Spoofing Attacks of Domain Name System Internet

Simar Preet Singh1, A Raman Maini2
Research Scholar1
Associate Professor2
University College of Engineering,
Punjabi University Patiala, Punjab, India - 147002

ABSTRACT
The Domain Name System (DNS) is a hierarchical naming system that is built on a distributed database for computers, services, or any other resource connected to the Internet or a private network. It translates the domain names meaningful to humans into the numerical identifiers associated with the networking equipment for the purpose of locating and addressing these devices worldwide [1]. The job of a DNS is to convert the human readable addresses entered on the address bar of the browser into machine readable IP addresses. DNS spoofing is a term that refers to the action of answering a DNS request that was intended for another server (a ―reall DNS server). This arrangement can be in a server-server exchange (a DNS server asks another for a mapping) or in a client-server dialog (when a client asks a DNS server for a mapping).
In the last many years, several security flaws have been discovered in the protocol and its specific implementations. This research paper gives an overview over the different threats to the DNS and their attack targets. We have discussed the various DNS Spoofing Attacks without IP Spoofing and DNS Spoofing Attacks with IP Spoofing and discuss their success chances and possible countermeasures. Finally, as a case study, DNS spoofing attack model is constructed and the availability of the attacked system is evaluated. The proposed approach can be used for other kinds of attacks and other types of systems, networks and applications.

Keywords
Network Security, E-mail Security, Threats, Keyloggers.

1. INTRODUCTION
Humans can‘t think like the computers. Humans just can‘t Remember dozens of IP addresses. They need easy-to remember names to locate their mail server or their favorite web pages. To make our lives on the Internet easy, DNS was Therefore invented. DNS stands for Domain Name ServiceAll in all, what it does is translate a host‘s name into its IP address [3]

1.1 Iterative and Recursive Queries
Before going on with the spoofing, we need to understand the difference between a recursive and an iterative DNS query. When a host queries a DNS server, it can choose to use a recursive query, in that case the client wants the answer, or an error message that what it‘s looking for could not be found anywhere. We see that the queried host must do whatever it takes to find the answer: query other servers until it gets the information, or until the name query fails. When a host queries a DNS server in the iterative way: it basically asks the server for an answer IF the server knows it (if it has the answer in its). If it does not, then the client will receive a_referral‘which is the name of the server that may have the answer (a authoritative server at a lower level in the Hierarchical structure as we talked about)[2]. Recursive Queries are usually made by client hosts so that they don‘t Have to take care of the whole search process, whereas local DNS servers usually make iterative requests.

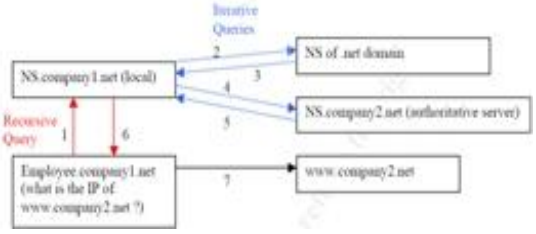

Fig. 1: shows the particularities of two different kinds of requests

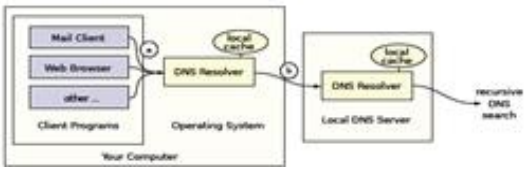

Fig. 2: Typical DNS resolution

To reduce the amount of traffic in the resolving procedures of domain names, the applications on a user's computer do not directly communicate with the DNS servers on the internet. The operating system features a resolver component which handles the DNS resolution for the applications, typically by forwarding the requests to a DNS server in the local network as shown in Figure 2.

This server resolves the recursive query by iterative queries in the DNS, starting at the root servers, down to the Authoritative DNS server for requested domain name (shown in Figure 3).

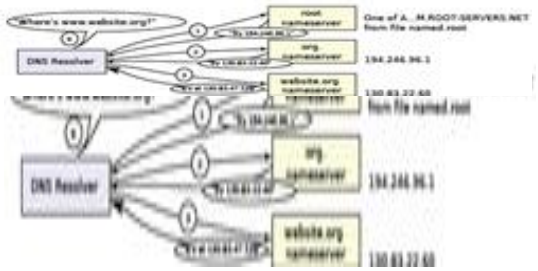

Fig. 3: Resolving a recursive query

The resolver as well as DNS server reduces the traffic by caching previous resolutions. A typical resolution request is processed as follows:
For example the browser needs to know the address of www.website.org. A resolution request is sent to the local resolver. The local resolver looks up, whether the resolution of www.website.org is already in its cache. In this case the query is answered directly[4]. Otherwise the resolver sends a recursive resolution request to the local DNS Server. If the desired information is not in the local DNS server's cache it starts an iterative lookup of the domain name as follows: (1) The lookup for www.website.org starts at one of the 13 root servers of the DNS. Every DNS server holds a list of these servers. The root server delegates to one of the DNS servers for the de top-level domain (TLD).
(2) In the next step the local DNS server queries this server for the resolution of www.website.org. The DNS server of the org-TLD delegates to one of the DNS servers of the website.org domain, e.g. name.server.website.org.
(3) Lastly the local DNS server queries this name server, which is authoritative for the whole website.org and receives the IP address of
www.website.org. At last the IP address is returned in an answer to the resolver of the users computer.

1.2 Attack Classification

There are many ways in which DNS can be attacked. There are the following five target groups:
(1) Local Network - The local network grouping includes the customer's host, the physical LAN, any proxy servers and egress firewalls. In addition, if the customer is located within a business environment, local DNS services may also
be included.
(2) ISP DNS Services - This group includes all the DNS servers used by the customer, located on the Internet, used for DNS resolution. It includes ISP DNS servers that cache lookup results as well as and resolving services.
(3) Global DNS Services - This group includes all the globally managed services used as part of the resolving process to identify the authoritative name servers for a domain. It includes all Root and TLD servers.
(4) Corporate Domain - This group includes all the services typically owned by a corporate entity to do carryout the IP address resolution of named hosts. As such it includes the authoritative name services for their domain, and any other final delegation processes.
(5) Related Resolution Services - This group includes
services not directly related to the DNS lookup process,
but which have a substantial effect on the resolution of

different DNS attacks as follows:

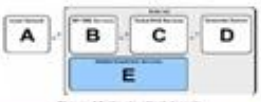

Figure 12: Key to attack targets

Attack Vector	Target Group				
	A	B	C	D	E
Human Factors					
The insider edge	Yes	Yes	Pos.	Yes	
Local Host and Local Network Attacks					
Modification of lookup processes	Yes				
Traffic observation and modification	Yes	Pos.			
Man-in-the-middle attacks	Yes				
Domain Registration Attacks					
Domain hijacking					Yes
Similar domain name registrations					Yes
Botnet name server registration					Yes
Domain Configuration Attacks					
DNS wildcards				Yes	
Poorly managed DNS servers		Yes		Yes	
DNS Spoofing					
DNS cache poisoning	Yes	Yes			
DNS ID spoofing with sniffing	Yes				
DNS ID spoofing without sniffing	Pos.	Yes			
The birthday attack	Pos.	Yes			
The "New DNS" Attacks					
Page rank escalation					Yes

Fig. 4: Key to attack targets

1.3 DNS Spoofing

DNS spoofing is a term used when a DNS server accepts and uses incorrect information from a host that has no authority giving that information. DNS spoofing is in fact malicious cache poisoning where forged data is placed in the cache of the name servers. According to the most recent "Domain Health Survey" (Feb 2003), a third of all DNS servers on the Internet are vulnerable to spoofing. Spoofing attacks can cause serious security problems for DNS servers vulnerable to such attacks, for example causing users to be directed to wrong Internet sites or e-mail being routed to non-authorized mail servers.

The Attack

Operating normally, a customer can expect to query their DNS server to discover the IP address of the named host they wish to connect to. The following diagram reflects this process.

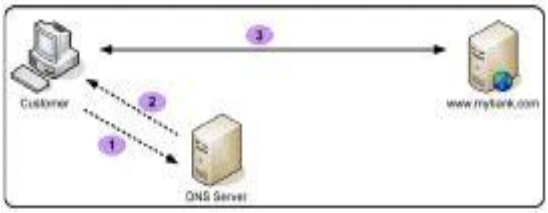

Fig. 5: The normal DNS resolution process

(1) The customer queries the DNS server - "What is the IP address of www.mybank.com?"
(2) The DNS responds to the customer query with "The IP address of www.mybank.com is 150.10.1.21"
(3) The Customer then connects to the host at 150.10.1.21 - expecting it to be www.mybank.com.

However, with a successful DNS spoofing attack, the process has been altered. The following diagram reflects this process[6].

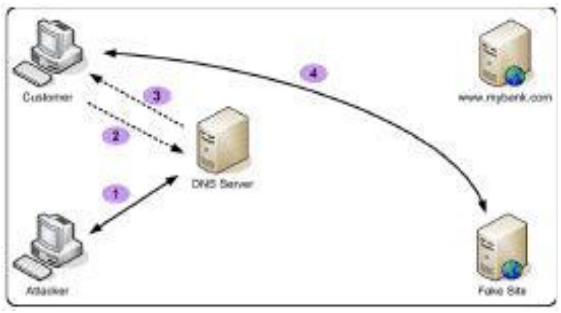

Fig. 6: The DNS resolution process having fallen victim to a DNS spoofing attack

(1) The attacker targets the DNS service used by the customer and adds/alters the entry for www.mybank.com — changing the stored IP address from 150.10.1.21 to the attacker‘s fake site IP address (200.1.1.10).

(2) The customer queries the DNS server —

"What is the IP address of www.mybank.com?"

(3) The DNS responds to the customer query with "The IP address of www.mybank.com is 200.1.1.10" — not the real IP address.

(4) The Customer then connects to the host at 200.1.1.10 — expecting it to be www.mybank.com, but in fact reaching the attackers fake site

1.4 DNS Spoofing Attacks without IP Spoofing

In this Section spoofing attacks will be discussed, which need no IP spoofing and are possible because the lacking authentication and ambiguity of protocol description[7].

1.4.1 DNS Cache Poisoning

In this attack the attacker abuses caching vulnerabilities within the DNS server to add multiple resolution entries for hosts not originally asked for and is not authorised to provide. While most new DNS service implementations are not vulnerable to cache poisoning, there are still a large number of vulnerable DNS servers that are.

The process in which a DNS server may have its cache poisoned can be explained in the following diagram-abuses caching While most authorized

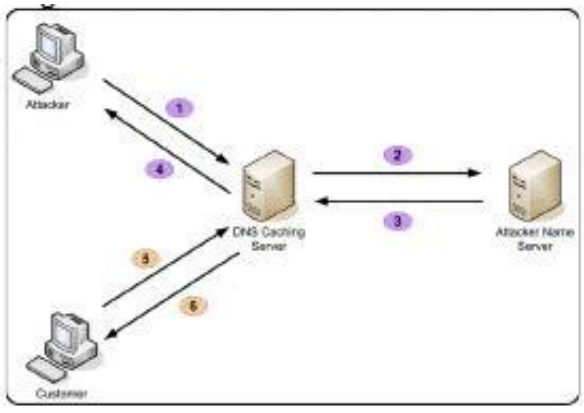

Figure 7: The DNS cache poisoning process

(1) The attacker queries the DNS server for the IP address for a host that is managed by a name server owned by the attacker — "What is the IP address of www.attackerowned.com?"

(2) The DNS Caching server does not have a cached entry for www.attackerowned.com and must resolve the IP address by querying the authoritative name server for the attackerowned.com domain. This authoritative name server belongs to the attacker.

(3) The attackers name server informs the DNS caching server that the IP address of www.attackerowned.com is 200.1.1.10. In addition, the attackers name server also includes additional (faked) resolution records such as:

a. www.mybank.com is 200.1.1.11 b.mail.mybank.com is 200.1.1.11

c. secure.mybank.com is 200.1.1.11

(4) The DNS caching server responds to the attacker's original query with — "The IP address of www.attackerowned.com is 200.1.1.10." This result, along with the extra resolution records, is cached by the DNS server for a period equivalent to the TTL supplied by the attackers name server.

(5) At a later date, an ordinary customer who also uses this DNS caching server queries it for the IP address of

www.mybank.com — "What is the IP address of www.mybank.com?"

(6) The corrupted DNS caching server responds to the customer query by supplied the previously cached (and fake) answer — "The IP address of www.mybank.com is 200.1.1.11" — instead of the real 150.10.1.21 address.

For instance, In July 1997 Eugene Kashpureff of AlterNIC used a program to "poison" the caches of major name servers around the world. This caused traffic originally destined for www.internic.net's address to go to the IP address of the AlterNIC web server. No attempt was made to disguise the attack, and customers who tried to reach www.internic.net were confronted with the AlterNIC website.

1.5 DNS Spoofing Attacks with IP Spoofing

The attacks in this Section all base on sending spoofed replies before the legitimate reply reaches the asking computer[8]. If the legitimate reply arrives after the faked reply, it is discarded.

1.5.1 Sequential IDs

The TID is the main security mechanism in the DNS protocol and should be randomized to complicate attacks. This made an attack easy, especially on DNS servers with recursion enabled. The attacker just needed to trigger a request to a name server under his control, to determine the current TID[11]. For a successful cache poisoning he could send in the attack just few spoofed packets just with the following TIDs. With a random TID the chances to send the right TID are only 1/65535 (1/216) per packet as the TID has a range of 16bit. For a 50% chance on a successful attack the attacker needs to send on time more than 3MB of data(at a packet size of 100 Bytes) to the victim, before the legitimate answer arrives. If the server is using also random source ports this chance is lowered even more by 216-1024, as the attacker has also to guess the right port in the range of 1025 to 65535. Overall the chance with random TID and port is

1/4.227.858.432 (< 1 / 0, 98*232). This increases the amount of data to be sent in time for a 50% chance to nearly 200GB.

It exploits a weakness discovered in 2002 relating to the fact that the most popular DNS implementation (BIND) would send multiple simultaneous recursive queries for the same IP address (now fixed in the latest versions of the software). This repetitive behavior means that a "Birthday Paradox" could be used to mathematically increase the speed and probability of a successful attack by reducing the number of spoofed guesses of the DNS transaction ID from tens of thousands down to a few hundred.

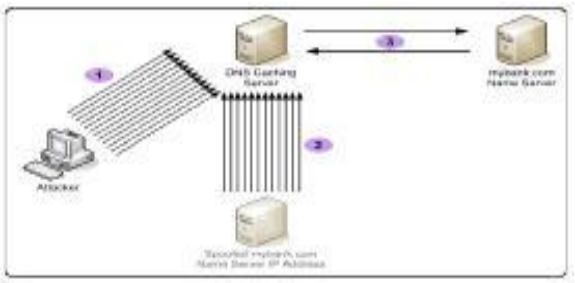

Fig 8: The DNS Birthday Attack

1.5.2 The Birthday Attack

It exploits a weakness discovered in 2002 relating to the fact that the most popular DNS implementation (BIND) would send multiple simultaneous recursive queries for the same IP address (now fixed in the latest versions of the software). This repetitive behavior means that a "Birthday Paradox" could be used to mathematically increase the speed and probability of a successful attack by reducing the number of spoofed guesses of the DNS transaction ID from tens of thousands down to a few hundred.

In the figure above, the birthday attack is carried out as follows

(1) The attacker launches repeated requests to the DNS caching server asking "What is the IP address of www.mybank.com" as fast as possible.

(2) Simultaneously, the attacker also sends repeated spoofed responses using different DNS transaction ID's stating that "The IP address of www.mybank.com is 200.10.1.11".

(3) For each request from the attacker in (1), the DNS server tries to resolve the IP address for www.mybank.com by querying the authoritative mybank.com name server — typically using a different DNS transaction ID for each request. Based upon the mathematical properties of the Birthday Paradox, there is a higher probability that the attacker can "guess" a correct DNS transaction ID (thereby "answering" the DNS servers query) faster than the real name server can respond.

To further increase the odds of the attacker supplying a correct DNS transaction ID with the spoofed message, the attacker could target the authoritative name server with other requests or denial of service techniques to slow down its response to the DNS caching server.

1.5.3 PRNG weakness

other internet protocols) is the weakness of many current Pseudo Random Number Generators (PRNG) against Phase Space analysis. Michal Zalewski [21] described this weakness first in 2001 for the random TCP sequence numbers of different operating systems, but further investigation by himself and Steward showed that the random transaction number generation of most DNS servers is also vulnerable. Especially the PNRG employed in BIND8 is predictable. With analysis of 100000 TIDs and knowledge of the previous TIDs is the following TID foreseeable with 100% certainity. This would make a spoofing with just one packet possible. BIND

4, using the same code base as BIND 8, shares this weakness. BIND 9 does not come with an own PRNG, but uses the one of the operating system. On Linux 2.4.19 the chance of a successful attack with 5000 packets on a BIND 9 (no port randomizing) is at 20%, compared to 7,6% with a normal spoofing attack[2]. DJBDNS has a slighly weaker PRNG with 30% success chance at 5000 packets to guess the TID, but is still more secure due to port randomization by default. The Microsoft DNS Server is not discussed in detail by Zalewski, but at least the NT4 pre-SP6 PRNG seems quite weak. Also some of the resolver libraries like MS as well as Linux (glibc2.1.9) employ a weak PRNG.

1.6 Case Study: Modeling and Evaluation of DNS Spoofing

In this section, as a case study, a spoofing process is modeled and evaluated by a simulation model[10].

1.6.1Simulation Model of the Normal Operation

The model shown in Fig. 10 consists of two clients in the left, a path combiner and an output switch as a DNS in the middle, and two servers as destination in the right. These clients generate packets with specified length and send them to a specified target server nominated in its packet. Packets are generated by Time-Based Entity

Generator with mean=50 and 200 from SimEvents library. Packets length and destinations are generated by Random Number Event-Based Generator with uniform distribution. Destination and length can be a number between 1 and 2, 6 and 10, respectively. Destinations are target servers. To set the specification of each client, Set Attribute blocks are used[9]. First attribute, A1 or Source is source packet generator for 1st client and is set to one and for the 2nd one, it is set to two. Second attribute, A2 or Length is packet length that its value is specified from random number generator block connected to set attribute block. Packets after generating are routed by source router to the destination server that is specified in set attribute block of its own generator. When packets were generated, they were stored in their limited-capacity queue in order to hold and guide packets. Capacity of both queues is 25 packets and no preemption is defined for passing packets from queues.

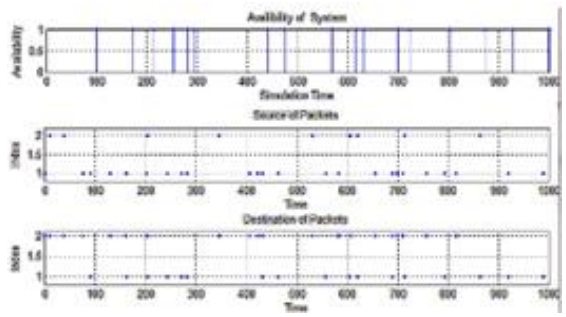

Fig.9: Simulation results for 1000 run-times

To simulate DNS, we use path combiner and output switch, which receives packet from input ports, find the right destination and leads them to target servers based on their destination. For each packet, destination can be one of the 1 or 2 servers. Service time of servers is adjusted to 10. After processing packets in server, they have been led to Entity Sink or used to measure and report system parameters. First, the server output is used for measuring A(t) of the first server. These simulation outputs are four axes. One spots generated packet from two clients, the other one shows received packets and the two lasts demonstrates availability of servers. As shown in Fig. 4, the first server is busy in 100, 180… 210… 1000 simulation times, so it is idle in 10, 20… 70… 990. The second server operates similarly.

1.6.2Simulation Model of the DNS Spoofing Attack

To illustrating spoofing in our model, we suppose that the second destination server is an attacker who wants to spoof the DNS system in order to lead packets of the first source to him/her. To simulate security failure and consequently a chance for spoofing DNS, we injected a security failure subsystem to our model. The security failure subsystem consists of (i) a Time-based Entity Generator that generates security failure entities with exponential distribution mean=1000 (in attack time), (ii) a security failure repair server which receives the repair time from a Random Number Generator with uniform distribution between 10 and 70.

DNS System is simulated by Path Combiner and Output Switch blocks[5]. Path Combiner receives packets from input port and then sends them to destination by using Output switch. In the meantime, DNS is maybe spoofed and led packets to attacker.

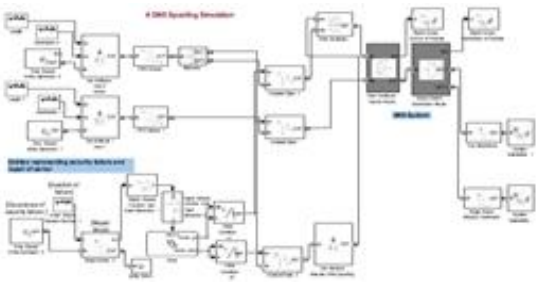

Fig. 10: DNS spoofing attack simulation model

Spoofing. First, in normal state, generating packets mean value of first source (Attacker Target) is 50, and then by decreasing the mean of generating security failure entity generator, attack happens and packets from first server are sent to attacker server so the availability of attacker server in all of the simulation time becomes one. Moreover, other outputs of this simulation illustrating the packets departed from the sources and destinations are shown. After 1000 run times of simulation and setting the security failure mean value to 8000, the output will be as Fig. 12. Now, we decrease the mean value to 1000 and run simulation for 1000 times again. Considering that during the total simulation time, the attacker took the control of victim server and packets are sent to attacker server instead of proper target. The results are shown in Fig. 12.

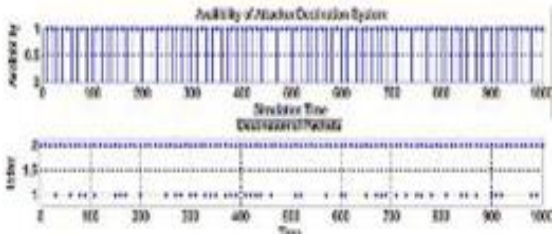

Fig. 11: Simulation results for 1000 run-times and security failure mean=1000

2. CONCLUSION

The most promising DNS spoofing attacks for an external attacker are Cache Poisoning, the exploitation of sequential IDs and birthday attacks. These vulnerabilities are fixed since several years, with exception of the reappearing Cache Poisoning in certain configurations with forwarding. The security mechanisms of DNS against spoofed packets, namely TID and source port, are quite weak compared to cryptographic methods. To guess the TID only an average of 32000 packets is needed. Nevertheless, for an external attacker it is quite hard to send the right spoofed packet before the answer of the legitimate server arrives. Furthermore, on a server without external recursion the correct source port is difficult to determine. For a server with port randomizing guessing is needed. This lowers the attack feasibility by nearly 216.

The exploitation of PRNG weaknesses requires a-priory analysis of a huge number of server queries and seems to be a more theoretical weakness.

As a case study, we constructed a simulation model for DNS spoofing attack. First, clients send packets to servers and the attacker as a client intrudes to the system and spoofs victim DNS and receives packets. During the simulation, the availability measure, A(t), of the server is measured.

As a future work, we can model and evaluate the availability measure in large and sophisticated computer and communication systems to see the potential benefits of the proposed simulation methodology

3. REFERENCES

[1] Wikipedia: Domain name system (2006)

[2] http://en.wikipedia.org/wiki/Dns.

[3] DNS and Bind", O'Reilly, 2001

[4] The Phishing Guide", Gunter 01/mann, 2004.

[5] Allen, R., Larson, M., Liu, C.: DNS on Windows Server 2003. O'Reilly (2003) ISBN: 0-596-00562-8.

[6] Salamon, A.: (Dns related rfcs) http://www.dns.net/dnsrd/rfc/.

[7] Center, S.I.S.: Dns cache poisoning update 7.4.2005

[8] http://isc.sans.org/diary.php?storyid=502.

[9] "Security Best Practice: Host Naming and URL Conventions", Gunter 01/mann, 2005

[10] DNS Spoofing Techniques http:// www.

[11] securesphere.net/download/papers/dnsspoof.htm

[12] Combettes, P.L, ―The Foundation of Set Theoretic Estimation‖,

[13] S. Rose and A. Nakassis. ―Minimizing Information Leakage in the DNS‖ IEEE Network Magne vol. 22 no. 2 April 2008.

[14] J. Damas and F. Neves, ―Preventing Use of Recursive Nameservers in Reflector Attacks‖, BCP 140, RFC 5358. October 2008.