IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 15-19
www.iosrjournals.org

Protection of DNS Server in DNS Traffic

S.S.Suganya, Dr.V.Kathiresan

[1]*Department of Computer Science, Dr.SNS Rajalakshmi college of Arts & Science, Coimbatore – 641 049, Tamil Nadu, India*
[2]*Department of Computer Applications (PG), Dr.SNS Rajalakshmi college of Arts & Science, Coimbatore – 641 049, Tamil Nadu, India*

Abstract: *The domain name service (DNS) provides a demanding function in directing Internet traffic. The security of DNS is related to the whole Internet. DNS query log file provide the insights of the DNS security. Defending DNS servers from bandwidth violation is assisted by the ability to effectively mine DNS log data for statistical patterns. Almost every Internet communication is preceded by a translation of a DNS name to an IP address. DNS queries can be monitored through DNSQuerySniffer is a network snout utility that shows the DNS queries sent on your system. For every DNS query, the following information is displayed: Host Name, Port Number, Query ID, Request Type (A, AAAA, NS, MX, and so on), Request Time, Response Time, Duration, Response Code, Number of records, and the content of the returned DNS records. You can easily export the DNS queries information to csv/tab-delimited/xml/html file, or copy the DNS queries to the clipboard, and then paste them into Excel or other spreadsheet application.*

Keywords: *DNS, DNS Logs, DNS Queries, DNS Traffic*

I. Introduction

1.1What is DNS?

Domain Name Servers (DNS) are the Internet's exact of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.

This is necessary because, although domain names are easy for people to recall, computers or machines, access websites based on IP addresses.

Information from all the domain name servers across the Internet are gathered together and housed at the Central Collection. Host companies and Internet Service Providers interact with the Central Registry on a regular schedule to get updated DNS information.

When you type in a web address, your Internet Service Provider views the DNS combine with the domain name, translates it into a machine friendly IP address (for example 216.168.224.70 is the IP for jimsbikes.com) and directs your Internet connection to the correct website.

After you register a new domain name or when you update the DNS servers on your domain name, it usually takes about 12-36 hours for the domain name servers world-wide to be updated and able to connection the information. This 36-hour period is referred to as propagation.

1.2Why DNS traffic is important

DNS has a necessary role in how end users in your enterprise connect to the internet. Each connection made to a domain by the client devices is recorded in the DNS logs. Inspecting DNS traffic between client devices and your local repeated resolver could reveal a wealth of information for forensic analysis.

DNS queries can reveal:

- Botnets/Malware connecting to C&C servers
- What websites visited by an employee
- Which malicious and DGA domains were accessed
- Which dynamic domains (DynDNS) accessed
- DDOS attack detection like NXDomain, phantom domain. random sub domain

II. Identifying The Threats Using Event Tracker

While parsing each DNS log, we verify each domain accessed against:

- Malicious domain database (updated on regular basis)
- Domain Generation Algorithm (DGA)

Any domain which matches any of the above specified criteria warrants attention and an alert is developed along with the client which accessed it, and the geographic information of the domain (IP, Country).

Using behaviour analysis, Event Tracker tracks the volume of connections to each domain accessed in the enterprise. If the volume of traffic to a unique domain is more than average, alert conditions are triggered. When a domain is accessed for the first time, we check the following: Is this a dynamic domain?

- Is the domain registered recently or expiring soon?
- Does the domain have a known malicious TLD?

Recent trends show that cyber culprit may create dynamic domains as command and control centres. These domains are turn on for a very short period and then discarded, which makes the above checks even more important.

Event Tracker does analytical/threshold checking of query, client, record type and error. This helps in encounter many DDOS attacks like NXDOMAIN attack, Phantom domain attack, random sub-domain attack, etc. Event Tracker's following of client DNS settings will help to detect DNS hijacking and develop an alert for anything mistrustful, including information about the client as well as its DNS setting. The Event Tracker flex dashboard helps in correlating attack detection data and client details, making attack detection simpler. Monitoring the DNS logs is a powerful way to classify security attacks as they happen in the enterprise, enabling successful blocking of attacks and fixing vulnerabilities

2.1 DNS server

The Domain Name System (aka DNS) is used to resolve human-readable hostnames like www.Dyn.com into machine-readable IP addresses like *204.13.248.115*. DNS also provides other information about domain names, such as mail services.

2.2Need for DNS?

DNS is like a phone book for the Internet. If you know a person's name but don't know their telephone number, you can simply look it up in a phone book. DNS provides this same service to the Internet.

When you visit *http://dyn.com* in a browser, your computer uses DNS to retrieve the website's IP address of *204.13.248.115*. Without DNS, you would only be able to visit our website (or any website) by visiting its IP address directly, such as *http://204.13.248.115*.

2.3How does DNS work?

Fig: How DNS Work

When you visit a domain such as *dyn.com,* your computer follows a series of steps to turn the human-readable web address into a machine-readable IP address. This happens every time you use a domain name, whether you are watching websites, sending email or listening to Internet radio stations like Pandora.

Step 1: Request information

The process begins when you ask your computer to resolve a hostname, such as visiting *http://dyn.com*. The first place your computer looks is its local DNS cache, which stores information that your computer has recently retrieved. If your computer doesn't already know the answer, it needs to perform a **DNS query** to find out.

Step 2: Ask the recurrent DNS servers

If the information is not stored locally, your computer queries (contacts) your ISP's **recurrent DNS servers**. These specialized computers execute the legwork of a DNS query on your behalf. Recursive servers have their own caches, so the process usually ends here and the information is exchanged to the user.

Step 3: Ask the root name servers

If the recursive servers don't have the answer, they query the **root name servers**. A **name server** is a computer that answers questions about domain names, such as IP addresses. The thirteen root name servers act as a kind of telephone switchboard for DNS. They don't know the answer, but they can direct our query to someone that knows where to find it.

Step 4: Ask the TLD name servers

The root name servers will look at the first part of our request, reading from right to left — *www.dyn.**com*** — and direct our query to the **Top-Level Domain (TLD) name servers** for *.com*. Each TLD, such as *.com*, *.org*, and *.us*, have their own set of name servers, which act like a receptionist for each TLD. These servers don't have the information we need, but they can refer us directly to the servers that *do* have the information.

Step 5: Ask the authoritative DNS servers

The TLD name servers review the next part of our request — *www.**dyn.com*** — and direct our query to the name servers answerable for this *specific* domain. These **authoritative name servers** are important for knowing all the information about a specific domain, which are stored in **DNS records**. There are many types of records, which each contain a different kind of information. In this example, we want to know the IP address for *www.dyndns.com,* so we ask the accurate name server for the **Address Record (A)**.

Step 6: Retrieve the record

The recursive server retrieves the A record for *dyn.com* from the authentic name servers and stores the record in its local cache. If anyone else requests the host record for *dyn.com*, the recursive servers will already have the answer and will not need to go through the lookup process again. All records have a **time-to-live** value, which is like an expiration date. After a while, the recursive server will need to ask for a new copy of the record to make sure the information doesn't become out-of-date.

Step 7: Receive the answer

Armed with the answer, recursive server returns the A record back to your computer. Your computer stores the record in its cache, reads the IP address from the record, and then passes this information to your browser. The browser then opens a connection to the web server and receives the website. This entire process, from start to finish, takes only milliseconds to complete.

III. Protection Of DNS Server

3.1Use DNS forwarders

A DNS forwarder is a DNS server that performs DNS queries on behalf of another DNS server. The primary reasons to use a DNS forwarder are to offload processing duties from the DNS server forwarding the query to the forwarder and to benefit from the potentially larger DNS cache on the DNS forwarder.

Another gain of using a DNS forwarder is that it prevents the DNS server forwarding the requests from combine with Internet DNS servers. This is exclusively important when your DNS server is hosting your internal domain DNS resource records.

3.2 Use caching-only DNS servers

A caching-only DNS server is one that is not authoritative for any DNS domains. It's configured to perform recursion or use a forwarder. When the caching-only DNS server receives a response, it caches the result and returns the answer to the system issuing the DNS query to the caching-only DNS server. Over time, the caching-only DNS server can amass a large cache of DNS responses, which can significantly improve DNS response times for DNS clients of that caching-only DNS server.

Caching-only DNS servers can improve security for your organization when used as forwarders that are under your administrative control. Internal DNS servers can be configured to use the caching-only DNS server as their forwarders and the caching-only DNS server performs recursion on behalf of your internal DNS servers.

3.3 Use DNS advertisers

A DNS advertiser is a DNS server that resolves queries for domains for which the DNS advertiser is authoritative. For example, if you host publicly available resources for *domain.com* and *corp.com*, your public DNS server would be configured with DNS zone files for the *domain.com*and *corp.com* domains.

What sets the DNS advertiser apart from any other DNS server hosting DNS zone files is that the DNS advertiser answers queries only for domains for which it is authoritative. The DNS server will not perform recursion for queries to other DNS servers. This prevents users from using your public DNS server to resolve names in other domains.

3.4 Use DNS resolvers

A DNS resolver is a DNS server that can perform recursion to resolve names for domains for which that DNS server is not authoritative. For example, you might have a DNS server on your internal network that's authoritative for your internal network domain, *internalcorp.com*. When a client on your network uses that DNS server to resolve the name *techrepublic.com*, that DNS server performs recursion by querying other DNS servers to get the answer.

The difference between this DNS server and a DNS resolver is that a DNS resolver is a DNS server that is dedicated to resolving Internet host names. A resolver could be a caching-only DNS server that isn't authoritative for any DNS domains. You can make the DNS resolver available to only your internal users, you can make it available only to your external users to provide a secure alternative to using a DNS server outside of your administrative control, or you can allow both internal and external users access to the DNS resolver.

3.5 Protect DNS from cache pollution

DNS cache pollution is an increasingly common problem. Most DNS servers are able to cache the results of DNS queries before forwarding the response to the host issuing the query. The DNS cache can significantly improve DNS query performance throughout your organization. The problem is that if the DNS server cache is "polluted" with bogus DNS entries, users can subsequently be forwarded to malicious Web sites instead of the sites they intended to visit.

Most DNS servers can be configured to prevent cache pollution. The Windows Server 2003 DNS server is configured to prevent cache pollution by default. If you're using a Windows 2000 DNS server, you can configure it to prevent cache pollution by opening the Properties dialog box for the DNS server and clicking the Advanced tab. Select the Prevent Cache Pollution check box and restart the DNS server.

3.6 Enable DDNS for secure connections only

Many DNS servers accept dynamic updates. The dynamic update feature enables these DNS servers to register DNS host names and IP addresses for hosts that use DHCP for host IP addressing. DDNS can be a great boon in reducing the organizational overhead for DNS administrators who otherwise would need to manually configure DNS resource records for these hosts.

You can reduce the risk of malicious DNS updates by requiring secure connections to the DNS server in order to perform the dynamic update. This is easily achieved by configuring your DNS server to use Active Directory integrated zones and requiring secure dynamic updates. All domain members will be able to dynamically update their DNS information in a secure context after you make this change.

3.7 Disable zone transfers

Zone transfers take place between primary and secondary DNS servers. Primary DNS servers that are authoritative for specific domains contain writable DNS zone files that are updated as needed. Secondary DNS servers received a read-only copy of these zone files from primary DNS servers. Secondary DNS servers are used to improved DNS query performance throughout an management or over the Internet. However, zone transfers are not limited to only secondary DNS servers. Anyone can issue a DNS query that will cause a DNS server configured to allow zone transfers to dump the entirety of its zone database files.

3.8 Use firewalls to control DNS access

Firewalls can be used to gain access control over who can connect to your DNS servers. For DNS servers that are used only for internal client queries, configure firewalls to block connections from external hosts to those DNS servers.

For DNS servers used as caching-only forwarders, configure firewalls to allow DNS queries only from those DNS servers that use the caching-only forwarders. An especially important firewall policy setting is to block internal users from using the DNS protocol to connect to external DNS servers.

3.9 Set access controls on DNS registry entries

On Windows-based DNS servers, you should configure access controls on the DNS server-related Registry settings so that only the accounts that require access to them are allowed to read or change those Registry settings.

The HKLM\Current Control Set\Services\DNS key should be configured to allow only the Administrator and System account access, and these accounts should have Full Control permissions.

3.10 Set access control on DNS file system entries

On Windows-based DNS servers, you should configure access controls on the DNS server-related file system entries so that only the accounts that require access to them are allowed to read or change those files.

The %system directory%\DNS folder and subfolders should be configured to allow only the system account to access the files, and the system account should be given Full Control permissions.

IV. Sample Ways To Monitor Dns Traffic For Security Threats

4.1Firewalls

This type of **firewall** is often built into routers, and filters TCP/IP **traffic** by protocol (UDP, TCP, IGMP, etc.), to/from IP address, and to/from port number. **DNS** mainly uses the UDP protocol - except for zone transfer which uses TCP. ... A **DNS** server listens for **requests** on port 53 (both UDP and TCP).

4.2Intrusion detection systems

An **intrusion detection system** (**IDS**) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any detected activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses filtering techniques to distinguish malicious activity from false alarms.

4.3Traffic analyzers

It is one of the DNS analytics. **Traffic Analyzer** is the surveillance (collection and **analysis**) of **DNS traffic** within a computer network. ... For example, **DNS** Analytics can be used to gather data on a lab where a large number of related requests for PC software updates are made.

Reference

Chapters in Books:

[1]. Marchal, S., Francois, J., Wagner, C., State, R., Dulaunoy, A., Engel, T., Festor, O.: DNSSM: A Large Scale Passive DNS Security Monitoring Framework. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. pp. 988–993 (Apr 2012).

[2]. Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C.: Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. ACM Trans. Inf. Syst. Secur. 16(4), 14:1–14:28 (Apr 2014).

[3]. Karasaridis, A., Meier-Hellstern, K., Hoeflin, D.: Detection of DNS Anomalies using Flow Data Analysis. In: Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE. pp. 1–6. IEEE (2006).

[4]. P. Bacher, T. Holz, M. Kotter, and G. WICH-ERSKI, "Know your enemy: Tracking botnets, 2005," URL http://www.honeynet.org/ papers/bots, vol. 4, pp. 24–33.

Books:

[5]. H. Choi, H. Lee, and H. Kim, "Botgad: detecting botnets by capturing group activities in network traffic," in Proceedings of the Fourth International ICST Conference on COMmunication System software and middleware. ACM, 2009.

[6]. Google Inc., Google Zeitgeist 2010, http://www.google.com/zeitgeist, 2010

[7]. D. Plonka and P. Barford, Context-aware Clustering of DNS Query Traffic, Proc. of IMC'08 Conference, 2008.

[8]. N. Brownlee, kc claffy, and E. Nemeth, "DNS measurements at a root server," in Proceedings of the IEEE GlobeCom, San Antonio, TX, Nov. 2001.