A Case Study Solution to DNS Cache Poisoning Attacks

Siddhant Agarwal and Sanket Pramanick [1], Nidhi Bhandari[2], [3]Dr. G. Usha[*]

Department of Software Engineering, SRM University,
Kattankulathur, Tamil Nadu 603203, India
ushag2@gmail.com,usha.g@ktr.srmuniv.ac.in

Abstract— In this paper we discuss about DNS cache poisoning attack, which exploits the vulnerabilities in the domain name system to divert Internet traffic away from legitimate servers and towards fake ones. DNS converts human-readable addresses like "google.com" to computer-readable IP addresses like "173.194.67.102". The Internet service provider runs its own DNS servers, which cache information from other DNS servers. The home router functions as a DNS server, which caches information from the ISP's DNS servers. The computer has a local DNS cache, so it can quickly refer to DNS lookups it's already performed rather than performing a DNS lookup over and over again. A DNS cache can become poisoned if it contains an incorrect entry. For example, if an attacker gets control of a DNS server and changes some of the information on it, let's say they make google.com actually point to an IP address the attacker owns. That DNS server would tell its users to look for Google.com at the wrong address and the attacker's address could contain some sort of malicious phishing website. DNS poisoning like this can also spread. For example, if various Internet service providers are getting their DNS information from the compromised server, the poisoned DNS entry will spread to the Internet service providers and be cached there. It will then spread to home routers and the DNS caches on computers as they look up the DNS entry, receive the incorrect response, and store it. Initially, this paper provides an introduction about DNS systems with an architecture diagram. Second, this paper discusses about various types of case studies in literature. Third, this paper also suggests solutions to avoid DNS cache poisoning attack.

Keywords- DNS; DNS cache poisoning; DNS protection

1. INTRODUCTION

DNS stands for "Domain Name System." Domain names are the human-readable website addresses that are used by every human in day-to-day life[1]. DNS works transparently in the background that is used to convert human-readable website names into computer-readable numerical IP addresses. DNS suffers for various types of security problems. One of the most vulnerable problems is known as DNS Cache Poisoning.

DNS Cache poisoning, also called domain name system (DNS) poisoning or DNS cache poisoning.DNS cache poisoning[2] is the process of corrupting an Internet server's domain name system table by replacing an Internet address with that of another, rogue(malicious) address. When a user seeks the page with that address, the request is redirected by the rogue entry in the table to a different address. The

following Fig. 1 is an example for DNS cache poisoning attack. The attacker replaces the alternate address record in the local DNS server. The home client forwards the Domain Name and IP (malicious) using HTTP Client. In this way the DNS cache gets poisoned and malicious packet gets forwarded from one client to other client. Next we discuss about the Cache Poisoning Technique in detail.

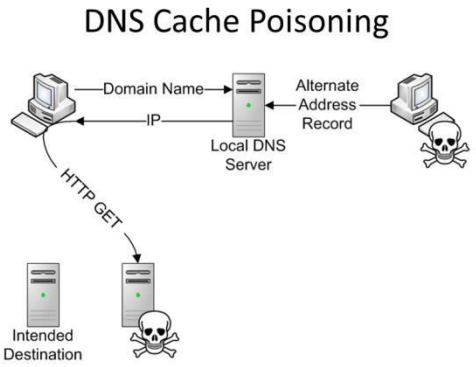

Fig. 1: DNS Cache Posioning Attack Example

DNS Cache Poisoning Technique

To perform a cache poisoning attack, the attacker exploits the flaws in the DNS software. Whenever a server wants to validate the DNS, it validates the DNS responses to ensure that they are from an authoritative source (for example by using DNSSEC); otherwise the server end up caching the incorrect entries locally and serve them to other users that make the same request. But in the case of Cache poisoning attack, the attacker redirects the user from a web site to other web site. The attacker does this with the help of spoofing IP addresses. An attacker spoofs the IP address DNS entries[3] for a target website on a given DNS server and replaces them with the IP address of a server under their control. The attacker then creates files on the server under their control with names matching those on the target server. These files usually contain malicious content, such as computer worms or viruses. A user whose computer has referenced the poisoned DNS server gets tricked into accepting content coming from a non-authentic server and unknowingly downloads the malicious content. This technique is used for phishing attacks. Phishing attack is a attack, that is used to create a fake version of a genuine website that contains personal details such as bank and credit/debit card details.

DNS poisoning attacks are more spreadable. For example, if various Internet service providers are getting their DNS information from the compromised server, the poisoned DNS entry will spread to the Internet service providers and be cached there. Then the entries will spread to home routers and the DNS caches on computers as they look up the DNS entry, receive the incorrect response, and store it. In next section, we discuss about the architecture of DNS technique.

2. *DNS ARCHITECTURE*

Fig. 2 explains the architecture of DNS.

DNS architecture consists of two components. They are

- Name servers
- Resolvers.

Name Servers:

Name servers are databases that stores information stored in it. They are more like a repository. They help in answering

queries by looking up the information they already possess.

Resolver: While, resolvers act as an interface between the clients and the name servers. These resolvers comprise of the algorithms required to find the information queried by the client. These methods can be structured

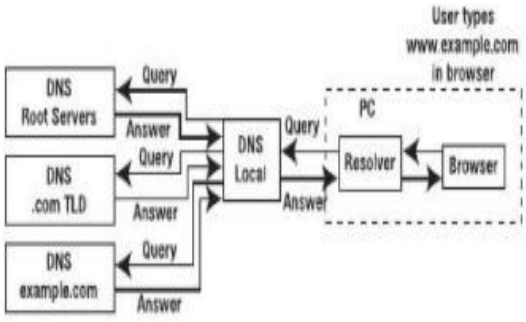

Fig. 2: DNS System Architecture

according to the needs of the environment. The resolver function can either be centralized in one or more special name servers or be separated in hosts such as PCs also known as a stub resolver.

The DNS name[4] space is the naming system on which the DNS is based. It is a hierarchical and logical tree structure with variable-depth where each node in the tree has a label. The root is reserved at the zero-depth label. Currently, the domain name space searching operations are case insensitive.

Mechanisms in DNS

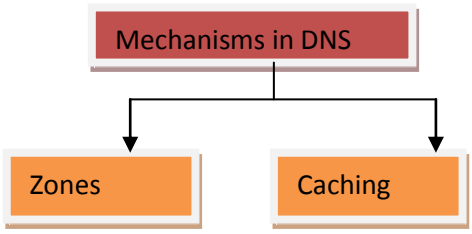

Fig. 3: Mechanisms in DNS

The DNS offers two mechanisms to send and receive data between the ultimate source and destination namely zones and caching. Zones are divided into sections of the system-wide distributed database which belong to specific organizations. The organization of a specific zone is responsible for distributing current copies of the zones to various servers which make the zones available to clients across the internet. Caching is a mechanism in which data acquired in response of the client's request can be stored on the local server against future requests by same or different clients.

One major operation carried out by the name server is to respond to queries from a local or remote resolver or another name server acting on behalf of the name server. The stub resolver is a software library installed on the host or PC which converts a user or application request to a query to the DNS. A typical query means locating the IP address of the Uniform-Resource-Locator or URL inserted by the user. The resolver will identify a locally configured DNS server to perform the queries.

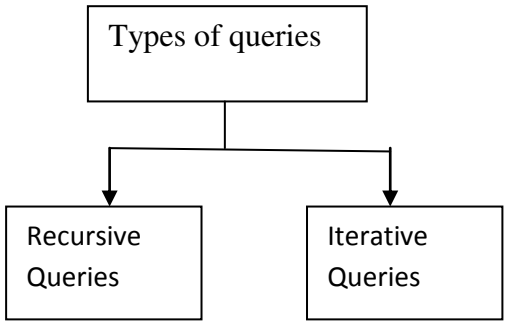

Fig. 4: Type of Queries

Recursive queries are those in which the recipient name server will do all the

working necessary to return the complete response to the query. Responding to a query recursively involves multiple transactions to the name servers and other name server systems. Name serves necessarily support recursive queries.

Iterative queries are those in which if a name server has answered, it will respond otherwise it will return useful information. But it will not make additional queries to other name server systems. Name servers must support iterative queries. The next section we discuss about the case study of the DNS cache poisoning technique.

3. CASE STUDY

Recently there were disturbing reports out of Turkey escalations. The Turkish Government attempted to block the social media like Twitter[5] and YouTube. The Turkish Internet Service Providers (ISPs) hijacked the routes to public DNS servers that provide services back to the citizens. In order to work effectively, the Turkish ISPs perform a "Man-In-The-Middle"(MITM) attack against their citizens and giving them false information. The situations gets highlighted when The Internet Society makes a statement on the subject, explaining its "deep concern" for the situation, along with Chief Internet Technology Officer describing how these moves "represent an attack not just on DNS Infrastructure, but on the global Internet routing system itself."

When the Turkish ISPs started implementing the government's ban on social media, they simply blocked the sites in DNS Servers. Whenever a Turkish Citizen tried to access the social media sites, the ISPs failed to give back the response even if their device queries DNS to get the correct IP address to connect to. But the flaw was that the Turkish Citizen could override the ban by changing their device's DNS settings to point to open public DNS Resolvers operated by Google. Then the Turkish ISPs started blocking the addresses for Google Public DNS Servers and other similar services in order to engage in the typical kind of "what-a-mole" game with their citizens where they found a new way to get around the censorship and tried to close them down.

After few days, the Turkish ISPs started taking it to a whole new level by hijacking the routes of Border Gateway Protocol (BGP) and pretending as the DNS Severs from the Public Google DNS itself and other similar services. That is, the devices operated by Turkish Citizens who tried to change the DNS settings to Public Google DNS Servers were getting back to the requests from the Turkish ISPs. Unfortunately, the Turkish citizens were receiving wrong answers from the Turkish ISPs instead of what they indented to get (YouTube or Twitter).

Often the DNS Servers are compared to a Phone Book as it serves the computer the address of a correct, quick, secure server it's looking for; the same way when someone tries to look up for a phone number in a phone book. But someone can change out the phone book with another one, which seems pretty much the same as the former, except that the listings for a few contacts showed up wrong phone numbers. That's exactly what happened in this case. The Turkish ISPs

set the servers up that masquerade as Public Google DNS servers.

As a result of it, the Turkish ISPs started "advertising" the more specific (wrong) route for the Public Google DNS services. More specifically, The normal Public Google DNS server settings is at "8.8.8.0/24", instead, they shared the route "8.8.8.8/32" which redirected to their own network. In BGP, a network device tries to connect a given IP address by selecting a specific route. But in this case, all the routers on the networks were connected to Turkish ISPs in their specified false route for Public Google DNS services.

In this way the Turkish ISPs were delivering false DNS information's to the Turkish citizens. This allowed the Turkish ISPs to extract personal information's of citizens very easily and this also opened new opportunities for the crackers to crack the personal information of the citizens of Turkey.

4. *SOLUTION TO PREVENT CACHE POISONING ATTACKS*

One of the most readily available defences against DNS attacks is to secure the attack points on the network infrastructure. There should be proper use of firewalls, and patches of known vulnerabilities should be applied periodically.

A specific technique to foil DNS attacks involves randomizing source ports on the DNS requester[6]. When this technique is applied, a DNS packet that does not come from a trusted source (attacker) will have a approximate 1/216 chance of going to the victim and the requester will know that this is an attack and the packets are discarded.

DNS resolution software can be implemented which acts as a poll to multiple other DNS servers in the event that the resolver running the software does not have information on a particular DNS server. Through this method, an DNS servers can be known to be malicious and its effect is ignored. But if the attacker gains control over more than half of the servers in DNS region, then there might be chances of any security foil for DNS servers.

DNS servers should be less trusting of information passed to them by other DNS severs. Moreover, they should ignore any DNS records passed back which are not directly relevant to the query. For example, some versions such as BIND 9.5.0-P1 perform the above checks.

DNS servers can use a combination of crypto graphical techniques to secure random numbers for selecting both the source ports. A 16-bit cryptographic technique can greatly reduce DNS attacks.

However, various network devices perform Network Address Translation (NAT), or Port Address Translation in specific often rewrites the source ports in order to track the connection state. During the process, the PAT devices remove the source port randomness implemented by the name server and stub resolver.

Cryptographic digital[7] signed with a trusted key certificate is used by the Secure DNS (DNSSEC) to determine the authenticity of data. DNSSEC was employed in Internet root zone servers only. But, even DNSEC can still provide

fake data without application-layer cryptography. Mitigation to this can be done at transport or application layers by performing end-to-end validation once a connection is established. Transport layer security and digital signatures can be used to counter the DNS attacks. For instance, using of HTTPS connection in which the client checks whether the server's digital certificate is valid and belongs to a particular expected website's owner. A similar kind of a system can be witnessed in the secure shell remote login program which checks digital certificates at endpoints proceeding with the session. The system can embed a copy of the signing certificate locally and validate the signature stored in the software update against the embedded certificate for the software's that download update automatically.

Organisations such as Dell and TCP Wave have Intelligent Analyst Cache Application who have watchdogs which ensure that the DNS processes do not get a cache poison by predefining the roots in the watchdogs. In this way mitigation of DNS Any cast cache poisoning attacks from malicious users can be done through source randomization via BIND backed up by a non-BIND DNS Server Software with intelligence blended into the BGP routing protocol.

In Intranet DNS poisoning there is a DNS poisoning attack over a LAN due to ARP poisoning man-in-the-middle attack. The counter measures are to use of static ARP and IP table, switched LAN, SSH encryption. Usage of sniff detection tools and tunnelled connection which support IPSec is recommended.

In remote DNS[8][9][10] poisoning occur due to negligence of victim to unknown files. It's due to opening of suspicious files and archives; the system is compromised by Trojans and Trojan vectoring methods.

DNS servers are maintained by local and primary DNS servers. Hence, all the DNS servers should be audited regularly to counter flaws in the security. Its due to a small vulnerability can lead to breach in security of DNS servers thus leading to DNS attacks. For usage of DNS server, one must provide extra layer of security such as installing DNS with bind-chroot package.

5. *CONCLUSION*

In this paper, we have studied about the DNS cache poisoning attack in detail. Our paper suggests the key contribution provided in literature to secure DNS cache poisoning attack. Our paper provides the DNS system architecture which describes the deployment of DNS in an organization. Next, our paper suggests some case study related issues which are proposed in turkey. Finally our paper suggests some solutions to avoid DNS cache poisoning attack. Our paper provides a roadmap in the state of art to know about cache poisoning attack. Our future work will provide solutions to prevent this DNS cache poisoning attack.

REFERENCES

[1] Bechtsoudis and Sklavos, "Aimig at higher Network Security through Extensive Penetration Testing", IEEE Latin America Transactions, Vol. 10, No. 3, April 2012.

[2] Kshetri, N., "The simple economics of cybercrimes", IEEE Security and Privacy (2006), Volume: 4, Issue: 1.

[3] Schonwalder, J., Pras, A., Harvan, M., Schippers, J. and Van de Meent, R., "SNMP Traffic Analysis: Approaches, Tools, and First Results", IEEE Integrated Network Management, 2007.

[4] Ansari, S., Rajeev, S.G. and Chandrashekar, H.S., "Packet sniffing: a brief introduction", IEEE Potentials (2003), Volume: 21, Issue: 5.

[5] Long, M., Chwan-Hwa Wu, Hung and J.Y., "Denial of service attacks on network-based control systems: impact and mitigation", IEEE Transactions on Industrial Informatics (2005), Volume: 1, Issue: 2.

[6] Bishop, M., "About Penetration Testing", IEEE Security and Privacy (2007), Volume: 5, Issue: 6.

[7] Haya Shulman and Michael Waidner, "Towards Forensic Analysis of Attacks with DNSSEC", IEEE Security and Privacy 2014

[8] A. Herzberg and H. Shulman, "Dnssec: Interoperability challenges and transition mechanisms," in Availability, Reliability and Security (ARES), 2013 Eighth International Conference on. IEEE, 2013, pp. 398–405.

[9] Lihua Yuan, "DoX: A Peer-to-Peer Antidote for DNS Cache Poisoning Attacks", UC Davis.

[10] Paul V. Mockapetris, "Development of the Domain Name System", USC Information Sciences Institute, Marina del Rey, California