Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 1, January 2019, pg.154 – 162

DNS TUNNELING EFFECT ON DNS PACKET SIZES

Ugur Tanik GUDEKLI
Information Institute
Gazi University
Tunus Caddesi No:35 Çankaya-Ankara
Turkey
Tel.: 90-3122023801
ugurtanik.gudekli@gazi.edu.tr

Bunyamin CIYLAN
Information Institute
Gazi University
Tunus Caddesi No:35 Çankaya-Ankara
Turkey
Tel.: 90-3122023801
bciylan@gazi.edu.tr

Abstract— DNS is a basic protocol that allows web applications such as browsers to work based on domain names. DNS's purpose is not creating a command channel or a basic tunneling. But, in order to creating a basic tunneling, many helper applications have been developed. Because of not being designed for general data transmission, DNS is less noticeable than other protocols. The malicious people who perform the cyber-attack know that DNS is a well-structured and reliable protocol. These people are also aware that many organizations do not control DNS traffic for malicious activity. With DNS tunneling, cyber criminals can easily install rogue software on these vulnerable systems or add stolen information to DNS queries and create a confidential communication channel across most firewalls. Although the DNS tunnel has some legitimate uses, many tunneling examples are intended to damage it. There are many current tunneling set of tools on the internet, so DNS tunneling has become a fairly easy process that does not need a separate technical expertise. At the same time, DNS tunneling is often used in very complex and massive attacks, including those supported mostly by nation states or directly governed by the nation state. In this research paper, DNS tunnels are reviewed and dns packet size was tested for detection of dns tunneling. In the result we examined that if taking only dns packet size into account is enough to detect dns tunneling in a network or not and also calculated packet size mitigation accuracy for our future work. We prepared a data pool and test pool to calculate accuracy of test. And we shared accuracy of checking only packet size approach.

Keywords: dns attack, dns tunneling, ddos, vulnerability,dns hijacking

I. INTRODUCTION

Until these days, at organizations DNS Tunneling implementation was a bit difficult, so we have been feeling more secure. But with last improvements of several tools which can easily getting data from a network. [1] So these improvements become a threat for all of us.

With this research we reviewed and analyzed on package size changes especially when the tunneling is open.

There are a few approaches to detect dns tunneling today. We can accept them in 2 groups; payload detection approaches and traffic analysis approaches.

One of the technique for determining the payload analysis is doing an analyze for the size of the request and the response queries. In Bianco's blog post (2006) the author describes a method for describing suspicious DNS tunneling traffic according to the ratio of source and destination sizes. [2]

Van Horenbeeck and Butler named researchers tell that DNS tunnels can be detected by exploiting the entropy values of interrogated domains in their studies. [3]

One method of detecting tunneling is to look for a particular character string in DNS names. Real / legal DNS name lengths tend to be generally short, while coded domain names are more flexible and have a large character set. For this reason, in a study presented in 2011, it was proposed to look at the percentage of the numeric characters in the domain names (Bilge, 2011). [4]

The use of character frequency analyzes to detect domain names generated by DNS tunneling has also been proposed as a different study when real / legal domain names are thought to be understood at a certain level in common language constructs. (Born, 2010). [5]

Searching for records that are not commonly used by a generic client, such as txt, is another method of tunnel detection. (Pietraszek, 2004). [6]

One of the basic methods is to observe the high amount of traffic that occurs in a specific domain (Butler, 2011). [7]

Geographic considerations are another factor that can be used in detection methods. As recommended by Skoudis in 2012, 'Large amounts of DNS traffic to parts of the world where you do not do business'. [8]

The number of hostnames for a given domain can also be used as a responder. (Guy, 2009). DNS tunnel programs require a unique host name for each request. This can lead to a much larger number than a typical normal domain name. [9]

In our works we will try to implement a point scoring system to detect dns tunneling activities. In point scoring system each control will have a coefficient. Firstly, we will study to understand packet size coefficient with checking and explaining dns tunneling affect to dns packet size in this paper.

We explained DNS, how it works and its vulnerabilities in place in section II in detail and continued with explaining DNS Tunneling with tunneling tools in section III. In section IV we shared out measure result with a tunneling tool on DNS packets. And we summarized the result and future works in conclusion, section IV.

II. DNS

A. Overview

Technological DNS is a distributed naming system that uses linked DNS servers in a hierarchical structure. DNS translates domain names into IP addresses and vice versa. The DNS namespace is hierarchically organized. A domain name is a subnet of a namespace. Top-level domains (called TLDs) are located just below the root register. A domain name is made up of smaller units called zones. Internet activities such as web browsing require users to rely on DNS to quickly access the information needed to connect to remote main servers. DNS mapping is distributed over the internet in the authority hierarchy. [10]

There are usually dedicated IP address ranges and assigned domain names assigned to governments, universities and organizations, as well as Internet providers and companies to access their web pages; DNS servers are usually run to manage the mapping of these domains to these reserved IP addresses. Most URLs are created on the domain names of web servers to respond to client requests. [10]

DNS A client / server network communication system in which DNS clients request and receive responses from DNS servers. Requests that contain a name that allows the server to return the IP address are called forward DNS lookups. Requests known as reverse DNS lookups are also supported. [10]

DNS database systems are in the hierarchy of a specially customized database server. When a Web browser makes a request that includes Internet domain names, a piece of software, often referred to as a DNS resolver that is included in the network operating system, communicates primarily with a DNS server to determine the IP address of the requested server. If the DNS server does not have the necessary mapping associated with that request, it will forward this request to a different top-level DNS server in the hierarchy itself. [10]

After sending various routing and authorization messages in the DNS hierarchy, the IP address of the given host arrives at the DNS resolver and the resolver completes the request over the Internet Protocol. [10]

DNS also includes support for request caching and redundancy control. Most network operating systems provide the configuration of primary, secondary and tertiary DNS servers that can serve customers' initial requests. [10]

B. DNS Vulnerabilities

In recent widespread network penetration testing, there are many security vulnerabilities around the organization's DNS servers. Attackers who exploit this vulnerability also develop many attacks on DNS. Nearly all applications, including networked web browsing, e-mail, e-commerce, and IP telephony systems, are largely DNS-bound, so many cyber-attacks that target DNS are causing organizations to face greater financial and administrative risk. [11]

DNS servers can be expressed as phone books of the Internet environment. They maintain Internet Protocol (IP) addresses and do bi-directional translations. It is easier for people to search for domains and remember their domain names, but the same web browsers need IP addresses to connect to a web server; domain names must be converted to IP addresses. [11]

The two most common types of attacks that target DNS are: DNS cache poisoning and DNS amplification known as the type of service-blocking attack. [11]

DDoS attacks targeting the DNS are extremely effective and can often be difficult to prevent. No matter how strong the infrastructure of the applications, the slowdown or crumbling of applications can occur if the DNS infrastructure is overwhelmed by intense attack requests. [11]

If we will examine how can we reduce the risk of DNS upgrade attacks on domain names; Beginning by modifying the configuration of owned "open" recursive DNS servers will be the first step. So this type of attack will not be very effective. In some cases, a recursive DNS server can also be configured to make recursive queries only on behalf of previously granted IP addresses (see Fig.1 below). A co-operation with the internet service provider may be used to use the source IP authentication to reject DNS traffic with fake IP addresses before reaching the domain. Using specialized devices and ready DDoS mitigation services to reduce DDoS attacks will also be an alternative method of protection for some organizations. [11]

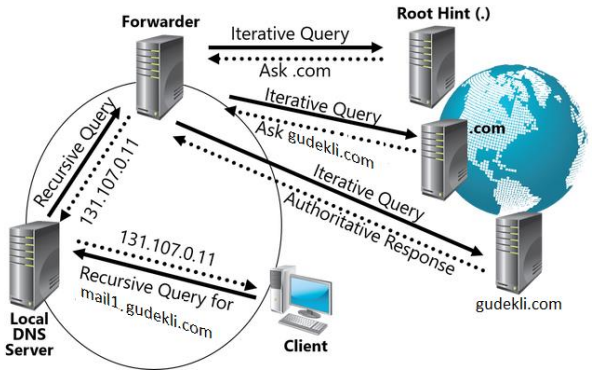

Fig.1 DNS Overview

In DNS cache poisoning, also known as DNS spoofing, hackers can upload spoofed data to the cache of a DNS server and redirect traffic to a malicious site by returning an incorrect IP address. Thus, hackers use a variety of vulnerabilities in the DNS software to load fake information, called poison, into DNS caches. To prevent this, DNS servers must verify that the DNS responses they receive from other servers are authentic; otherwise they can cache and present poisonous entries to the user and mislead them. [11]

The most effective way to protect against DNS cache poisoning is to ensure that DNS servers are up-to-date and correctly configured to have the latest security features of their software. For example, Domain Name System Security Extensions (DNSSEC) provide secure DNS authentication and reduce the risk of cache poisoning. [11]

If we look at other security measures; firewalls and intrusion prevention systems (IPS) provide security against DNS attacks and prevent attackers from taking over the administration of servers. Also using a network traffic analyzer, alerts can be received for traffic originating from malicious DNS queries on the network. I listed and explained a few DNS Vulnerability before DNS Tunneling topic.

1) DDoS

In a commonly known DDoS attack, an attacker obtains authorization in the system by taking advantage of a security vulnerability in a computer system and mastering itself. The attack master system finds systems with other security vulnerabilities and takes control by infecting these systems with malicious software or by bypassing authentication checks. [11]

A system under the control of a malware or any device connected to the network is called a zombie or bot. An attacker is creating a control server to command the networked bot called a botnet. The person controlling this botnet is also called the botnet master. Botnets can contain any number of bots; botnets of tens or hundreds of thousands of nodes are becoming increasingly widespread today and may not be the upper limit of their size. [11]

2) DNS Amplification Attacks

DNS upgrade is a method that used in DDoS attacks against DNS servers deployed in "recursive" configurations. Recursion is a commonly used useful and necessary DNS feature that allows domain name resolution to be transferred to a dairy level domain name server. However, attackers have begun to use "open" DNS servers, which cannot be controlled or whose authority cannot be controlled, to perform more effective DDoS attacks. [11]

3) DNS Hijacking

DNS Hijacking is a process which hackers can redirect a computer's TCP / IP settings to another pre-built DNS and present it to users. DNS Hijacking is a fraud method and often used as a phishing attack.
For DNS Hijacking, installing a malicious software to victim's computer is a prerequsite. The results of the DNS query of the victim's computer sent by this software to a different result. Thus, the user thinks that he has entered the original website, but in actual fact he is directed to the hacker-managed website.

The DNS Hijacking process is often used by hackers for phishing attacks called phishing. Internet service providers can do this to prevent access to a site. For example, if a different page is encountered when accessing a banned website, the service provider actually uses some kind of DNS Hijacking method. [11]

4) Cache Poisoning

Caching of content on the web to improve performance on the server and client side is a commonly used improvement method. Unfortunately, the protocol used in the caching process only performs some server-side checks. The lack of authentication for this reason makes it possible for attackers to create an attack method called cache poisoning.

In the event of cache poisoning, users accessing the poisoned web cache can view illegal content or content created by an attacker, or they can be redirected to an attacker-generated URL. This effect continues until the poisoned cache is cleared. Integrity technologies such as SRI (Sub report integrity) can be used to protect against this attack.

For cache poisoning in general, the following approach works:
- The attacker will examine vulnerabilities in the code contained in the web content and use them to add unwanted headers to the http head.
-Shielder deletes cached content from the original cache server

-The stealthy client sends specially crafted requests to the cache server to send custom malicious requests.
- Unwanted malware is placed and stored in the cache server.

As a result, web sites are becoming obsolete in this attack method, creating a distrust for users who use these sites. Using tools like DNSSEC to protect against this attack is a common method. [11]

III. DNS TUNNELING

DNS tunneling is a method that creating a secret communication channel between a computer in the network and an illegal server outside the network. Hackers use this method for command execution and control, data leakage or tunneling within any internet protocol traffic. Because of not designing data transfer environment with DNS, security professionals can easily ignore this threat of malicious communication or data leakage. [12]

Therefore, it is important to ensure that DNS requests are correctly monitored to create a secure environment. If there is evidence of data leakage or evidence of DNS tunneling, the necessary measures should be taken by reviewing DNS monitoring logs or examining query histories. Whether or not DNS tunneling can be detected by performing load analysis or traffic analysis. The density or size of the requests can be used to determine the tunneling. [12]

The combined use of these two analysis methods provides a stronger framework for taking measures against DNS tunneling and other threats.. [12]

A. DNS Tunneling Tools

There are a few DNS Tunneling tools today. We will summarize a few of them in this section.

1) Dns2tcp

The Dns2tcp tool is an application that allows TCP traffic to be tunneled using DNS traffic. Basically the Dns2tcp tool has two main components. The first is Dns2tcpd, which is usually run on a remote server, and the second is Dns2tcpc, which runs as a client. [12]
A configuration file on the server contains a list of resources that it has. Each source is a local or remote service that listens for TCP connections. The client application listens for a predefined TCP connection and forwards incoming connection requests to the endpoint service. [12]

2) DNSCat

Dnscat is designed to allow two servers to communicate with each other on the internet. It routes all traffic through a local DNS server. This structure has significant advantages:

- All network firewalls can be passed.
- Many local firewall can be passed.
- It is hidden because it does not pass through a known gateway / proxy.

In addition to these advantages, it also has some disadvantages.
For example; Because the DNS is not case sensitive, the data must be encoded alphanumeric. This means that the size is doubled as there is no compression.

3) Iodine

The Iodine tool allows IPv4 data to be tuned through a DNS server. This is useful in different situations where Internet access has a firewall, but DNS queries are also allowed.

When compared to other DNS tunneling tools, Iodine offers some advantages:
- It uses NULL data type, which allows download data to be transmitted without encoding.
- Unix systems and Windows32 platforms can work.
- Uses a challenge-response input method encrypted with MD5 hash method. In addition, it also filters all packets from other IPs except the login IP.

4) OzymanDNS

OzymanDNS is a tool that contains four basic Perl scripts. Two of these allow the end user to upload and download files using DNS. The other two scripts are in the role of a server-client structure. The server simulates a DNS server operating structure and is listening to port 53 for incoming requests. then, the Client computer converts

the incoming data into DNS requests sent to a specific area. These two commands can be used to create a tunnel with SSH. The end user will have to manually map the ports to pass the traffic passing through the tunnel.

IV. MEASUREMENTS

We have 2 collection of DNS traffic. First collection consists of clean traffic. The other one contains tunneled traffic.

A. Process

We prepared a high-performance workstation with 2 network cards for aggregating the cleaned traffic. One of these interfaces is for management and was named -MGMT-. And the external dns network traffic was copied with network tapping tools to other network interface card and was named -DNS-.

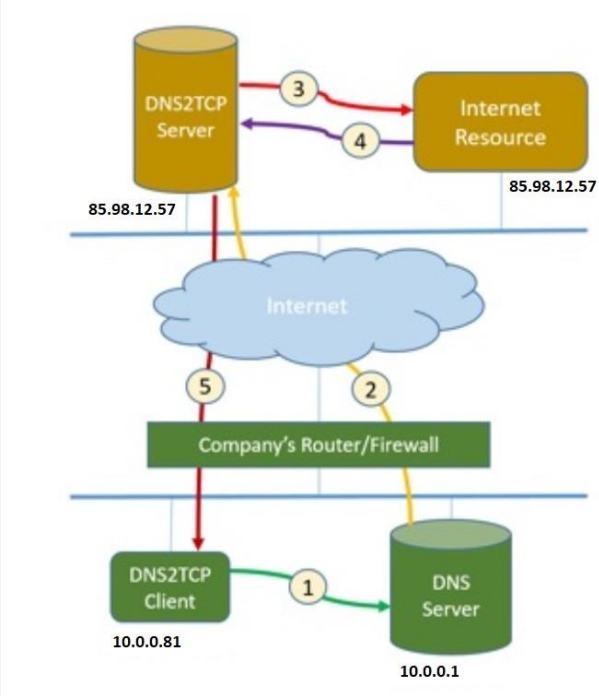

Fig 2. How DNS2TP Works

We ran the Wireshark [13] application on the workstation and collected traffic packets on the DNS interface. We had to make sure that this network did not have any dns tunneling activity. So, we considered the collected dns data as clean data. Our Wireshark query for getting DNS traffic was:

"Dns or port 53"

Recursive queries:

```
  udp port 53 and (udp[10] & 1 == 1) and src net not <net1> and src net not
<net2>
```

We had 150mb pcap data file. It was equal to 13170 dns query line with their response. Pcap file includes lots of different queries to normal domains cause captured from a client daily traffic.

As far as we found in the measurements we made, the average of the packets in clean DNS traffic is 116 bytes. After this we started to create some tunneled dns traffic with DNS Tunneling tools on prepared network. And continued to collect data to create tunneled – dirty packet pool.

Like most tunneling technologies, DNS2TCP requires a public domain which can be used for the DNS tunneling. Once a public domain is configured and DNS2TCP software is installed, we can start DNS2TCP tool to run SSH/POP/SMTP or any other applications.

We explained the detailed steps on how DNS2TCP works:

- Start DNS2TCP client from the laptop (in our setup, the IP address is 85.98.12.57), which has a default DNS server configuration (in our setup, the IP address is 10.0.0.1). When a user configures the DNS2TCP and starts an SSH session, the DNS2TCP client software will encapsulate SSH payloads into multiple subdomains on the pre-configured public tunneling domain and send these DNS subdomain requests to DNS server.

- Most domains can be resolved by DNS server without any issue, but for the DNS tunneling domain, (in our setup, we are using a fake domain, dns2tcp.tunnel.srt.blox), the DNS server cannot resolve them and will forward the request to the DNS2TCP server (the IP address is 10.0.0.81).

- The DNS2TCP server receives the DNS request, decapsulates the payload, and uses as a proxy to connect to the Internet resource. In our testbed, I setup an SSH server (85.98.12.57 – behind NAT – it has another internal IP) as Internet resource.

- Then Internet resource responds to the request and sends the payload to the DNS2TCP server.

- The DNS2TCP server encapsulates the response payload from the Internet into DNS response packet and sends back to the DNS2TCP client. The DNS2TCP client receives the DNS response traffic and decapsulates them. Then the client is able to receive all the response traffic from the Internet resource.

1) DNS2TCP Server

Firstly, we install dns2tcp package in the Ubuntu guest OS.

apt-get install dns2tcp

Then we configure the DNS2TCP server options. The listing IP address is its local network interface 85.98.12.57, and the port number is the DNS port tcp/53. The domain we will use for DNS tunnel is dns2tcp.tunnel.srt.blox.

Since the DNS2TCP server has an IP address is 85.98.12.57, we put the listing IP address on this interface, and port number to be 53. In this study, we only focus on SSH session over DNS tunneling.

```
root@kali-server:~# cat >>.dns2tcpdrc <<END
listen = 0.0.0.0
port = 53
user=nobody
chroot = /root/dns2tcp
pid_file = /var/run/dns2tcp.pid
domain = dns2tcp.kali.org
key = secretkey
resources = ssh:127.0.0.1:22
END
root@kali-server:~# dns2tcpd -f .dns2tcpdrc
root@kali-server:~#
```

Last, we start DNS tunneling daemon by running the command below:

dns2tcpd –F –d 3 –f dns2tcpdrc

2) DNS2TCP Client

Like DNS2TCP server, first we need to install dns2tcp package in the Ubuntu guest OS. Second, we configure the DNS2TCP client as follows.


```
root@ubuntu:~# more settings.tunnelclient.txt
domain = tunnel.gudekli.com
resource = ssh
local_port = 22
debug_level = 3
key = secretkey
server = dns2tcpdrc
root@ubuntu:~#
```

At last, we run DNS2TCP application as proxy in the client,

dns2tcpc –d 3 –f settings.tunnelclient.txt

Now, we can start SSH session to the SSH server 85.98.12.57 using DNS tunneling.
ssh jxia@127.0.0.1 –p 2222 –D 8081

It should be able to connect to the SSH server through DNS tunneling channel.

B. Results by Tool

We observed that the average size of outgoing / incoming packets in the tunnel we made using Dns2Tcp tool is 208 bytes. Remember the average size of the packets in clean DNS traffic was 116 bytes.

If we accept the tunneled packet size is equal to near 208 byte its mean is that if packet size increase %79 of its average size the traffic is tunneled. Let's measure the result with this acceptance to see real packet size affect on DNS Tunneling.

When we check the acceptance criteria on tunneled and clean data pool we get the classification [14] table as shown:

Table 1: Classificiation

Classification	Tunneled	Clean
Test Says Tunneled	TP:162	FP:691
Test Says Clean	FN:498	TN:388

TP: True Positive, Packet size is equal or over 208 bytes and it is really going to gudekli.com and tunneled.
FP: False Positive, Packet size is equal or over 208 bytes, but it is not a tunneled traffic.
FN: False Negative, Packet size is lower than 208 bytes, but it is a tunneled traffic because it is going gudekli.com
TN: True Negative, Packet size is lower than 208 bytes and it is not a tunneled traffic in real

$$Accuracy = \frac{TP+TN}{\text{Total Packets}} \rightarrow \frac{162+388}{1739} \rightarrow 0.31$$

Fig 3. Accuracy Rate

We can see the numbers of packets besides classification table.
Accuracy shows us querying only dns packet sizes to detect dns tunneling is not a good option with itself.
And we will accept this ratio value as packet size coefficient value at our point scoring system in future work.

V. CONCLUSIONS

We see that checking only packet size for dns tunneling detection is not enough to get a over corrected detection. DNS Tunneling Detection require to calculate more subjects than packet size. In future work we will continue to analyze concurrent packet count in a second and also, we will try to do a real time reputation analyze for requested domain.

REFERENCES

[1] DNS Tunneling Detection, International Journal of Recent Trends in Engineering & Research

[2] Bianco, D. (2006, May 3). A traffic-analysis approach to detecting dns tunnels. Retrieved from http://blog.vorant.com/2006/05/traffic-analysis-approach-to-detecting.html

[3] Van Horenbeeck, M. (2006). Dns tunneling. Retrieved from http://web.archive.org/web/20060613210141/http://www.daemon.be/maarten/dnst unnel.html

[4] Bilge, L. (2011). Exposure: Finding malicious domains using passive dns analysis. Retrieved from http://www.syssec-project.eu/media/page-media/3/bilgendss11.pdf

[5] Born, K. (2010a). Psudp: A passive approach to network-wide covert communication. Retrieved from http://www.kentonborn.com/sites/default/files/psudp_born_slides_bh_2010.pdf

[6] Pietraszek, T. (2004, October 31). Dnscat. Retrieved from http://tadek.pietraszek.org/projects/DNScat/

[7] Butler, P. (2011). Quantitatively analyzing stealthy communication channels. Informally published manuscript, Computer Science, Virginia Tech, Blacksburg, VA, .

[8] Skoudis, E. (2012, February 29). The six most dangerous new attack techniques and what's coming next?. Retrieved from https://blogs.sans.org/pentesting/files/2012/03/RSA-2012-EXP-108-Skoudis-Ullrich.pdf

[9] Skoudis, E. (2012, February 29). The six most dangerous new attack techniques and what's coming next?. Retrieved from https://blogs.sans.org/pentesting/files/2012/03/RSA-2012-EXP-108-Skoudis-Ullrich.pdf

[10] DNS in Action, A Detailed and Practical Guide to DNS Implementation, Configuration and Administration

[11] DNS Security, Defending the Domain Name System, Allan Liska – Geoffrey Stowe

[12] http://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152

[13] DNS2TCP Tool http://www.aldeid.com/wiki/Dns2tcp

[14] Wireshark Tool https://www.wireshark.org/

[15] Binary Classifiers http://www.wikizero.info/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvRXZhbHVhdGlvbl9vZl9iaW5hcnlfY2xhc3NpZmllcnM

