Multi-Disciplinary Journal
ISSN No- 2581-9879 (Online), 0076-2571 (Print)
www.mahratta.org, editor@mahratta.org

Review Paper on DNS Tunnelling

Sumeet Khole
sskhole@gmail.com

Dr. Anup Girdhar
Professor, Department of computer Science
Tilak Maharashtra Vidyapeeth, Pune-37

Abstract
Our internet is made possible by the DNS system; misuse or error can cause it to go down for several days, weeks, or even for a whole region. Similar accidents have in the past caused global disruptions. Tunnelling or command execution was not the intended intent of DNS. Several tools have been developed over time to enable tunnelling, which employs DNS tunnels to allow the transfer of arbitrary data across the DNS infrastructure. DNS security has not gotten as much attention lately because it was not intended for widespread data transport. Consequently, DNS intrusions increasingly pose serious security risks to companies. This paper will review the principles behind the working of DNS tunnelling and some of the utilities that can be used to support it and I will also discuss ways of preventing such attacks using payload analysis and traffic analysis techniques.

Keywords : Internet, DNS, Data transit.

Introduction
When searching for information, we routinely use the internet to do this task. Applications can utilize names to represent IP addresses, such as strathmore.edu, thanks to the Doman Name System (DNS), which is used by email clients and web browsers. DNS is not necessary for the internet protocol to function, according to Bojan, Nevil, and Duane (2007). But since users need to be able to identify different machines by name, the DNS protocol is necessary to resolve names and IP addresses. Despite not being intended for data transport, DNS can be exploited to conduct malicious communications or obtain personal information about clients (Greg Farnham, 2012). Most companies don't monitor DNS because they believe it to be low risk. Instead, they are keeping an eye on traffic that presents a poses a greater risk, including email and web traffic. Given that DNS is not Because it is centralized, attacks or exploitation from outside the company are possible. It also allows attackers to quickly alter domain name records or even exchange domain names to get around admin blocking. titles. Additionally, vulnerabilities can be exploited by attackers even in networks carefully controlled by organizations. in DNS traffic inadequate client-side security (Bojan, Nevil, & Duane 2007). Attackers have made use of this prohibition on using DNS to transfer data and to create rogue DNS servers that let them authority over the victim's internet behavior. According to Ryan W. Neal (2013), there was no truth to the rumors that Google Kenya had been hacked, and that a Bangladeshi hacker had taken control of a Kenyan DNS server and redirected users to another website. Many intricate attacks and exploitations can be made against the DNS protocol by employing a range of methods. The vast array of tools on the market are used to meet a variety of needs, including intrusive attacks, obtaining free internet from WiFi hotspots with captive portals, running remote operating system commands, file transfers, and even full IP tunnels, according to Greg Farnham (Detecting DNS Tunnelling, 2012). DNS signaling and DNS tunnelling are the two primary techniques for DNS manipulation. We'll talk about DNS tunneling in this essay.

Overview of DNS
DNS is an acronym for domain name system. As defined by Wikipedia, a domain name system is a distributed naming scheme for computers, gadgets, and/or any resource connected to the internet. It translates human-readable domain names into IP addresses that link to computers worldwide. Two of

the main features and requirements of DNS are scalability and availability. The DNS name space is segmented into zones according to locations or organizations in order to achieve this.

Each organization also has a single authoritative zone in the DNS hierarchy. A full qualified domain name (FQDN) represents a node's entire domain name. The entire path of a domain name, from a leaf to the tree root, is described in the FQDN. Every node is represented by a label that indicates its zone (Kenton & Dr. David, 2010). DNS traffic uses TCP or UDP to exchange data on port 53. However, the majority of its resolvers use UDP, which is where most of its communication takes place. TCP was initially limited to zone transfers, but it was later extended to allow data transfers for requests larger than 512 octets. The database that makes up the domain space is hosted by several name servers. Even if different regions of the domain space are maintained in separate names servers, a copy of a data item will be replicated across two or more name servers. Consequently, the standard goes on to specify the types of data name servers are adequate at reading. The domain spaces set of zones is the first kind. Periodically, authoritative data is checked to ensure that its zones are up to date; if not, a copy of the current information. Cache data is obtained by a local resolver and is the other type. This improves output. when local data is retrieved and is accessed multiple times. Figure below outlines the steps involved in taking care of a domain name.

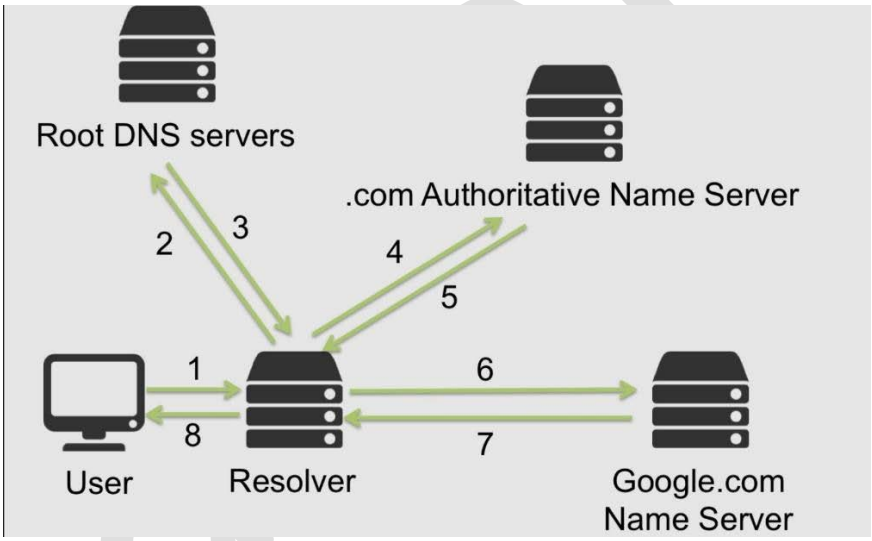

Figure1–Process of Resolving DNS name.

DNS Tunneling

DNS tunneling enables the tunneling of other protocols via DNS. DNS tunnelling, according to Beauregard (2013), is a method that enables hackers to encrypt data from other protocols or applications into DNS queries and answers. Concerns have been raised about this since the late 1990s. DNS tunneling was first intended to get around captive portals in WiFi hotspots, but as time has gone on, its use has grown and presented more risks to businesses. At the highest level of tunneling, a client would need to be taken advantage of via social engineering, phishing, or malware. It is not necessary for the client to have internet access for them to be compromised. All that is required of the client is access to an internal DNS server.

Is there a risk to security?

Consequently, in order to enable server side tunneling and decoding programs, an attacker needs to possess a domain and a server that will serve as an authoritative server for that domain. In a typical situation, DNS Data is encapsulated in queries and replies using base32 and base64 during tunneling over a tunnel encrypting. The bidirectional data request is then sent via the DNS domain name lookup system. Thus, for You can tunnel any type of data you want to, including any common and successful domain name lookups on a network a distant system.

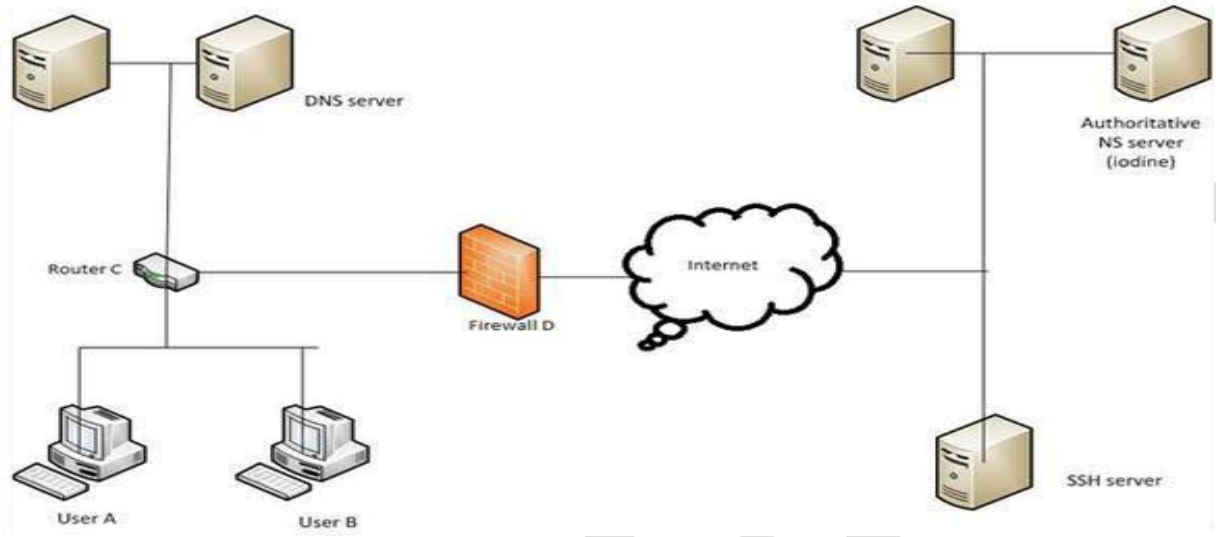

Figure2-Fruz2014

Fruz (2014) states that in the a forementioned scenario, User A and User B are seated behind a corporate firewall D. Except for traffic arriving via port 53, all traffic is blocked by the firewall policies. However, user A and user B can access the internet by tunneling DNS traffic. User A will attempt to visit any websites that are cached by the DNS server on the left, as it has the ability to cache data; the request will not be sent to the iterative server.As a result, when user A makes a new request, the DNS will forward it to an external DNS server since it cannot locate its A record on the DNS server. User Tools for DNS Tunneling - OzymanDNS

Plenz (2006) states that the OzymanDNS client is a Perl script that uses a DNS request to encode and send all data on STDIN to its destination. Responses are then sent to STDOUT. Thus, it is not functional as a 23-10-2023 Page 4/6 of https://www.checkforplag.com standalone program, but it will be utilized to tunnel traffic via SSH. Tcp Dns2

Nicolas Collignon and Olivier Dembour wrote dns2tcp. It runs on Linux and is written in C. The client is compatible with Windows. According to Debour (2008), it supports KEY and TXT request types. Dodecyl Updates to the DNS tunneling program Iodine were made as recently as 2010. It was first made available in 2006. Erik Ekman and Bjorn Andersson developed it. Iodine is a C-written program that runs on Windows, Linux, and Mac OS X. These are the clients that help smart phones with DNS tunneling. Marcel (2012) claims that element53 is an Android DNS tunnel that is ready to use. A TCP tunnel is established to a SOCKS proxy server.

Identification of DNS Tunneling. Payload evaluation. One method for analyzing the payload is payload analysis content of a DNS request or response to look for irregularities. Numerous studies have produced methods instead of examining a DNS query's content, it is predicated on examining different facets and behaviors displayed by the guidelines. Fruz (2014) states that determining DNS anomalies entails determining the length of characters that appear in a DNS request and answer. A typical request shouldn't be longer than 64 characters.

Traffic Examination
This method primarily entails examining the traffic destined for a specific domain to identify irregularities that may indicate DNS tunneling requests. As a result, DNS trap and other similar tools use artificial neural networks to detect tunneling. The artificial neural network is trained to detect tunneling using five attributes by the tool. The domain name, the quantity of packets sent to a specific domain, the average length of the packets sent, the average number of distinct characters in the Low-Level Discovery, and the separation between the Low-Level Discovery are some examples of these characteristics. Strategies to Avoid DNS Tunneling
Now, statistical anomaly detection on the network is the most effective technique to detect DNS tunneling. For example, if you are operating a premium Wi-Fi hotspot, you should make sure that the server responds to all local queries until a payment is received; after that, the client can use DNS lookup to access the internet. Additionally, you can stop it by making sure that requests are not processed recursively for any other zones or domains (Finux, 2011). In addition, blocking TXT record queries is unlikely to result in DNS tunneling and doesn't complicate the system. Miller (2005) claims that snort signatures can be designed to provide alerts when a significant number of TXT DNS requests happen quickly.

Additionally, he recommends that businesses use split DNS. Here's where the client-side systems fall short web proxies are used in its place to resolve external domains to facilitate web browsing. This is because it stops DNS requests from outside the internal network from leaving. In summary As it happens, one of the best covert channels ever created is DNS tunneling. It's a difficult undertaking to halt this traffic because there is no clear indication that IP over DNS is involved. burrowing. However, several ways have been devised to lessen the dangers and risks associated with DNS tunneling.
Given DNS Tunnelling's nature, attackers can take advantage of some of its flaws to permit free mobile phone internet access. Therefore, organizations will experience losses and potentially dangerous attacks if they fail to take the required precautions to stop this from happening. Citations [1] Beauregard Chap. [2 0 1 3] Name Servers Turnaround time: I am the creator of the product. Return to the original Not a single one dns tunneling security threat[2] can be found at https://www.neustar.biz/blog. G. Farnham (2013).

Tools User for DNS tunneling
Ozyman DNS
According to Plenz (2006) OzymanDNS client is a Perl script which encodes and transfers everything on STDIN to its destination via a DNS request then replies are written to STDOUT. So, it cannot work as a standalone program but will be used with SSH to tunnel traffic.

Dns2tcp
Dns2tcp was written by Olivier Dembour and Nicolas Collignon. It is written in C andruns on Linux. The client can run on Windows. It supports KEY and TXT request types(Dembour,2008).

Iodine
Iodine is a DNS tunneling program first released in 2006 with updates as recently as 2010 .It was developed by Bjorn Andersson and Erik Ekman. Iodine is written in C and it runs on Linux, Mac OS X, Windows and others. Iodine has been ported to Android. It uses a tun or tap interface on the end point(Andersson,2010).

Magic Tunnel, Element 53,VPN-over-DNS(Android)
These are clients that facilitate tunneling over DNS in smart phones. According to Marcel (2012) element53 is a ready to use DNS-tunnel for android. It creates a TCP-tunnel to a SOCKS proxy server.

Detecting DNS Tunneling.

Payload Analysis.
Payload analysis is a technique that is used to analyze the content of DNS query or response in order to detect anomalies. Various research have come up with ways of analyzing the content of a DNS query, it is based on analyzing various aspects and behaviors exhibited by the protocol. According to Fruz (2014) identifying DNS anomalies involves checking the length of characters in a DNS request and response. Often the length of a normal request should not be more than 64 characters. So it's likely that tunneled traffic will have more than 64characters.

Traffic Analysis
This technique majorly involves analyzing the traffic going to a certain domain in order to detect anomalies which might contain potential DNS tunneling requests. Accordingly, tools like DNStrap help in detecting tunneling by using artificial neural networks. The tool uses 5 attributes to train the artificial neural network to detect tunneling. These attributes include the domain name, the number of packets send to a certain domain, the average length of packets sent, the average number of distinct characters in the Low Level Discovery and the distance between the Low Level Discovery.

Ways of Preventing DNS Tunneling
Currently the best way of identifying DNS tunneling is through statistical anomaly detection on the network. For instance, if you are running a premium Wi-Fi hotspot you should ensure that the server answers to all local queries until a payment has been made, thereafter the client can access internet via DNS lookup. Also, you can prevent it by ensuring that requests to any other domains or zones are not handled recursively (Finux, 2011). Also preventing queries for TXT records is most likely to prevent DNS tunneling and does not bring complications to the system.
According to Miller (2005) snort signatures can be created to alert when largenumberofTXTDNSrequestsoccursoverashortperiodoftimeandgivealertsonmultiple large DNS requests, or large number of DNS requests going to a single domain. He also suggests that organizations should implement split DNS. This is where the client-side systems cannot be able to resolve external domains and instead web proxies are used resolve external domains for web browsing. This is because it prevents external DNS requests from exiting the internal network.

Conclusion
DNS Tunneling happens to be one of the best covert channels that was ever designed. It is a very challenging task to stop this traffic, because there is no specific indication that it concerns IP over DNS tunneling. But various solutions have beendeveloped to mitigate risk/threats arising from DNS tunneling. According to the nature of DNS Tunneling attackers can exploit some of its weaknesses to allow free internet access on mobile phones. So if organizations will not take necessary measures to prevent this from happening they are going to suffer loses and attacks with high risk potential.

References
[1]Beauregard,C.(2013)DNSTunneling:Isitasecuritythreat?RetrievedfromNeustar:https://www.neustar.biz/blog/dns-tunneling-security-threat
[2] Farnham, G. (2013). Detecting DNS Tunneling. Retrieved from Sans:http://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling- 34152
Finux, A. (2011, April 27).
[3] Dns tunneling its all in the name. Retrieved fromSlideshare:www.slideshare.net/bsideslondon/dns-tunneling-its-all-in-the-name
[4]Fruz,A.(2014)DNSTunneling.RetrievedfromINFOSECINSTITUTE:http://resources.infosecinstitute.com/dns-tunneling

[5] Marcel. (2012). Android DNS tunnel: Element53. Retrieved from Marcel'sweblog:http://blog.bokhorst.biz/7681/computers-and-internet/android-dns-tunnel-element53
[6] Miller, T. (2005). Reverse DNS Tunneling Staged Loading Shellcode. Retrieved fromBlackhat: 08/Miller/BH_US_08_Ty_Miller_Reverse_DNS_Tunneling_Shellcode.pdfMockapertris, P. (n.d.). Domain Names - Implementation and Specification. RetrievedfromIETF: https://www.ietf.org/rfc/rfc1035.txt
[7] Plenz,J.(2006).DNStunnel.de.RetrievedfromDNStunnel.de:http://dnstunnel.de