DNS

- **Domain Name System (DNS):** maps host names to addresses
  - Large distributed database

- **Critical Internet infrastructure**
  - Control of DNS allows attacker to impersonate any site (in the absence of end-to-end host authentication using SSH or TLS)

- Target of attacks
DNS

- **DNS Record:** Unit of information in DNS
  - **Type:** type of data it contains
  - **TTL:** time to live

- **A record:** IP address for a host name

- **NS record:** Name server to contact for a domain

- **MX record:** SMTP (mail) server for domain
A DNS server has a set of records for which it is the authoritative source

`; Got answer: `nn`->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30439nn`nn`flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 6nn`
n`-; QUESTION SECTION:nn`nn`; bob.ucsd.edu. IN A

`; ANSWER SECTION:n`nn`bob.ucsd.edu. 3600 IN A 132.239.80.176

`; AUTHORITY SECTION:n`nn`ucsd.edu. 3600 IN NS ns0.ucsd.edu.nnn`nn`ucsd.edu. 3600 IN NS ns1.ucsd.edu.nnn`nn`ucsd.edu. 3600 IN NS ns2.ucsd.edu.
DNS Roles

- **Authoritative server**: provides authoritative information for a set of domains
  - Does not handle queries about other domains

- **Recursive resolver**: provides recursive resolution of a domain to return requested record to client
  - Handles queries about all domains

- Same protocol for both types of servers
  - Distinction is in intended purpose only
Security of DNS

- Basic DNS uses UDP without any authentication
- Man-in-the-middle attacks are possible
  - Forging response to observed query is trivial
- Off-path attacks require guessing two parameters
  - Query ID from response (16 bits)
  - Source port (approx. 15 bits)
DNS Packet

DNS Header

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>QR</td>
<td>Opcode</td>
<td>DNS Flags</td>
<td>RCode</td>
</tr>
<tr>
<td>Total Questions</td>
<td>Total Answers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Authority Resource Records</td>
<td>Total Additional Resource Records</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DNS Header Format
Created by Troy Jessup - http://www.troyjessup.com
Caching

- Recursive resolvers cache records to avoid repeating recursive resolution process for each query

- Lifetime of record determined by record TTL
  - Could also be evicted from cache because of limited memory

- Injecting spoofed records into a resolver’s cache is called DNS cache poisoning
  - No protocol-defined way for client to refresh cached record
Forging DNS Replies

- For performance reasons, some DNS resolvers (e.g. BIND) re-used the same socket for all queries
  - **If source port is same:** can be determined by attacker by having recursive resolver query attacker-controlled authoritative server

- 16-bit query ID now only thing need to guess
  - Non-random query ID generators may make it easier
Naive Cache Poisoning

1. Query resolver for target domain www.ucsd.edu
   - If name already in cache, nothing you can do, wait for it to expire
   - Some resolvers may only serve particular hosts, need insider

2. Recursive resolver issues query to .edu authoritative server to get authoritative server for ucsd.edu

3. Generate a flood of forged replies with different query IDs appearing to be from .edu authoritative server
   - Some chance of guessing query ID and winning packet race
Naive Cache Poisoning

- Requires guessing query ID and source port
- Throttled by TTL (discussed later)
Additional Record Injection

- DNS query results include Additional Records section
  - Provide records for anticipated next resolution step
- Early servers accepted and cached all additional records
  - What is wrong with this?
Early Attack Strategy


2. The user's computer asks the targeted name server to translate www.BadGuysAreUs.com into an IP address.

3. The targeted name server has not cached the address, so the query is routed through a root name server, a .com name server, and finally the BadGuysAreUs.com name server.

4. The BadGuysAreUs name server responds with an IP address but adds a false IP address for a completely different Web site, www.paypal.com.

5. The targeted name server stores the false IP address for paypal.com.

6. When people using this name server attempt to go to www.paypal.com, they are directed to a Web site that looks like PayPal's but works only to harvest their user names and passwords.

Additional Record Injection

- DNS query results include Additional Records section
  - Provide records for anticipated next resolution step
- Early servers accepted and cached all additional records
  - What is wrong with this?
- Can we just stop using additional section?
  - Only accept answers from authoritative servers?
DNS Glue Records

- Can we just stop using additional section?
  - Only accept answers from authoritative servers?

- **Glue records**: non-authoritative records necessary to contact next hop in resolution chain
  - Necessary given current design of DNS
What should be the policy for accepting additional records?
Bailiwick Rules

- **Bailiwick rule**: defines what response records a recursive resolver will accept

- **Bailiwick**: set of domains about which a server is has direct or indirect authority to speak
  - Bailiwick determined by the initiator of query

- **Answer should be relevant** (in response to request)

- **Answer should be in bailiwick**
Bailiwick Checking Rule from BIND

● Authorities must be for queried domain
  ○ \texttt{ns0.csd.edu} accepted as authority for \texttt{ucsd.edu} only when initiating query was for subdomain of \texttt{ucsd.edu}

● Additional records must be \textit{in bailiwick} for query
  ○ A record for \texttt{ns0.ucsd.edu} accepted because \texttt{edu} server has indirect authority over \texttt{ns0.ucsd.edu}
DNS should have been designed with addresses, not names, in NS records and MX records. The “additional section” of DNS responses should have been eliminated. RFC 1035 observes correctly that NS indirection and MX indirection “insure [sic] consistency” of addresses; however, this indirection should have been handled by the server, not the client.

— Daniel J. Bernstein

source: http://cr.yp.to/djbdns/notes.html
Kaminsky’s Attack

- Naive attacks on DNS are throttled by TTL
- Each result of query cached for TTL duration
  - Future attack queries answered from cache
- **Attack**: inject forged *additional records* when non-existent domains are queried
- Bypasses TTL throttling
Kaminsky DNS Attack

1. An attacker issues a DNS query for the nonexistent aaa.paypal.com.

2. The attacker immediately sends fake responses to his own query, each containing a different query ID number and a false IP address for www.paypal.com.

3. The attacker repeats steps 1 and 2 using different prefixes: aab.paypal.com, aac.paypal.com, and so on, until the targeted server finally accepts a spoofed response. The spoofed response “poisons” the cache of the name server with a false address for www.paypal.com.

4. Users accessing www.paypal.com through the poisoned name server are directed instead to a Web site that looks like PayPal’s but works only to harvest their user names and passwords.
Goal: Increase entropy in DNS

Idea: Vary capitalization of queried domain
  • DNS is case insensitive, so this is okay

DNS server’s response must use same capitalization
  • All known servers happen to do this

One additional bit of entropy per letter
  • Attack must guess capitalization also
DNSSEC

- Cryptographically sign DNS records
- Chain of trust form root to subdomains, etc.
- DNSKEY record: a public key
- RRSIG record: digital signature of data
- NSEC record: negative response
- DS record: delegation record