variations to one another have been unleashed in the recent years against critical network infrastructures. Also, in the literature there exists a critical mass of publications that propose mechanisms aiming to detect this kind of attack at its beginning [20, 21] or to reduce its impact. In this paper though, we opt to neglect focusing extensively on such contributions because they occupy themselves with countermeasures rather than discussing new attack flavors of DNS amplification. Nevertheless, for reasons of completeness in section 3.4 we summarize the most current and accredited methods to cope with such attacks (including the one introduced in the context of this paper).

3. Attack Scenario

3.1. Attack planning and execution

As already mentioned, our attack scenario is divided into two independent phases. First off, a large pool of IP addresses belonging to network devices that operate as (open) DNS forwarders needs to be collected. Recall that a DNS forwarder accepts DNS queries, and after it consults a recursive NS, it returns the appropriate answer to the initiator of the request. Bear in mind that usually DNS forwarders afford cache capabilities as well. Namely, they cache the received RRs with the intention to fulfill subsequent similar requests.

The discovery process of DNS forwarders is akin to that given in [22]. First, for one or more countries, we acquire its block of IP addresses. A straightforward way to do so is to utilize data from www.countryipblocks.net. In our case, as explained later in this section, these countries are Greece, Ireland and Portugal (in alphabetical order). Next, a DNS query is dispatched for a given RR to each IP address in the country-list. Specifically, the requested RR is contained in a domain zone under our administration. The first *label* of the domain name in the question section (QNAME) contained in the request is an indication of the IP address of the device that the packet is headed to. Moreover, the request has the DO flag enabled. This flag designates if the machine being queried supports DNSSEC. Therefore, by doing so, we distinguish which forwarders are able to send back DNSSEC-related RR. On the other hand, with the help of a typical packet sniffer we capture the DNS requests reaching our authoritative server. These requests are trying to resolve the queried RR of the previous step, meaning that they are originated from devices that have the ability of resolving DNS queries recursively.

Comparing the IP address contained in the QNAME with that of the source IP address of the request, one can determine whether the device operates as recursive NS or as forwarder. That is, in the case of a request originated from a recursive NS both IPs are identical. The resolution of RRs that differ in the first label of the QNAME is performed with the help of wildcards following the directions given in RFC 1034 [23]. This situation is exemplified in Fig. 2. Specifically, the client asks every IP in the range from 1.0.0.0 to 239.255.255.255 (excluding reserved IPs like 127.x.x.x) found in the IP blocks of the country of interest, to resolve the corresponding queried domain name. Every device that operates as open DNS recursive NS or open forwarder receiving the query will undertake to resolve it. But to do so this device will send the query to our authoritative server informing us that this IP truly belongs to a machine that acts as a recursive or forwarder.

In the incoming answers the EDNS0 section of the packet is examined. In fact, we select only those answers that have the DO flag enabled (i.e., they support DNSSEC) and simultaneously advertize a large UDP buffer size (e.g. 4096 bytes). As it is known, DNSSEC related RR (RRSIG, DNSKEY, DS, NSEC3) are large in size [24]. As a result, our aim is to filter and keep only forwarders that support DNSSEC and are configured to respond with large payload size. These devices will be used in the latter phase of the attack with the purpose to augment its amplification factor.

After compiling the list of preferable forwarders, one is ready to launch the actual attack. To do so we utilize a network of attacking nodes that repeatedly send DNS requests for DNSSEC-related records toward the forwarders contained in the final list. Though, the source IP address of the packet is being spoofed to contain the IP of the victim, so that all replies are eventually directed toward the target. The requested domain names are related with Top Level Domain (TLD) zones that have adopted DNSSEC. The process of locating the desired zones and afterwards examine the size of the reply is trivial, since the information of which zones have adopted DNSSEC is publicly available. We execute two variations of the attack scenario depending on the destination port the attack flood is delivered. In the first one, this port is 53, while in the second is totally random. Bear in mind that typically all DNS queries are sent from an ephemeral source port ($\geq$49152) to destination port 53, while responses are sent from source port 53 to the same ephemeral, but this time, destination port. The tool used for coding the script that fabricates the DNS packets is Scapy [25]. Algorithm 1 presents a pseudocode version of the script, while Fig. 3 depicts the actual way the attack unfolds.

Algorithm 1 : Scapy pseudocode

```
1:  procedure SendSpoofedPacket(LIST)
2:      READ Forwarders IP Addresses from LIST
3:      while notEndOf(LIST) do
4:          CREATE UDP_PACKET
5:          UDP_PACKET.DestinationAddress ← IPAddress
6:          UDP_PACKET.DestinationPort ← 53
7:          UDP_PACKET.Protocol ← DNS
8:          UDP_PACKET.DNS.RD ← 1 /*Recursion is desired*/
9:          UDP_PACKET.DNS.QR ← QUERY
10:         UDP_PACKET.DNS.QNAME ← "DNSSEC − enabledTLD"
11:         UDP_PACKET.DNS.QTYPE ← ANY
12:         UDP_PACKET.DNS.QCLASS ← IN
13:         UDP_PACKET.EDNS_FLAG ← DO /*Enable DNSSEC*/
14:         UDP_PACKET.DNS.SourceAddress ← 195.251.161.155 /*IP address of target*/
15:         UDP_PACKET.DNS.SourcePort ← 53 /*version 1, port of DNS service*/
16:         //OR for the second variation of the attack
17:         UDP_PACKET.DNS.SourcePort ← Random /*version 2, Not a specific service*/
18:         SEND UDP_PACKET
19:     end while
20: end procedure
```

3.2. Results

For compiling the pool of forwarders, we considered to examine the network IP blocks of three European countries which have more or less the same allocation of IP addresses, but are expected to differ on the level of security awareness. In fact, this assumption is proved to stand true by the results presented further down. Those countries are Greece, Ireland and Portugal. We test the

first phase of the attack (i.e., detection of open forwarders) several times in different days (working days, weekends and holidays) and time zones (working hours and nights), with the purpose to figure out whether their existence is something ephemeral or ordinary. This procedure have been performed twice in a time frame of six months. In the case of Greece, we detected about 60K open forwarders on average per execution. For Portugal the probing process returned about 35K, while for Ireland 10.5K forwarders on average. The exact numbers per country used for implementing the attack are given in the last line of Table 1.

To further investigate the contribution of each forwarder to the effectiveness of the attack, we examine the size of the responses these devices return to the DNSSEC-related query. The results are also summarized in Table 1. As it can be observed, a small but not negligible portion of the forwarders return an answer that exceeds 2,900 bytes. For instance, this number for Greece is 1,094, while for Ireland is much smaller, about 65. In any case, these forwarders are very important to be included in the arsenal of the described attack, as they can present two significant benefits. Firstly, by exploiting them an attacker is able to accomplish an amplification factor of at least 40 (assuming an average size of 70 bytes per DNS request). Secondly, due to its large size, the response is fragmented into three or more IP datagrams. This means that the reassembling (and perhaps reordering and fragment loss) process of packets also conduces to the consumption of resources at the victim side. An attacker is able to integrate this filtering of forwarders in the first phase of their attack as the case may be.

Size of response in bytes	**Amplification Factor**	**Greece**	**Portugal**	**Ireland**
< 1000 (or No response)	< 14	42,569 (69.95%)	25,983 (74.17%)	8,809 (82.35%)
$1000 - 1500$	$14 - 21$	15,112 (24.83%)	6,603 (18.85%)	963 (9.00%)
$1500 - 2000$	$21 - 28$	1,962 (3.22%)	2,205 (6.29%)	802 (7.50%)
$2000 - 2500$	$28 - 35$	80 (0.13%)	26 (0.07%)	42 (0.39%)
$2500 - 2900$	$35 - 41$	41 (0.07%)	9 (0.03%)	16 (0.15%)
≥ 2900	> 41	1,094 (1.80%)	204 (0.58%)	65 (0.61%)
	Total forwarders:	60,858 (100%)	35,030 (100%)	10,697 (100%)

Table 1: Percentages of open forwarders per country in regards to the size of response they return

As already mentioned, the amplification factor is the most crucial element for an attack to be effective. With the purpose to better estimate its magnitude in the context of the attack described in this paper, we initially run only one instance of the script for both attack variations and counted how many responses arrived at the victim and what their size was. More specifically, the script dispatched a single DNS query towards the 1,363 forwarders given in the last but one line of Table 1. As already pointed out, each DNS query packet created by scapy has a size of 70 bytes. Considering the first variation of the attack, the total number of packets arrived at the target machine reached 3,110 packets having a total size of 3,526,046 bytes. For the second variation, we recorded 3,539 packets with a total size of 4,187,901 bytes. Therefore, it can be safely argued that the amplification factor for the first version of the attack is almost 37, whereas for the second is nearly 44.

From the above result, it can be observed that for every query we dispatch, the target receives 3 packets that are reassembled in one DNS response sized about 3,100 bytes. Furthermore, the volume of the incoming packets is a little smaller in the first variation of the attack, which means in the case the destination port of the responses is not 53 but rather random, we are able to accomplish

a slightly bigger amplification factor. This difference is anticipated as many firewalls are configured to block egress DNS queries originated from sources other than their internal network recursive NS.

For the needs of the real attack we utilized 22 typical Personal Computers (PC) each one connected to a 100 Mbps network interface. Our tests demonstrated that every attacking node is capable of sending nearly 880 DNS queries per second on average, or 61.6 KBps. This is the case when a device runs 25 instances of the attack script simultaneously. In the ideal case, this means that each attacking node is capable of flooding the target with 880 DNS responses per second. However, this data volume is multiplied by the amplification factor, that is, 37 and 44 for each attack variation respectively. Consequently, a single attacking node unleashes on average a stream of 2.28 and 2.71 MBps respectively towards the victim. In order to investigate the accumulative impact of each node that joins the attacking group, we progressively triggered the scripts on each one of them.

On the other side, the target, acting as DNS authoritative NS, was a desktop machine equipped with a Dual 2.8 GHz CPU and 4 GB RAM connected to 100 Mbps network interface. This machine had the DNSSEC extensions enabled. Table 2 summarizes the progress of the attack for both of its variations. Regarding the first variation, besides the inbound traffic, we also recorded the CPU overhead caused by the *bind* process, i.e. how much the incoming unsolicited DNS packets affect the performance of the victim as authoritative server.

Figure 4 depicts the level of resource consumption at the victim-side during both attack variations. More precisely, figure 4(a) shows the incoming traffic in MBps for both variations, while 4(b) introduces the CPU consumption due to *bind* process (port 53). Time 1050 signals the moment that the initial scripts are beginning to terminate. However, the effects linger to recede because, as explained further down, the use of more than 12 attacking nodes inflict the same impact on this particular target.

As we can easily deduce from Table 2, with a small number of attacking nodes we were able to exhaust the network bandwidth of the victim's machine. To put it another way, for both attack variations, it is apparent that only a dozen of nodes are capable to flood a 100 Mbps network. More importantly, as it can be observed from Fig. 4(a) the addition of new attacking nodes to the testbed does not achieve cumulative effects on the target. In fact, in the last stage of each scenario, the network was so overflowed, that a great amount of fragmented packets was dropped and the ratio of the crafted requests versus the ingress corresponding responses is fallen below 1:1. Actually, due to fragmentation, this ratio is anticipated to be 1:3, which for the initial phases of the attack scenario stands true. Comparing the two attack variations, we can put forward that for the second one the volume of the flow has augmented by 45% on average with relation to the first variation. This is obvious when comparing the values contained in the third and sixth column of table 2. As a matter of fact, this applies to the first three phases of the attack, because in the latter phases the network was so monopolized leading a great amount of packets to be discarded. However, for the first variation of the attack, in addition to the intense increase in the volume of incoming traffic, the impact on performance is also significant. This is because when a DNS response arrives at (standard) port 53 it is being processed by *bind*, thus consuming resources on the victim's machine. In contrast to this, when a DNS packet arrives at a random port (second variation of the attack) it is simply dropped. Therefore, in the case the victim is an authoritative or recursive NS, the most effective way of performing the attack is to use the standard destination port.

During attack escalation, we queried a group of 75 open recursive NS for unique RR contained in the victim's domain zone. Hence, these queries need to be resolved by the victim. This would give us a clear estimation on how the sufferer - being under continuous flooding - will behave when trying to serve legitimate DNS requests. Thus, we record the average time a request needs to be fulfilled, as well as the ratio of the abortive queries. That is how many queries are lost and return a "connection timed out, no servers could be reached" error. Clearly, the average query time increases proportional to the volume of flooding traffic the victim undergoes. It is remarkable to point out that at the latest stages of the attack, where 550 instances of the script were active, almost the half of the queries were lost.

	Scenario 1: port = 53				**Scenario 2: port = Random**		
Number of attacking nodes	**CPU utilization (*bind*)**	**Inbound traffic**	**Average query time**	**Loss Packet Ratio**	**Inbound traffic**	**Average query time**	**Loss Packet Ratio**
None	0%	0.5 KBps	119 msec	0%	78 KBps	113 msec	0%
2 (50 scripts)	5.65%	1.48 MBps	125 msec	0%	2.16 MBps	118 msec	0%
4 (100 scripts)	10.8%	2.91 MBps	128 msec	0%	4.44 MBps	128 msec	0%
8 (200 scripts)	20.1%	5.67 MBps	137 msec	0%	8.06 MBps	149 msec	0%
12 (300 scripts)	25.78%	8.47 MBps	156 msec	0%	11.28 MBps	157 msec	40%
22 (550 scripts)	30.94%	12.05 MBps	241 msec	41%	12.50 MBps	243 msec	42%

Table 2: Effects on target proportional to the power and number of attacking nodes per scenario

Each individual stage of the amplification attack, corresponding to the gradual activation of new attacking nodes as described in Table 2, lasts for approximately 2 minutes, while altogether the various stages have a duration of about 27 minutes. During this time, the attacking nodes via the use of the scripts dispatched about 14,993,000 DNS requests. If we consider that in average our DNS request has a size of 70 bytes, in overall the attacking nodes dispatched approximately 0.97 GB of network flow toward the 1,363 DNS forwarders contained in the joined list obtained for all three countries. However, regarding the first variation of the attack, the victim was flooded with 10,507,706 packets of roughly 12.8 GB volume, while in the latter the victim suffered 10,999,752 packets of nearly 14.2 GB. Amongst other reasons, it is evident that we have a loss of roughly the half of the volume due to the excessive traffic. This is also verified from the fact that in the last step of the attack 42% of the DNS queries were timed out.

3.3. Discussion

The first phase of the attack described above, sadly exhibits that a worryingly large number of DNS forwarders operate in the open Internet and do serve DNS requests originating from sources outside their network. Further analysis of the hardware and Operating System (OS) of these devices reveals that their majority are Small Office Home Office (SOHO) network devices, such as network printers, ADSL routers, NAT (Network Address Translation) devices etc. OS fingerprint of the forwarders with the XPROBE2 tool indicated that 75% of the forwarders in Greece, 45% in Ireland and 55% in Portugal have HP JetDirect, Foundry Networks IronWare OS or Cisco IOS as their OS. In any case, these devices erroneously or due to misconfiguration operate as DNS proxies. Moreover, WHOIS analysis attests that most forwarders belong to Autonomous System of ISP networks. Actually, the possibility of malicious exploitation of a device in order to turn

it into an open forwarder is highlighted in a very recent Common Vulnerabilities and Exposures (CVE) report issued by NIST [26]. This report introduces the corresponding vulnerability for DNS related library, but it is quite possible this security weakness to stand true for other software as well.

As we can observe from Table 2, at the time that a dozen attacking devices were running, the victim had to cope with an ingress traffic of nearly 10 MBps that overwhelmed the victim's network bandwidth capacity. Even though this rate cannot be considered as an (D)DoS attack for a real target connected to a Gbps network interface, we believe that with the proper scaling of the attacking network, a determined attacker is able to achieve a very large volume of DNS flow. For example, think of a case where a large group of ill-motivated persons scattered around the world start simultaneously running several instances of the attack script. If we consider that a simple PC, as those used in our experiments, is capable of dispatching a maximum of 880 DNS queries on average, ideally this machine could contribute a 2,5 MBps stream toward the target. Of course this rate is the upper limit according to the testbed used, but with proper equipment and a larger pool of forwarders (more countries under consideration) the attacker could easily exceed these limits and paralyze the victim. Besides, the amplification factor of the attack - which is independent of the number of attacking nodes, and as already mentioned, is in the order of 37 to 44 - is self-evident about its effectiveness.

As already pointed out, the advantage of our proposal compared to that of the standard amplification attack is the elimination of all traces that can be used toward disclosing the attacker's actions. Essentially, there is no need to infect any machine with malware in order to turn it into a bot. For assembling a botnet, the aggressor only needs to recruit in a transparent to them manner the available forwarders existing out there. This is also strongly in favor of the attack as the usage of the forwarders conceals the attack's real source. Thinking of a large number of forwarders participating in the attack, the only reaction left to the victim is to block the inbound traffic arriving from numerous sources. This for example could be done by instructing the firewall to ban traffic originating from certain but constantly changing IPs. Nevertheless, in practice this could be proved quite hard to achieve. Also, the attacker does not have to penetrate into an authoritative NS in order to place a large RR in the DNS hierarchy. Instead, to intensify the amplification factor of the attack, one simply has to exploit the existence of large DNSSEC-related RRs.

Moreover, the recursive NSs that provide the large responses to the forwarders do not possess any record of the attacker's actions. They only observe legitimate queries coming from devices residing in their internal network. Provided that some of the forwarders may have caching capabilities (this is the usual case), the forwarders do not even consult continuously the corresponding recursive NS, but only for the very first query. Finally, with the described attack scenario one becomes able to bypass the countermeasures against amplification attacks taken by the majority of recursive NS. This is because the attack does not involve any recursive NS but only the forwarders, meaning that any countermeasure deployed on the recursive-side is not applicable in our case. Since, all recursives are queried not directly, but through devices located in their inner network, we involve them in the attack like they operate as open recursive NS (see Fig. 3). To put it in other way, these recursive NSs usually do not function as open, but by following the aforementioned strategy we force them to act like they are. Moreover, our attack is also immuned to the recently proposed DNS Response Rate Limiting (DNS RRL) [27], which is integrated in the functionality of the authoritative NS (more details of DNS RRL are given in the next section).

One can argue that the first phase of our proposal, i.e., the discovery of the forwarders, is

very noisy (due to its large volume of packets it produces) and easily detectable. However, this is partially true. During the experiments of the first phase, we sent legitimate DNS queries to all the IP address allocated for all of the countries. We executed this process twice during a six month period. However, as already explained in section 3.2, each time we carried out the discovery process for seven different occasions to include different days and time zones. This means, that we sent almost 160M DNS packets towards IP addresses of these countries. However, the detection of such legitimate but unusual traffic is entirely up to the administrator of the corresponding network domain. From our experience in the experiments described above, only one network administrator noticed our bizarre queries and notified our abuse list. Also, this information gathering process may be quite more chronologically separated from the actual attack, meaning that the exact time the attack will be unleashed and its target is entirely up to the aggressor.

As previously mentioned, to the best of our knowledge, the only work in the literature that analyzes data stemming from real DNS amplification attacks is given in [10]. In this work the authors report a maximum amplification factor of nearly 60 but they exhibit a DNS query size of 60 bytes. It should be noted that for achieving the aggressors placed a large TXT RR in the DNS hierarchy that had a fixed size of 4000 bytes. On the other hand, according to the scenario at hand the response's size depends on the DNSSEC related resource records that the forwarders opt to include in the packet. That is, the attacker does not need to place a large RR, but rather to exploit a DNSSEC enabled zone.

3.4. Countermeasures

Actually, DNS amplification attack is not impossible to preclude. If any (or preferably all) of the following measures have been put in place then the described scenario would be quite difficult to get fruitful results.

- *Source validation*: As this kind of attack mandates the spoofing of the source IP address of the packet, any IP address validation would block malicious packets. Of course, it is not possible for every firewall or router to examine the source IP of all UDP packets passing through it. Though, the devices located at the borders of a network should inspect and allow a packet to pass through only if it has a source IP address assigned from an internal subnetwork. This guideline is explicitly outlined in RFC 2827 [28].

- *Disable open recursion:* Any recursive NS should only accept DNS queries from clients residing inside its network. Yet, as it can be observed in the results obtained, an attacker is able to evade such restriction with the exploitation of the forwarders. For this reason, the administrators must disable DNS forwarding to all network devices. Whenever, the installation of a forwarder is a requisite, its service should be restricted to solely trusted or internal users.

- *Detection of DNS amplification:* Network administrators should adopt solutions such as those described in [20, 21], which aim to detect and suppress DNS amplification attacks at their very beginning. Further information on this type of solutions are given below in this section.

- *DNS Response Rate Limiting (DNS RRL):* This up-to-date solution limits the identical responses that can be returned to the same requestor within a time interval. Therefore, it hinders a potential aggressor to entangle an NS into their amplification attack. Mostly, RRL is applicable to authoritative NS, as usually the normal DNS flow to an authoritative

contains limited duplicate queries from the same source. This is due to caching facility every recursive affords. Note that DNS RRL is already implemented in BIND 9 [27].

As already pointed out, so far, most of the research work in this area concentrates on designing methods to detect and suppress DNS amplification attacks at their very beginning. So, for reasons of completeness, it is necessary here to shortly present the most important ones of them. The authors in [20] proposed a mechanism that could be integrated into the functionality of a DNS NS. This solution capitalizes on the matching of DNS requests and corresponding responses of the NS. Therefore, any response reaching the server that does not correspond to a request (the server solicited) is inevitably characterized as suspicious. When the ratio of the unsolicited responses exceeds a predefined threshold, then an alert is generated and banning rules are automatically set/updated in the firewall in order to block traffic stemming from the attacking nodes. The authors in [21] extended the aforementioned work [20] by incorporating bloom filters in an effort to speed up the process of detection. Further, bloom filters have been recruited by the authors in [29] to deal with DNS amplification. Nevertheless, this time the solution proposed was based on hardware, aiming to efficiency.

A different approach has been followed by the authors in [30] and [31]. Specifically, they made use of an Intrusion Detection System (IDS) capable of detecting DNS amplification with the help of Neural Networks (NN) and Support Vector Machines (SVM) Machine Learning Classifiers.

Finally, in [32] the authors introduced a probabilistic model based on Continuous Time Markov Chain model to conduct a cost-benefit analysis for DNS amplification countermeasures. In their work the three countermeasures under consideration were: 1) filtering and blocking the attack sources, 2) random drops of DNS (UDP) packets as described in [33] with the purpose to regulate the incoming traffic, and 3) aggressive retries from the clients for increasing the legitimate traffic [34]. According to the authors, this probabilistic model was able to deduce significant reductions in the DNS amplification attack probabilities when the aforementioned countermeasures are deployed. Also, their model indicated that the usage of DNSSEC is more vulnerable than that of DNS, and thus, DNSSEC gains noticeable fewer benefits from the proposed countermeasures.

Based on the results gathered from the experiments described in section 3.2, we can easily deduce that a significant number of the inspected network devices do not support any of the aforementioned countermeasures. In the aftermath of the results obtained, we can safely argue that poor practices and omissions from the side of network administrator and ISP companies may put the Internet at risk. Putting it another way, usually, for reason of cutting down cost and wastage, network providers decide not to confirm with security advisories which would greatly hinder the feasibility of amplification attacks.

4. Conclusions

Without doubt, DNS constitutes the backbone service of the Internet. However, from a security point of view, DNS is also a very tempting target for attackers who are in a constant effort to use it as a reflector for launching (D)DoS. This work introduces and raises awareness on a new stealthy flavor of DNSSEC-powered amplification attack that exploits the vast number of misconfigured forwarders living out there. By using as a case study three different countries, we argue that these forwarders might be easily recruited by attackers to launch large-scale DDoS assaults. Specifically, we showed that with the help of legacy PC equipment and a certain degree of reconnaissance one is able to achieve an amplification factor that fluctuates between 37 and 44 depending on the

destination port the attack flood is headed to. It is also showed that the attacker traces at the victim's side will indicate the forwarders as the source of the attack. This means that the true perpetrator will leave no exploitable evidence of its actions, so practically the attack is very difficult to be traced back to its original source. This is in contrary to all other incidents of the standard amplification attacks taken place so far, which specifically employ botnets to execute the assault. However, the creation and management of a botnet is bound to leave traces.

A side contribution of this article is to underline the necessity of immediate action to be taken on the network administrator and Internet provider side in deterring such serious attack incidents from happening.

References

[1] Anonymous, Operation global blackout, http://pastebin.com/NKbnh8q8 (February 2012).

[2] R. D. Graham, No, #Anonymous can't DDoS the root DNS servers, http://erratasec.blogspot.gr/2012/02/no-anonymous-cant-ddos-root-dns-servers.html (February 2012).

[3] G. Evron, Battling botnets and online mobs, Georgetown Journal of International Affairs, 9 (2008) 121–126.

[4] R. Lemos, Largest-Ever DDoS Campaign Demonstrates Danger of New Attack Method, http://www.eweek.com/security/largest-ever-ddos-campaign-demonstrates-danger-of-new-attack-method/ (March 2013).

[5] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4033: DNS security introduction and requirements, http://www.ietf.org/rfc/rfc4033.txt (2005).

[6] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4034: Resource records for the DNS security extensions, http://www.ietf.org/rfc/rfc4034.txt (2005).

[7] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4035: Protocol modifications for the DNS security extensions, http://www.ietf.org/rfc/rfc4035.txt (2005).

[8] D. Dagon, M. Antonakakis, K. Day, X. Luo, C. P. Lee, W. Lee, Recursive DNS architectures and vulnerability implications, in: Proceedings of 16th Network and Distributed System Security Symposium (NDSS), 2009.

[9] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, J. Bellmor, A centralized monitoring infrastructure for improving dns security, in: 13th International Conference on Recent Advances in Intrusion Detection, Springer, 2010, pp. 18–37.

[10] R. Vaughn, G. Evron, DNS amplification attacks (Preliminary Release), http://www.isotf.org/news/DNS-Amplification-Attacks.pdf (2006).

[11] US-CERT, The Continuing Denial of Service Threat Posed by DNS Recursion (v2.0), Tech. rep., US-CERT, http://www.us-cert.gov/reading_room/DNS-recursion033006.pdf (2013).

[12] D. Jackson, DNS Amplification Variation Used in Recent DDoS Attacks, http://www.secureworks.com/cyber-threat-intelligence/threats/dns-amplification (February 2009).

[13] A. Singh, B. Singh, H. Joseph, Vulnerability Analysis for DNS and DHCP, in: A. Singh, B. Singh, H. Joseph (Eds.), Vulnerability Analysis and Defense for the Internet, Vol. 37 of Advances in Information Security, Springer US, 2008, pp. 111–124.

[14] A. Cowperthwaite, A. Somayaji, The futility of DNSSEC, in: Proceedings of 5th Annual Symposium Information Assurance (ASIA'10), 2010, pp. 2–8.

[15] Y. Yao, L. He, G. Xiong, Security and Cost Analyses of DNSSEC Protocol, in: Y. Yuan, X. Wu, Y. Lu (Eds.), Trustworthy Computing and Services, Vol. 320 of Communications in Computer and Information Science, Springer Berlin Heidelberg, 2013, pp. 429–435.

[16] O. Gudmundsson, S. Crocker, Observing DNSSEC validation in the wild, in: Securing and Trusting Internet Names (SATIN), 2011.

[17] K. Rikitake, K. Nakao, S. Shimojo, H. Nogawa, UDP Large-Payload Capability Detection for DNSSEC, IEICE TRANSACTIONS on Information and Systems E91-D (5) (2008) 1261–1273.

[18] ICANN Security and Stability Advisory Committee (SSAC), SSAC Advisory SAC008 DNS Distributed Denial of Service (DDoS) Attacks, Tech. rep., ICANN (2006).

[19] R. Chandramouli, S. Rose, Secure Domain Name System (DNS) Deployment Guide, Recommendations of the National Institute of Standards and Technology (NIST), http://csrc.nist.gov/publications/nistpubs/800-81r1/sp-800-81r1.pdf (April 2010).

[20] G. Kambourakis, T. Moschos, D. Geneiatakis, S. Gritzalis, Detecting DNS amplification attacks, in: Critical Information Infrastructures Security, Springer, 2008, pp. 185–196.
[21] S. Di Paola, D. Lombardo, Protecting against DNS reflection attacks with Bloom filters, in: Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2011, pp. 1–16.
[22] D. Dagon, N. Provos, C. P. Lee, W. Lee, Corrupted DNS resolution paths: The rise of a malicious resolution authority, in: Proceedings of Network and Distributed Security Symposium (NDSS08), 2008.
[23] P. Mockapetris, RFC 1034: Domain names-concepts and facilities, http://www.ietf.org/rfc/rfc1034.txt (1987).
[24] M. Anagnostopoulos, G. Kambourakis, E. Konstantinou, S. Gritzalis, DNSSEC vs. DNSCurve: A Side-by-Side Comparison, IGI Global, 2012, p. 201.
[25] Scapy project, http://www.secdev.org/projects/scapy/.
[26] NIST, Vulnerability Summary for CVE-2012-3411, http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-3411 (March 2013).
[27] V. S. Paul Vixie, DNS Response Rate Limiting (DNS RRL), http://ss.vix.com/~vixie/isc-tn-2012-1.txt (June 2012).
[28] P. Ferguson, D. Senie, RFC-2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, http://www.ietf.org/rfc/rfc2827.txt (2000).
[29] C. Sun, B. Liu, L. Shi, Efficient and low-cost hardware defense against DNS amplification attacks, in: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE, IEEE, 2008, pp. 1–5.
[30] S. Rastegari, M. I. Saripan, M. F. A. Rasid, Detection of Denial of Service Attacks against Domain Name System Using Neural Networks, International Journal of Computer Science Issues 6 (2009) 23–27.
[31] S. Rastegari, M. I. Saripan, M. F. A. Rasid, Detection of Denial of Service Attacks against Domain Name System Using Machine Learning Classifiers, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2010 (2010) 444–447.
[32] T. Deshpande, P. Katsaros, S. Basagiannis, S. A. Smolka, Formal analysis of the DNS Bandwidth Amplification Attack and its countermeasures using probabilistic model checking, in: High-Assurance Systems Engineering (HASE), 2011 IEEE 13th International Symposium on, IEEE, 2011, pp. 360–367.
[33] A. Mankin, Random drop congestion control, in: Proceedings of the ACM symposium on Communications architectures & protocols, Vol. 20, ACM, 1990, pp. 1–7.
[34] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, S. Shenker, DDoS defense by offense, in: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2006, pp. 303–314.

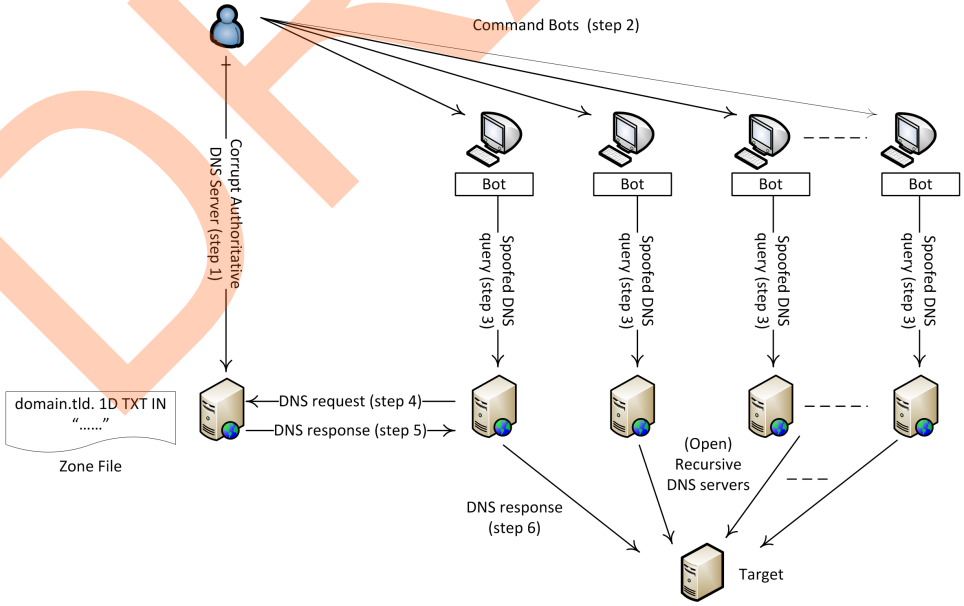


Figure 1: High-level architecture of a typical DNS amplification attack

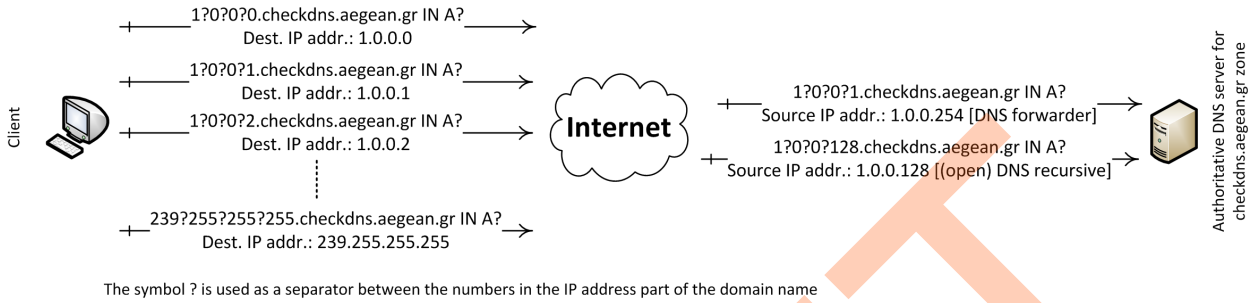


Figure 2: DNS forwarders discovery process

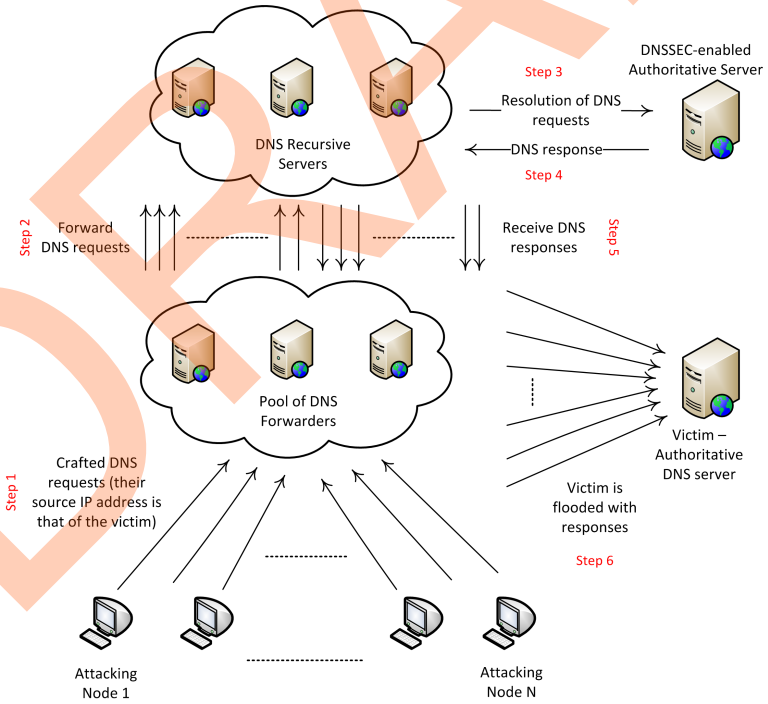
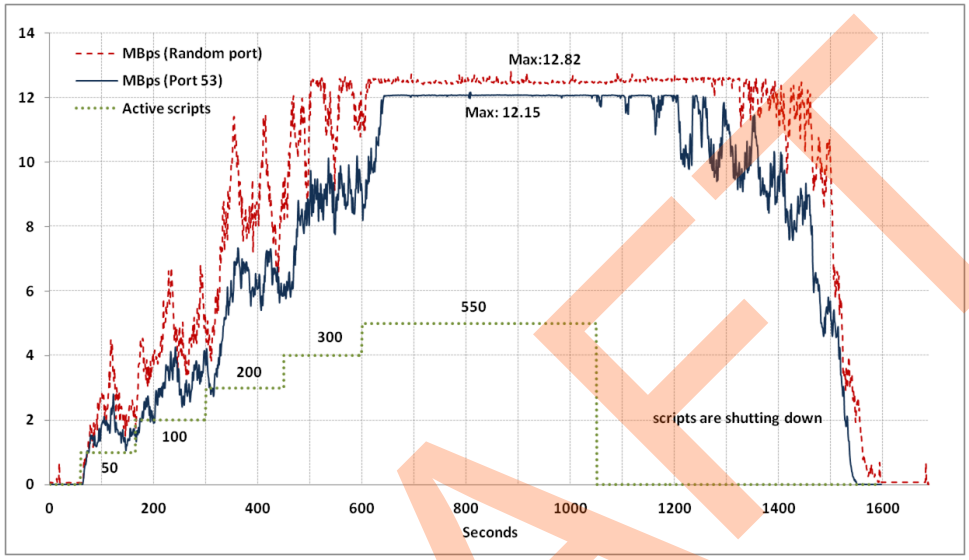
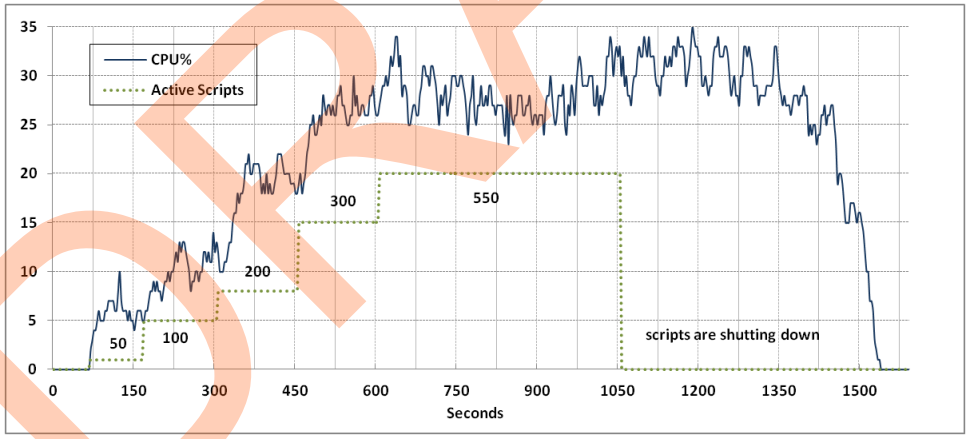


Figure 3: High-level architecture of the attack introduced in this work



(a) Incoming traffic in MBps for both attack variations



(b) CPU consumption (port 53)

Figure 4: Progress of Resource Consumption at the victim-side