DNS Security

CMSC 426/626 - Fall 2014

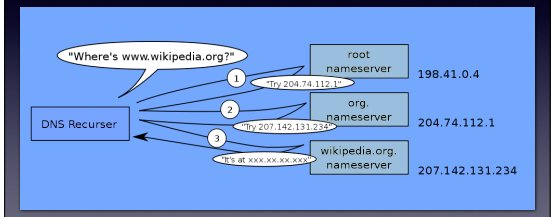
Overview

- DNS Overview
- DNS Attacks
- DNSSEC

DNS Review

- The *Domain Name System* (DNS) resolves domain names to IP addresses
- Hierarchical system of *name servers*
- *Root name servers* store addresses of *authoritative name servers* for their subdomains
- Subdomain servers store addresses of hosts in their domain and of other authoritative servers

A DNS Query



DNS graphics courtesy *LionKimbro* (Wikipedia); Public Domain

DNS Packets

- DNS is (usually) sent in UDP
  - Header includes a *Query Identifier* or *Transaction Identifier* - a 16-bit value
  - Query consists of a domain name and type of record requested
  - Answer is a sequence of DNS records

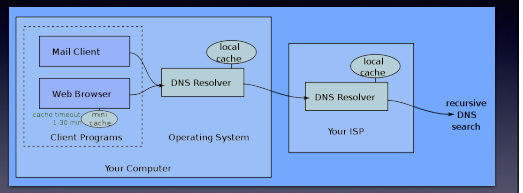
DNS Records

- Consists of the following fields
  - *Name* - full domain name
  - *Type* (2 bytes)
    - "A" for standard address resolution
    - "NS" for name server info
    - "MX" for email resolution info
    - etc.

DNS Records (cont)

- *Class* (2 bytes) - broad categories, e.g. "IN" for Internet domains
- *TTL* (4 bytes) - how long a record will remain valid (in seconds)
- *RDLENGTH* (2 bytes) - length of data
- *RDATA* (variable length) - requested results, e.g. IP addresses when Type is "A"

DNS Caching



- Name servers retain recently received DNS records; period of validity determined by TTL

Exploring DNS

- Windows
  - `nslookup`
  - `ipconfig /displaydns`
- Linux
  - `nslookup`
  - `dig`

Dig

```
% dig @130.85.24.201 umbc.edu MX
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.23.rc1.el6_5.1 <<>> @130.85.24.201 umbc.edu MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4915
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 11, ADDITIONAL: 10

;; QUESTION SECTION:
;umbc.edu.                      IN  MX

;; ANSWER SECTION:
umbc.edu.               9162    IN  MX  10 mxin.umbc.edu.

;; AUTHORITY SECTION:
umbc.edu.               7313    IN  NS  dnsexternal1.umbc.edu.
umbc.edu.               7313    IN  NS  UMBC4.umbc.edu.
    [ 9 lines deleted ]

;; ADDITIONAL SECTION:
UMBC5.umbc.edu.         7316    IN  A   130.85.1.5
umbc10.umbc.edu.        7316    IN  A   130.85.1.10
dnsexternal2.umbc.edu.  7316    IN  A   130.85.1.11
    [ 7 lines deleted ]

;; Query time: 1 msec
;; SERVER: 130.85.24.201#53(130.85.24.201)
;; WHEN: Tue Nov 11 10:14:13 2014
;; MSG SIZE  rcvd: 502
```

Cache Poisoning

- Attempts to create false entries in the DNS cache of a local nameserver
- Attacker issues many DNS requests to local server; multiple queries sent to higher-level nameservers
- Attacker responds to queries with spoofed IPs; local server accepts and caches fake response

Obstacles

- *Timing* - fake responses must be received by local nameserver before any legitimate responses are received
- *Query ID* - nameserver will ignore fake responses if query ID does not match that in the nameserver's request

Birthday Paradox

- Attacker generates *n* DNS queries
- Local nameserver generates *n* queries to higher-level nameservers, each with a different random query ID (ID_q)
- Attacker generates *n* responses, each with a different random query ID (ID_r)

What is the probability that *at least* one ID_r is equal to one of the server query IDs (ID_q)?

Probabilities

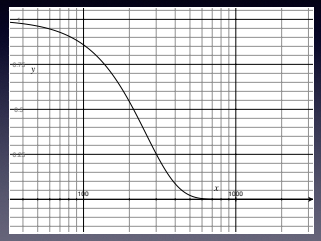
- Probability that a single ID_r does *not* match any of the *n* ID_q values:

$$\left(1-\frac{n}{2^{16}}\right)$$

- Probability that *no* ID_r matches any of the *n* ID_q values:

$$\left(1-\frac{n}{2^{16}}\right)^n$$

Probability Plot



Only need *n* around 200 to have probability 0.5

Let me sum up...

The probabilities show that an attacker only needs to generate a little more than 200 queries and responses to achieve probability 0.5 that the query IDs will match for at least one pair (nameserver query, fake response).

Nonetheless, there are limitations on the attack, TTL in particular - attacker must wait for valid DNS cache entry to expire

Subdomain Poisoning

- Attack ISP's nameserver as above, but query non-existent subdomains
  - `aaaa.example.com`, `aaab.example.com`, etc.
- Authoritative nameserver sees that subdomains do not exist and ignores requests
- Attacker sends fake response with *glue record* containing false IP for `ns.example.com`
- ISP nameserver caches false IP

Glue Records

- Exist to prevent infinite loops in DNS resolution, e.g. `example.com` with DNS server `ns.example.com`
  - Must resolve `example.com` before `ns.example.com`
  - Need to resolve `ns.example.com` to query for `example.com`'s IP
- *Glue record* from high-level server can provide `ns.example.com`'s IP without first resolving `example.com`

Why it works

- Combination of two problems...
  - Nameservers not respondoning to requests for non-existent domains
  - Reliance on 16-bit query ID to authenticate response; Birthday Paradox
- *Source port randomization* adds a bit of defense by making it harder to construct valid responses

Client-side Poisoning

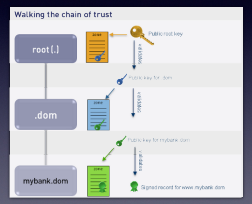
- Web site with lots of image tags - will cause lots of DNS queries to be sent
- Attacker detects navigation to page and sends many DNS responses with random query IDs and poisoned glue records
- If successful, will poison the client's DNS cache

DNSSEC

- Protocol extensions that include digital signatures on DNS response values (RRSIG record)
- Digital signature is hash of response, signed with nameserver's private key
- To verify signature, must obtain appropriate public key (DNSKEY record)
- Validity of nameserver's public key can be established by via "chain of trust" (DS and DNSKEY records)

DNSSEC Keys

- Authoritative servers (DS) can delegate authority to sub-domain servers
- Sub-domain server's key signed by parent
- Ideally, recursive resolver will "chain up" to root DNSKEY



DNSSEC Root

First root zone key deployed June 16, 2010 at a high-security facility in Culpeper, VA.

http://www.icann.org/en/news/announcements/announcement-2-07jun10-en.htm

Exercises are on the website.