A Fair Solution to DNS Amplification Attacks

Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis and Stefanos Gritzalis

Laboratory of Information and Communication Systems Security
Department of Information and Communication Systems Engineering
University of the Aegean, Karlovassi, GR-83200 Samos, Greece
Tel: +30-22730-82247
Fax: +30-22730-82009

Email:{gkamb, tmos, dgen, sgritz}@aegean.gr

Abstract

Recent serious security incidents reported several attackers employing IP spoofing to massively exploit recursive name servers to amplify DDoS attacks against numerous networks. DNS amplification attack scenarios utilize DNS servers mainly for performing bandwidth consumption DoS attacks. This kind of attack takes advantage of the fact that DNS response messages may be substantially larger than DNS query messages. In this paper we present a novel, simple and practical scheme that enable administrators to distinguish between genuine and falsified DNS replies. The proposed scheme, acts proactively by monitoring in real time DNS traffic and alerting security supervisors when necessary. It also acts reactively in co-operation with the firewalls by automatically updating rules to ban bogus packets. Our analysis and the corresponding experimental results show that the proposed scheme offers an effective solution, when the specific attack unfolds.

I Introduction

Internet architecture and consequently the World Wide Web (WWW), at least at their early stages, have been designed without taken into serious consideration any security issues. Attackers try to exploit this fact in order to achieve an unauthorized access or to cause a Denial of Service (DoS) in the provided services. By the term DoS we mean any malicious attempt aiming to render the provided service and / or communication unavailable to legitimate users.

More specifically, DoS attacks could take two major forms. In the first one, the adversary featly crafts packets trying to exploit vulnerabilities in the implemented software (service or a protocol) at the victim's side. This category of attacks includes outbreaks like the ping of death [1]. In the second one, the aggressor attempts to overwhelm critical system's resources affecting the provided service, like memory, CPU, network bandwidth by creating numerous of well-formed but bogus requests. This type of attack is also well known as flooding. Various incidents in the Internet have been already reported in the literature [2]-[5] as flooding attacks targeting either on the provided service or on the underlying network infrastructure. The most severe among them is presented in [2] and is known as Reflection Distributed DoS (RDDoS). Such an attack can rapidly paralyze a targeted network realm, costing both money and productivity.

Furthermore, most recent attack incidents verify the catastrophic outcomes of this category of attacks. For instance, as reported in [2], in October 2002 eight out of the thirteen root DNS servers were suffered a massive DoS attack. Many other similar attacks were triggered against

Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007)
1-0/07 $25.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

DNS in 2003 and 2004 [13], [14]. In a recent study, the DDoS activity in the Internet was projected employing a method called “backscatter” [15]. The results of this study manifested that nearly 4,000 DDoS attacks are released each week. In February 2006, name servers hosting Top Level Domain (TLD) zones were the frequent victims of enormous heavy traffic loads.

In contrast to normal DDoS attacks, where an arsenal of bots launches an assault on a single targeted server, the new attacks unfold sending queries to DNS servers with the return address aiming at the victim. This situation is more difficult to repel because in this case the DNS server performs the direct attack. For instance, in an ordinary DDoS attack, one can potentially block a bot instructed to launch a DDoS attack by effectively blocking the bot’s IP address. On the contrary, it is not so simple to block a DNS server without affecting and damaging the operation of a corporate network. The amplification factor in such recursive DNS attacks stems from the fact that tiny DNS queries can generate much larger UDP responses. Thus, while a query message is approximately 24 bytes (excluding UDP header) a response message could easily triple that size. Generally, this outbreak takes advantage the fact that the DNS is needed by any service (http, ftp etc) requires name resolution.

In this paper we focus on DNS amplification attacks suggesting a novel and effective solution to eliminate its consequences. Our repelling mechanism implements an effective and practical solution that can proactively alarm administrators before the attack affect network resources. This is also achieved and bolstered up by reactively blocking attackers’ IP addresses at the firewall. The rest of the paper is organized as follows. Section II presents basic background information regarding DNS. Section III focuses on DoS flooding attacks in DNS, while Section IV presents the existing countermeasures and remedies proposed so far. Moreover in this section we introduce our approach capable of reactively detecting and repelling DNS amplification attacks. Finally, Section V concludes the paper giving also some pointers for future work.

II The Domain Name System

The DNS is a hierarchical distributed system providing the necessary mapping or binding between human comprehensible domain names and the corresponding numerical IP addresses. This mapping procedure is also known as address resolution service. In the root of this hierarchy tree is located the mapping of top level domains, like “.gr”, “.com”, “.org” etc, to the IP addresses of the corresponding authoritative DNS server. Each of these domains and the subsequent sub-domains form a specific zone. The leaf of each zone in this hierarchy stores the mapping of a specific domain name to its IP address; this information is kept in the corresponding DNS Resource Record (RR). An example of this hierarchy is depicted in Figure 1.

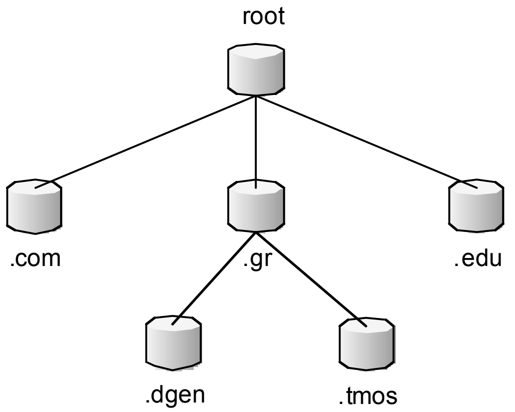

Figure 1. DNS hierarchical distributed architecture

Figure 2 illustrates the interaction among a DNS Server and a resolver from the one side and the client components from the other. More specifically, consider the case in which a client tries to connect to “www.tmos.gr”. At first, the client generates the appropriate query for www.tmos.gr and passes it to the local resolver. The resolver contacts the DNS cache server. If the DNS cache server has the requested mapping available, it responds with the requested RRs, otherwise inquires recursively the root DNS and the corresponding authoritative DNS for the IP address of .gr, tmos.gr accordingly. This procedure continues until the cache server receives the actual RR of www.tmos.gr. As soon as the DNS cache server receives the corresponding mapping stores it in its cache and forwards it back to the resolver, which in turn passes it to the client. More details about DNS can be found in [7]-[9].

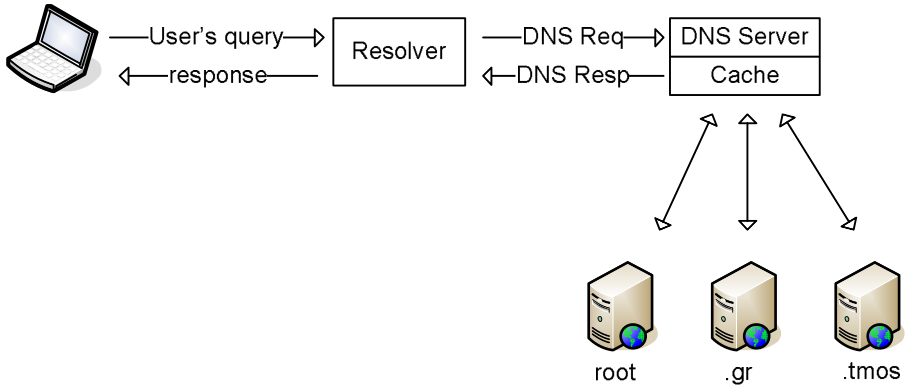

Figure 2. DNS Resolution Name Procedure

III Flooding Attacks and The Domain Name System

As already mentioned, the main goal of any flooding attack is the consumption of critical system resources in order to paralyze the provided services and make them unavailable to its legitimate users. Assuming that such an attack takes place against or exploits a critical component like the DNS it is very likely that would quickly incapacitate the overall network’s services making it unavailable to any legitimate user. Several researchers have pointed out the threat of flooding attacks using recursive DNS name servers open to the world. For instance, according to a recent study [17], which is based on case studies of several attacked ISPs reported to have on a volume of 2.8 Gbps, one event indicated attacks reaching as high as 10Gbps and used as many as 140,000 exploited name servers.

Flooding attacks against DNS are similar to other well documented Internet services flooding attacks and could be launched in two distinct ways. In the first one the attacker sends a large number of bogus DNS requests either from a single or multiple sources, depending on the flooding architecture utilized [4], [5]. An example of multiple sources flooding architecture attack against a DNS is depicted in Figure 3. According to this scenario, the attacker orchestrates usually innocent hosts, called zombies, to simultaneously generate fake DNS requests aiming at disrupting the normal DNS operation by consuming its resources; mainly memory and CPU.

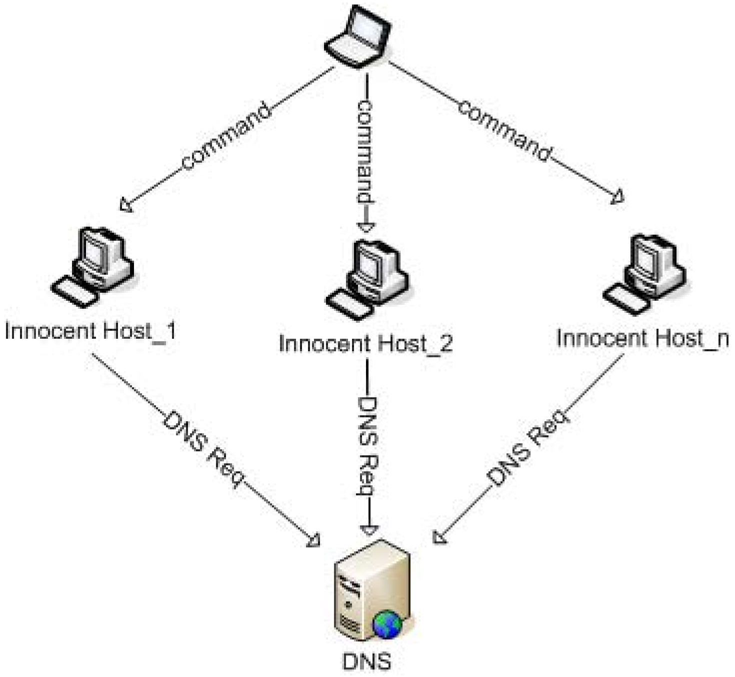

Figure 3. Multiple sources flooding attack architecture

On the other hand, the most sophisticated and "modern" attacks exploit the DNS components themselves in an attempt to magnify flooding attack consequences. Putting it another way, in a DNS amplification attack scenario, the attacker exploits the fact that small size requests could generate larger responses. The relation between a request and the corresponding response is known as the amplification factor and is computed by the following formula:

Amplification Factor = size of (response) / size of (request)

The bigger the amplification factor is, the quicker the bandwidth and resource consumption at the victim is inflicted. Consequently, in the case of DNS amplification attack the aggressor is based on the fact that a single DNS request (small data length) could generate very larger responses (bigger data length). For example in [8] the DNS response was restricted up to 512 bytes length, while in [9] even bigger. The attack unfolds as follows: The attacker falsifies the source address field in the IP datagram to be that of a host on the victims' network. Using the spoofed address, a DNS query for a valid resource record is crafted and sent to an intermediate name server. The latter entity is usually an open recursive DNS server, which follows the resolving procedure presented in Section II and forwards the final response towards the target machine as illustrated in Figure 4. The attacker will repeatedly send the query to the intermediate name server but with all the responses going to the victim network. Potentially, the adversary could consume the entire bandwidth of a T1 line by generating a few thousand responses.

Supposing that the attacker employs a distributed architecture similar to that presented in Figure 3, it is obvious that the bandwidth and resources consumption rate at the victim increase very rapidly. Furthermore, it should be noted that the attacker featly spoofs all query requests to include a specific type of DNS resource in order the authoritative DNS server to generate large responses. This task could be managed either by discovering which DNS servers store RRs that when requested create large responses or by compromising a DNS server and deliberately include a specific record – also known as the amplification record - that will create a large response. An example of this technique, exploiting large TXT records which is introduced in EDNS [9]. After that, the attacker collects a list of open recursive name servers that will recursively query for, and then return the amplification record he/she created. Even a list of known name servers may be more than adequate. As stated in [17] there is a 75% chance that any known name server is an open resolver too, thus a copy of a TLD zone file may be sufficient. A detailed description of DNS amplification attacks is presented in [6].

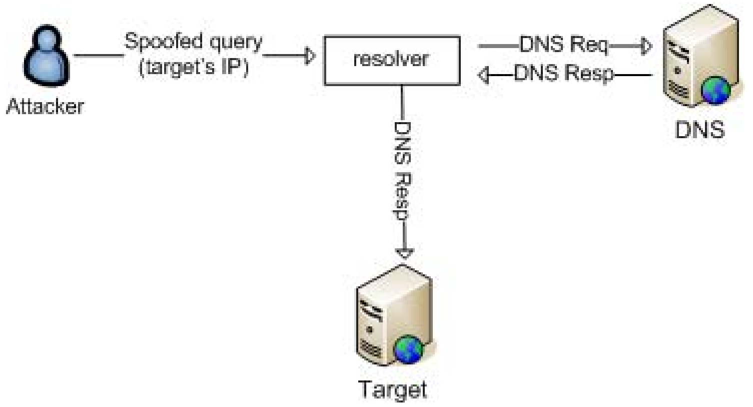

Figure 4. General Architecture of a DNS

IV Protection Mechanisms Against DNS Amplification Attacks

In order to shield against DNS DoS attacks different protection layers must be deployed. In this section we present known countermeasures and our practical and novel solution to defend against amplification attacks. Having these mechanisms acting simultaneously, it is very possible to build a more secure, redundant and robust DNS infrastructure and shield our network against this category of attacks.

A. General Countermeasures and Remedies

DNS employs UDP to transport requests and responses. As a result, the malicious user is able to fabricate the appropriate spoofed DNS requests very easily. Thus, as a first level of protection it should be introduced a spoof detection / prevention mechanism like the ones proposed in [10]-[13]. Moreover, to mitigate DNS cache poisoning and Man-In-The-Middle (MITM) attacks, which usually launched at the early stages of a DNS amplification attack, additional security mechanisms should be employed. These are necessary in order to ensure the integrity and origin authentication of the DNS data that reside either in RR cache or in the zone file [10],[14].

Apart from well accepted practices to securely configure DNS servers [19], another effective remediation, at least against outsiders, is to disable open recursion on name servers from external sources and only accepting recursive DNS originating from trusted sources. This tactic substantially diminishes the amplification vector [18]. Until now, available data reveal that the majority of DNS servers operate as open recursive servers. The Measurement Factory [17] reports that more than 75% of domain name servers of approximately 1.3 million sampled permit recursive name service to arbitrary querying sources. This leaves abandoned name servers to both cache poisoning and denial of service attacks.

B. Limitations of Countermeasures & Remedies

Although the generic countermeasures and remedies referred in previous section could decrease the chances of potential attackers to launch a flooding attack are not able to provide an effective solution against DNS amplification attacks. More specifically, it is well known that these mechanisms are employed only by a limited number of DNS servers, therefore many DNS are unprotected or misconfigured, which in turn are exploited by malicious users in order to amplify the hazardous effects of flooding attacks as described previously. Moreover, solutions like DNSSEC [10] do not offer an efficient countermeasure against flooding attacks as already argued in [15]. In addition, these mechanisms do not provide any security against (malevolent) insiders, who it is well known that are responsible for many security incidents. On the top of that, the traffic generated in a DNS amplification attack seems to be normal, so the prevention of such an attack could not be achieved only with the employment of the security mechanisms presented in Section IV.A. Therefore, the introduction of a specific detection / prevention mechanism against DNS amplification attacks should be considered mandatory.

To the best of our knowledge until now the only method that specifically addresses DNS amplification attacks is the DNS-Guard one [20]. This approach involves several policies that generate some form of cookies for a DNS server to implement origin authentication; that is to verify whether each incoming request is indeed from where the request datagram says it is from. However, the main problem with DNS-Guard is that it introduces large traffic and delay overhead and mandates wide scale deployment.

C. The Proposed Approach

The proposed solution is based on the one-to-one mapping of DNS requests and responses. Specifically, under DNS normal operation when a client requests a name resolution sends a request towards the appropriate DNS, which is responsible to create the corresponding response (see Section II). Nevertheless, when a DNS amplification attack takes place, the targeted DNS server receives responses without having previously sent out the corresponding request. As a result, such data (orphan pairs) must be immediately classified as suspicious and discarded.

Based on the aforementioned simple but fruitful idea, we employ a monitor to record both DNS requests and responses using the IPtraf tool [16]. At the same time, our custom-made PHP based tool, namely DNS Amplification Attacks Detector (DAAD) , process on-the-fly the captured data, which are stored in the appropriate database (see Table 1 & 2). Thereby, the incoming DNS traffic are characterized as suspicious or not and generate the corresponding alert in the case of an undergoing attack. Note, for example, that the second line of Table 2 (response) matches with the first line of Table 1 (request). The architecture employed by the proposed scheme is depicted in Figure 5, while the overall DAAD's detection logic is presented in Figure 6. The interface of the DAAD tool is publicly accessible at: http://f6tmos.samos.aegean.gr/~tmos (username: user & password: kalimera!).

In a nutshell, when a DNS message is received the DAAD engine determines whether the message is a response or a request. For any received {request, response} pair the DAAD tool creates a new entry to the request / response table (see Tables 1 & 2 accordingly). When a message is identified as a response the DAAD module checks for the existence of the corresponding request. If the response does not match with none of the requests logged previously in a given time frame is marked as malicious. Additionally, as soon as the number of malicious messages exhibits a given administrator-specified threshold an alert is generated and all firewall rules are automatically updated to block the attacker's data as depicted in Figure 5. All the parameters in the aforementioned procedure i.e. time frame, threshold, can be dynamically updated and depend on the administrator's security policies in the specific network domain.

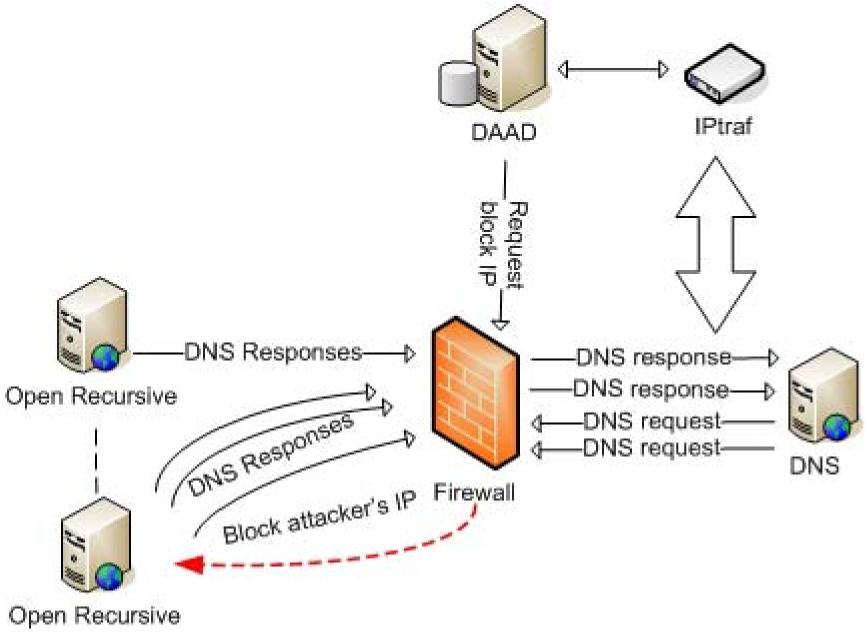

Figure 5. The proposed DNS Amplification Detection Architecture

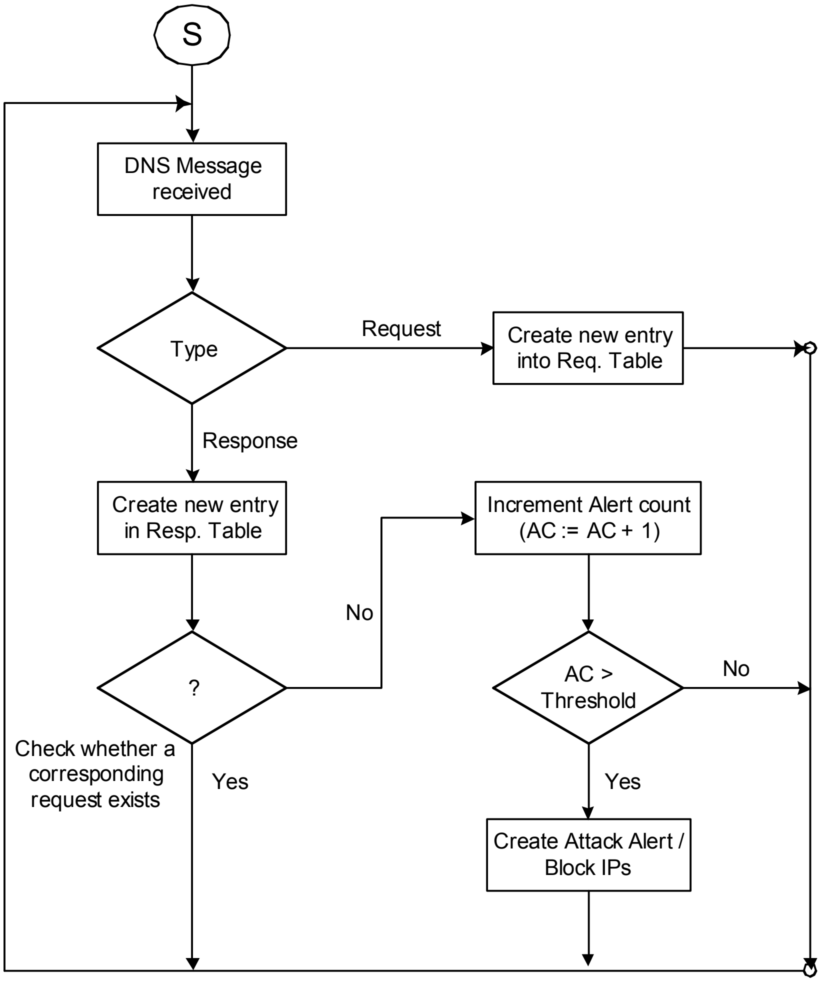
S
DNS Message received
Type
Request
Create new entry into Req. Table
Response
Create new entry in Resp. Table
Increment Alert count (AC := AC + 1)
No
?
Check whether a corresponding request exists
Yes
AC > Threshold
No
Yes
Create Attack Alert / Block IPs

Figure 6. DAAD's engine detection logic

In order to evaluate the accuracy of the proposed approach we have employed the architecture presented in Figure 5 and performed a self-attack. According to the attack scenario, the aggressor generates spoofed DNS requests and sends it towards the local DNS server, trying to cause a DoS (see Section III). The victim can be either the DNS server itself or another machine residing inside the target network. So, the IP of the requests are properly spoofed to contain the victim's IP. The plots in Figure 7 illustrate the relation of the DNS requests (data_out) and responses (data_in), whereas in case of mismatched responses the corresponding data are characterized as malicious (attacks). As already mentioned, when the number of attack data exhibits a given threshold, the DAAD tool creates a request, which is delivered to the firewall to block the attacker. An example of such a rule instructing the firewall to ban the IP address 192.168.1.1 is presented below.

iptables -I RH-Firewall-1-INPUT -p udp -s 192.168.1.1 -m state --state NEW -m udp --sport 53 -j REJECT

Table 1. An Example of the DNS requests Table

Source IP	Source Port	Destination IP	Destination Port
195.251.162.96	32790	195.251.128.5	53
195.251.162.96	32790	194.177.210.210	53
195.251.162.96	32790	194.177.210.210	53
195.251.162.96	32790	195.251.177.9	53
195.251.162.96	32790	192.33.4.12	53
195.251.162.96	32790	192.5.6.32	53
195.251.162.96	32790	192.12.94.32	53
195.251.162.96	32790	192.12.94.32	53

Table 2. An Example of the DNS responses Table

Source IP	SourcePort	Destination IP	Destination Port	Status
194.177.210.210	53	195.251.162.96	32790	OK
195.251.128.5	53	195.251.162.96	32790	OK
195.251.177.9	53	195.251.162.96	32790	OK
192.33.4.12	53	195.251.162.96	32790	OK
192.5.6.32	53	195.251.162.96	32790	OK
192.12.94.32	53	195.251.162.96	32790	OK
192.12.94.32	53	195.251.162.96	32790	OK

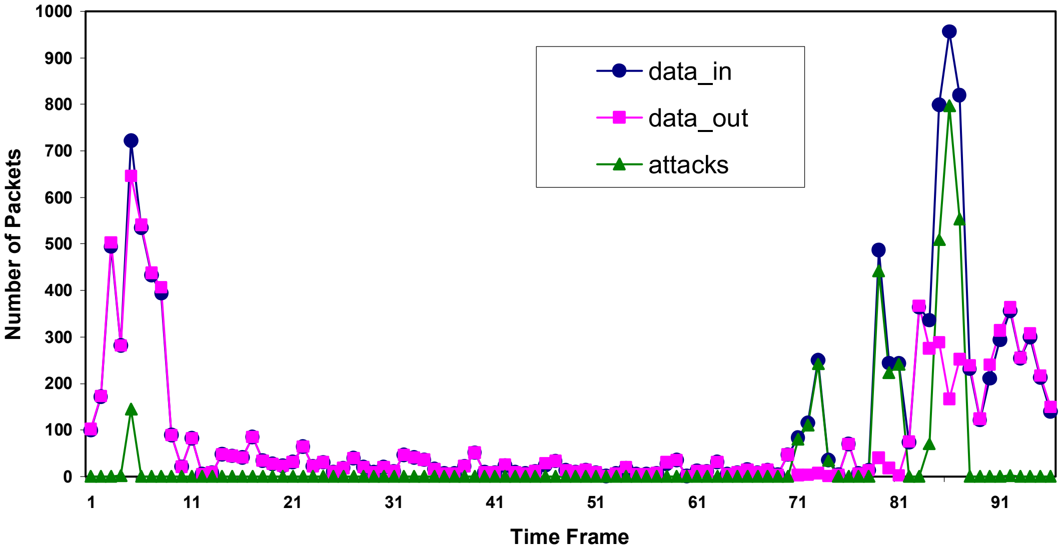

Figure 7. Processed {request, response} pairs by the DAAD tool

V Conclusions & Future

Capitalizing on the extended DNS functionality and the availability of large number of open resolvers that allow recursive DNS queries from arbitrary sources attackers try to exploit the powerful nature of DNS amplification attacks. The critical factor here is the amplification effect that is based on the fact that tiny queries can potentially generate much larger UDP packets in response. In this paper several aspects of these attacks were discussed and

Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007)
0-7695-2941-0/07 $25.00 © 2007 IEEE

analyzed. On the top of that, we presented a novel, practical and efficient scheme, namely DAAD, to defend against them.

While in its current pilot stage the proposed solution is easy to implement in any network realm it has at the same time a main drawback. This is based on the fact that the database size would increase rapidly in cases of high traffic rate. As a result alternative data stores like Bloom Filters [21] should be investigated. This would not only improve the performance of the DAAD tool, but make it scalable as well. One the other hand, detection accuracy, in terms of false positive rate and false negative rate, and the overheads in terms of performance are currently under inspection.

References

[1] Cert Advisory CA-1996-26, "Denial of Service Attack via ping", http://www.cert.org/advisories/CA-1996-26.html, December 1997.

[2] Gibson, S., “DRDoS Distributed Reflection Denial of Service”,http://grc.com/dos/drdos.htm, 2002.

[3] Glenn C., George Kesidis, G., Brooks, R. R. and Suresh Rai, “Denial-of-Service Attack-Detection Techniques” IEEE Internet computing 2006.

[4] Peng, T., Leckie, C. and Kotagiri, R., "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems", Accepted by ACM Computing Surveys.

[5] Mirkovic, J. et al., Internet Denial of Service: Attack and Defense Mechanism.

[6] Security and Stability Advisory Committee, “DNS Distributed Denial of Service (DDoS) Attacks”, http://www.icann.org/committees/security/ dns-ddos-advisory-31mar06.pdf, March 2006.

[7] Mockapetris P., “Domain Names – Concepts and Facilities”, RFC 1034, November 1987.

[8] Mockapetris P., “Domain Names – Implementation and Specification”, RFC 1035, November 1987.

[9] Vixie P., “Extension Mechanisms for DNS”, RFC 2671, August 1999.

[10] Arends, R., Austein, R., Larson, M., Massey, D., Rose, S., "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[11]Arends, R., Austein, R., Larson, M., Massey, D., Rose, S., “Resource Records for the DNS Security Extensions”, RFC 4034, March 2005.

[12] Arends, R., Austein, R., Larson, M., Massey, D., Rose, S., “Protocol Modifications for the DNS Security Extensions”, RFC 4035, March 2005.

[13] Guo, F., Chen, J., and Chiueh, T., “Spoof Detection for Preventing DoS Attacks against DNS Servers”, In Proceedings of the 26th IEEE international Conference on Distributed Computing Systems , July 2006

[14] Chandramouli, R. and Rose, S. “An Integrity Verification Scheme for DNS Zone file based on Security Impact Analysis”, In Proceedings of the 21st Annual Computer Security Applications Conference, December 2005.

[15] Atkins, D., Austein, R., “Threat Analysis of the Domain Name System (DNS)”, RFC 3833, Auguest 2004.

[16] IPTraf - An IP Network Monitor, http://iptraf.seul.org/.

[17] Vaughn, R. and Evron, G., “DNS Amplification Attacks, A preliminary release”, March 17, 2006.

[18] ICANN Report, “DNS Distributed Denial of Service (DDoS) Attacks”, Security and Stability Advisory Committee (SSAC), March 2006.

[19] Vixie, P., SAC004, Securing The Edge, http://www.icann.org/committees/security/sac004.txt.

[20] Guo, F., Chen, J. and Chiueh, T. “Spoof Detection for Preventing DoS Attacks against DNS Servers,” in Proc. of ICDCS 2006.

[21] Bloom, B., “Space/time trade-offs in hash coding with allowable errors” Communications of ACM, 13(7), pp. 422-426, July 1970.