Protecting the DNS Infrastructure of a Top Level Domain: Real-Time monitoring with Network Sensors

João Afonso
FCCN [1]
Lisbon, Portugal
joao.afonso@fccn.pt

Pedro Veiga
University of Lisbon
Lisbon, Portugal
pedro.veiga@di.fc.ul.pt

Abstract

In this paper we propose a solution to strengthen the security of Domain Name System (DNS) servers associated with one or more Top Level Domains (TLD). The proposed solution has been developed and tested at FCCN, the TLD manager for the .PT domain. Through the implementation of network probes that monitor the network in real-time, we are able to dynamically prevent, detect or limit the scope of attempted intrusions or other types of attacks to the DNS service. The platform relies heavily on cross-correlation allowing data from a particular sensor to be shared with the others. Administration tasks such as setting up alarms or performing statistical analysis are made through a web-based interface.

1. Introduction

The DNS is a critical application for the reliable and trustworthy operation of the Internet. DNS servers assume a pivotal role in the normal functioning of IP networks today and any disturbance to their normal operation can have a dramatic impact on the service they provide and on the global Internet.

Although based on a small set of basic rules, stored in files, and distributed hierarchically, the DNS service has evolved into a very complex system [1].

According to recent studies [2], there are nearly 11.7 million public DNS servers on the Internet. It is estimated that nearly 52% of them allow arbitrary queries (thus allowing denial of service attacks or "poisoning" of the cache).

About 31.1% of the servers also allow for the transfer of their areas of DNS.

They are still nearly 33% of the cases where the authoritative nameservers of an area are on the same network, which facilitates the attacks of Denial of Service (DOS).

Furthermore, the type of attacks targeting the DNS are becoming more sophisticated, making them more difficult to detect and control on time. Examples are the attacks by Fast Flux (ability to quickly move the DNS information about the domain to delay or evade detection) and its recent evolution to Double Flux [3].

A central aspect of a security system is the ability to collect statistically useful information about network traffic. This information can be used to monitor the effectiveness of the protective actions, to detect trends in the collected data that might suggest a new type of attack or simply to record important parameters to help improve the performance of the service.

The fact that the DNS is based on an autonomous database, distributed by hierarchy, means that whatever solution we use to monitor, it must respect this topology. In this paper we propose a distributed system using a network of sensors, which operate in conjunction with the DNS servers of one or more TLDs, monitoring in real-time the data that passes through them.

The ability to perform real-time analysis is crucial in the DNS area since it may be necessary to act in case of abuse, by blocking a particular access, and notifying the other sensors on the origin of the problem, since several types of attacks are directed to other DNS components

The use of a Firewall solution whose triggering rules are dynamically generated by the network sensors is a fundamental component of the system, to filter attacking systems and returning to the initial situation when the reason to filter different traffic patterns has

[1] Foundation for National Scientific Computing

1-4244-2575-4/08/$20.00 © 2008 IEEE

ceased to exists, guarantees an autonomous functioning of the platform. The use of alarms can also help in monitoring the correct functioning of the whole solution. Special care was taken to minimize the detection of false positives.

The remaining of the paper is structured as follows: Section 2 provides background information regarding related work. Section 3 introduces System Requirements. In section 4 we describe the proposed solution. Section 5 presents a case study for validation of the proposal. In Section 6 the results gathered in the case study are analyzed. Finally, Section 7 presents some conclusions and directions for further work.

2. Related Work

B. Guenter and R. Kolar, developed a tool called sqldjbdns [4]. Their proposal uses a modified version of the traditional BIND [5] working together with a Structured Query Language (SQL) version inside a Relational database management system (RDBMS). For DNS clients, this solution is transparent and there is no difference from classic BIND.

B. Zdrnja presented a system for Security Monitoring of DNS traffic [6], using network sensors without interfering with the DNS servers to be monitored. This is a transparent solution that does not compromise the high availability needed for the DNS service.

P. Vixie proposed a DNS traffic capture utility called, DNSCap [7]. This tool is able to produce binary data using pcap format, either on standard output or in successive dump files. The application is similar to tcpdump [8] – command line tool for monitoring network traffic, and has finer grained packet recognition tailored for DNS transactions and protocol options, allowing for instance to see the full DNS message when tcpdump only shows a one-line summary.

Another tool available is DSC - DNS Statistics Collector [9]. DSC is an application for collecting and analyzing statistics from busy DNS servers. Major features include the ability to parse, summarize and search inside DNS queries detail. All data is stored in an SQL database. This tool, can work inside a DNS server or in another server that "captures" bi-directional traffic for a DNS node.

J. Kristoff also proposed an automated incident response system using BIND query logs [10]. This particular system, besides the common statistical analysis, also provides information regarding the kind of consultations operated. All information is available through the Web based portal. Each security incident can result in port deactivation.

3. System Requirements

The main requirements of the system we developed are as follows:

Firewall
The system monitors the traffic after the corporate firewall that protects the network which houses the DNS server of a given TLD by examining all packets that enter and leave the DNS server. A set of pre-defined heuristic rules should determine, in real-time, the need to deny access to a given range of addresses by acting directly on the firewall. Later, after a period of quarantine, restrictions should be lifted.

Database
All information collected by the probes should be stored in a relational database (SGBDR) and all kinds of events related to the management of the DNS services associated with them.

Web Portal
The operative information regarding the various probes (security and statistical information) should be available via a Web interface.

Integrated operation
For an integrated protection of multiple DNS servers, it should be possible to exchange information between probes. This operation should prevent an attack on a server from a source, identified by another probe as malicious.

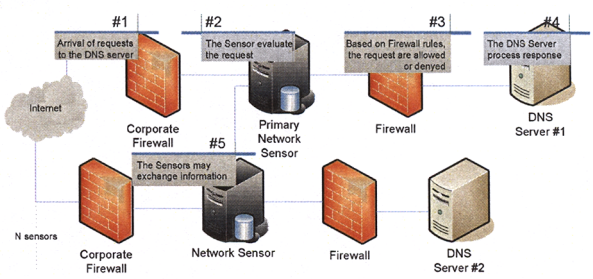

Figure 1. Diagram of the desired solution

Management of alarms
The solution should include appropriate alarm management features, with notification to the responsible servers, integrating mechanisms for instant messaging.

Open source
The solution should seek to use software packages available as Open Source due to the convenience of inspecting code prepared by others .

4. Proposed Solution

4.1 Overall Architecture

The solution proposed (Fig. 2) is based on a network of sensor engines that analyze all traffic to the DNS server in the form of valid or invalid queries, process information received from other probes and issue restrictions for specific network addresses. In case an abnormal behavior is detected or there is suspicious behavior from a certain address, it will be blocked in the firewall and the other probes notified so they can act accordingly.

The system can also calculate the response time for each operation to evaluate the performance of the server.

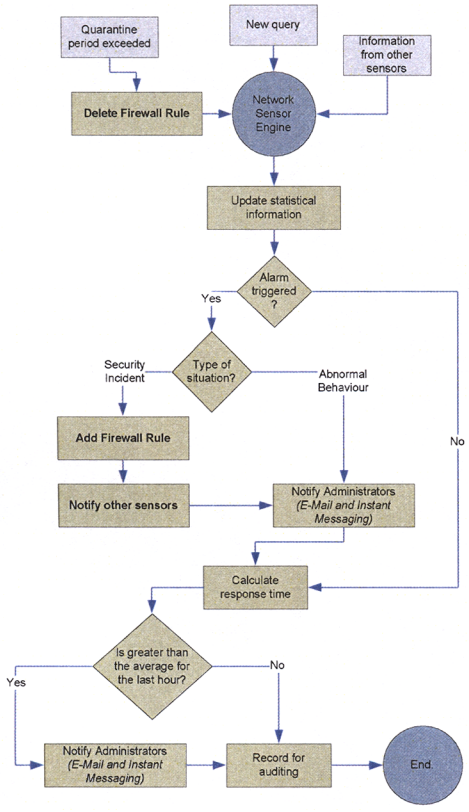

Figure 2. Block Diagram of proposed solution

According to our design, all data that flows through the probe heading for the DNS server is treated according to a standard set of global firewall rules, followed by specific local rules regarding to the addresses that are being blocked in real time. The packets are then delivered to the parser to be analyzed and stored in the RDBMS. At the top is the system of alarms and the Web portal (Fig. 3).

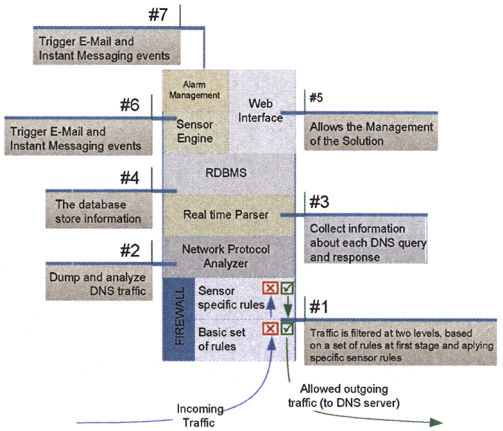

Figure 3. Network data flow

All information collected is stored in a MySQL [11] database. Taking into consideration the need to optimize the performance of the queries and to reduce the volume of information stored, the data is divided into a number of different tables (Fig. 4).

The conversion of the IP address of source and destination (DNS server) into an integer format, has allowed for much more efficient data storage, and significant improvements in the overall performance of the solution.

The information regarding all queries made, is stored daily into a log, and kept available during the next 30 days.

Two tables containing the set of rules that are dynamically applied – add or removed, based on situations that have been triggered - control the correct operation of the firewall. For auditing purposes every action is registered.

The information required for auditing and statistical tasks never expires.

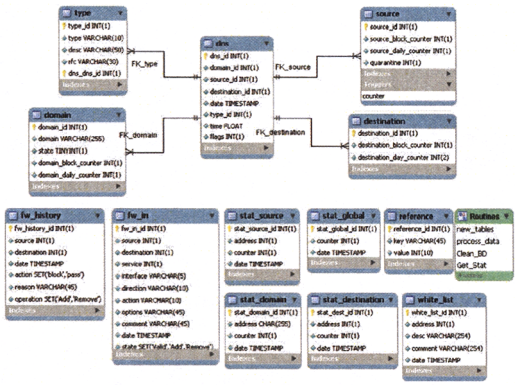

Figure 4. Database schematic

4.2 Methods of protection

A set of rules were defined that can detect in real time the possible incidents that can affect the DNS servers of a TLD. These rules are intended not only to combat any attacks, but also detect inappropriate behavior in the operation of a server, or even abuses arriving from a given source (which does not always mean misuse), thus allowing its notification.
Some of these rules are static, such as the number of queries allowed per hour or per day from a given source, which may not exceed certain values deemed abusive, such as those origination in DDOS attacks. Other rules are based on heuristics that are derived from the statistical evolution of access patterns to the servers.

Two of such examples can be found below:

- For each identified source IP address, the percentage of successful responses to requests, as a proportion of the answers missed, may not differ substantially from the standard statistics observed.
- The percentage of the consultations repeated cyclically by a given source, can not deviate considerably from the standard statistical values.

In addition to the rules that are shared by all sensors (global), each sensor has its own rules that result from the application of the heuristic methods to the (local) probe-specific traffic.

4.3 Managing notices

Whenever an alarm is triggered, administrators are notified by e-mail and instant messaging to allow a real-time monitoring of the situation. The option of Instant messaging is based on the XMPP protocol [12] using a virtual user that sends automated messages to admin users. In the same way, if a problem is solved, the supervisors are also informed.
In the case when the administrators are notified and take corrective measures, the resolution of problem is also followed by the system.

4.4 Dual operation mode: Sensor and Firewall

In order to deal with various incidents, and act upon them, after triggering a rule that could compromise the proper functioning of a server from a given source outside the TLD, the probe changes instantly the filter rules package that is active at the firewall in order to inhibit the transmission to the DNS server of requests from the source in question. The correction of the firewall rules through the introduction of restrictions to the IPFilter [13] - a stateful system firewall, enables the continuity of service for all the other consumers of information from the server.

4.5 Quarantine

Addresses considered suspicious, enter a period of deadlock that is usually not less than 2 days. The administrator can, of course, manually remove the restriction if, for some reason, the threat no longer exists or has been identified as a false positive. Otherwise, if the system does not continue to see failures from the rules defined, addresses are released by removing the restriction created at the firewall. These procedures are fully automated, and administrators notified of its operation.

4.6 White List

To avoid compromising the Internet service, considering the key role played by DNS, the White List protects key addresses from being blocked in case of false positives.
This list is created from a record of trusted sources, allowing all addresses listed here to be protected from being added to the Firewall rules. One example is the list of internal addresses, and the DNS servers of ISPs.

4.7 Statistical analysis

The statistical information collected and stored in the database has a significant amount of detail. It is possible, for example, to calculate, for each sensor, the evolution of queries per unit of time (hour, day, etc) badly formatted requests, DNS queries of rare types and determine the sources that produce the larger number of consultations. It is also possible to see the standard deviation of a given measure so we can relate it to that is seen with the other hits [14].

4.8 Performance evaluation

The performance of the servers is permanently measured, regarding the response time per request. Data is constantly registered and an alarm is raised in case normal times are exceeded.

4.9 Spread of rules between sensors

Better than the ability to react in real time to a situation such as denial of service situation, is the capacity to avoid such situations. With the spread of information between probes, it is possible for a probe to spread the information about a given source that is considered suspicious, allowing other servers to act in advance of a potential attack.

4.10 Web interface

All information collected in real time is available through a web interface, allowing authenticated users to consult the active firewall rules for the various probes, the events triggered, as well as detailed statistical information. The interface can also be used to perform management tasks such as the addition or removal of active rules and the inclusion of new situations to monitor.

Through this interface, after an alarm is triggered, it is possible to analyze the type of queries that were made by that source.

5. Case Study

Our proposal have been under development since September 2006 at FCCN – who has the responsibility to manage, register and maintain the domains under the .PT TLD.

At this time, there are two sensors running attached to the DNS servers (one at the primary DNS and another working together with a secondary DNS server).

The hardware chosen is a Sun fire T1000 server, 6 core, 1.0 GHz Ultra Sparc T1, 8GB DDR2 memory, running Solaris as operating system.

The network analyzer is tshark [15], and the firewall used is IPFilter. The real time parser was programmed in Java, collecting the information received from the tshark. The Web server is running Apache with PHP. Regarding the Xmpp server we choose the Jive messenger platform. All modules are integrated together as shown in Fig. 5.

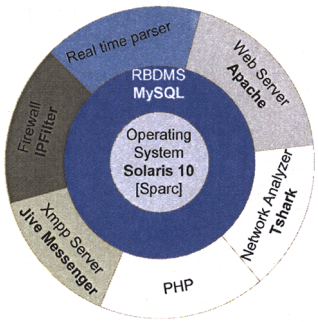

Figure 5. Hierarchical representation of the solution

The entire sensor solution, as described above, as well as the web platform we developed went on-line on the 1st of January 2007, and the data from the various agents was collected from the 10th of May 2008 till now.

Fig 6 shows a snapshot of the web interface developed.

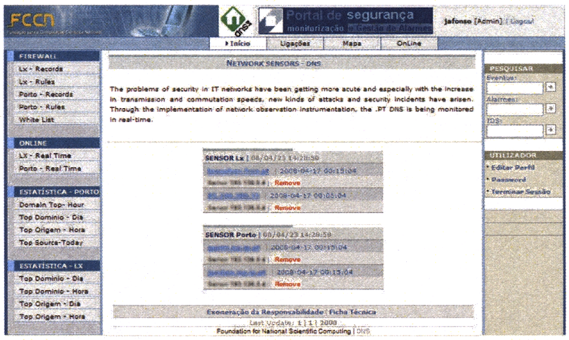

Figure 6. Web Interface

6. Results

We present here the results of the last 12 months of data collection (between May 2007 and May 2008).

The Average number of requests to the primary DNS server is up to 11,989,033 per day (138 per sec.).

The performance of the data analysis program is above 1240 requests processed per sec. (filtered, validated and inserted in the database).

Using the data collected by the sensors, during this time period, we were able to:

— Collect useful statistical information. E.g., daily statistics by type of registers accessed (Fig. 7).

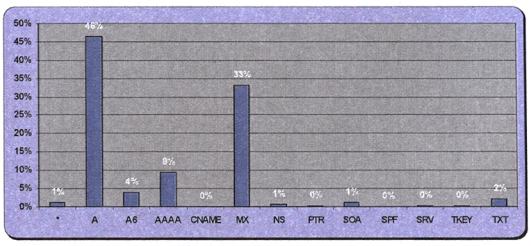

Figure 7. Statistical analysis by type of records accessed

— Obtain details regarding queries for DNS invalid domains: badly formatted or relative the domains that are not delegated in the .PT domain (Fig. 8);

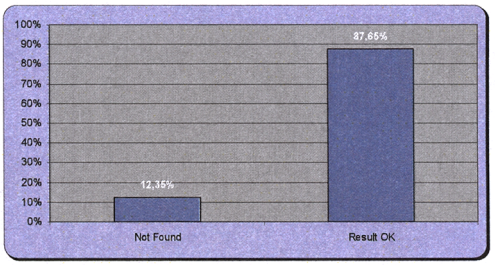

Figure 8. Records not found / OK

— Detect examples of abnormal use (that are not security incidents). For example we were able to detect that a given IP was using the primary .PT DNS server as location resolver. The number of queries made was excessive when compared with the average value per source, reaching values close to some Internet Service Providers that operate under the .PT domain.
— Estimate the impact of the removal of a given domain, noting the number of consultations made to the primary .PT zone, after its removal.
— Detect situations of abuse, including denial of service, with the execution of massive queries. In last 12 months of analysis there are 17 DOS attacks triggered. They were instantly blocked, and addresses placed in quarantine (Table 1).

Table 1. Examples when the sensor detected situations that required the Firewall rules to change.

Source Address	Date / Time	Operation	Sensor
xx.xx.200.35	2008-04-15 02:05:04	Add rule	xx.xx.44.62
xx.xx.94.139	2008-04-15 04:27:19	Add rule	xx.xx.44.63
xx.xx.13.231	2008-04-15 07:35:58	Remove rule	xx.xx.44.63

— Repair situations of inefficient parameterization of the DNS server. On the DNS server side, considering the capacity of the probe to determine the processing time for each consultation, it is possible to detect cases of excessive delay, which was later confirmed to coincide with of moments of zone update (Fig. 6).

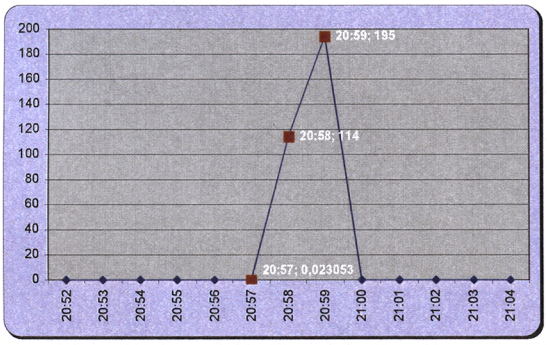

Figure 9. DNS query response time

The average response time for the server was approximately 0.022 seconds per query. During periods of zone update, it was possible to detect response times with values that reached a maximum of 194 sec. There were close to 500 operations that exceeded the response time of 1 second. This information was used to optimize the performance of the server

7. Conclusions and Future Work

The solution presented here, builds up on the existing solutions that collect statistical information regarding DNS services, by adding the ability to detect and control security incidents in real time. It also adds the advantage of operating in a distributed way, allowing the exchange of information between probes, and the reinforcement of its own security, even before it is threatened.

Currently, the solution presented does not allow the processing of addresses in the IPv6 format. The technical aspects that led to this situation are linked to the need to optimise the performance of the data recorder application making it possible to store the data from all consultations. Considering the issues of data storage optimization (unsigned int instead of char (15)) and processing speed, the conversion of IPv4 address is done using native MySQL functions. In the currently available version - 5.0.1 - only the IPv4 is contemplated. Nevertheless, all queries made to IPv6 addresses are contained in this solution (AAAA types).

Considering the expected increase of IPv6 and, as a consequence, the proliferation of addresses with IPv6 format, the update of the current architecture for the new IP version is currently being planned.

We are also working on extending the data correlation capabilities of the system by adding information collected from other sources (intrusion detection systems for instance). We anticipate that this could be a valuable approach to reduce considerably the number of false positives and negatives [16].

8. References

[1] P. Vixie, "DNS Complexity", ACM Queue vol. 5, no. 3, April 2007.
[2] D. Wessels, "A Recent DNS Survey", DNS-OARC, November 2007.
[3] ICCAN Security and Stability Advisory Committee, "Advisory on Fast Flux Hosting and DNS", January 2008.
[4] SQLDNS website, http://home.tiscali.cz:8080/~cz210552/sqldns.html.
[5] BIND website, http://www.isc.org/products/BIND.
[6] Bojan Zdrnja, "Security Monitoring of DNS traffic", May 2006.
[7] P. Vixie, D. Wessels, "DNSCAP – DNS traffic capture utility", CAIDA Workshop, July 2007.
[8] D. Wessels, "Whats New with DSC", DNS-OARC, November 2007.

[9] Lawrence Berkeley National Laboratory. Tcpdump website http://www.tcpdump.org.
[10] J. Kristoff, "An Automated Incident Response System Using BIND Query Logs", June 2006.
[11] MySQL website – (Open Source Database), http://www.mysql .com.
[12] P. Saint-Andre, Ed., Extensible Messaging and Presence Protocol (XMPP): Core, RFC 3920, 2004.
[13] IP FILTER – TCP/IP Firewall/NAT Software, http://coombs.anu.edu.au/~avalon.
[14] João Afonso, Pedro Veiga, "Analysis and implementation of security control measures based on real-time incident detection at DNS level", SINO2007, November 2007.
[15] Tshark website – The Wireshark Network Analyzer, http://www.wireshark.org.
[16] João Afonso, Edmundo Monteiro, "Development of an Integrated Solution for Intrusion Detection: A Model Based on Data Correlation", in Proc. of the IEEE ICNS'06, International Conference on Networking and Services - ICNS'06, Silicon Valley, USA, July 2006.