The Problem

• The Domain Name Service (DNS) translates human-readable names to IP addresses
  – E.g., thesiger.cs.ucla.edu translates to 131.179.192.144
  – DNS also provides other similar services
• It wasn’t designed with security in mind

DNS Threats

• Threats to name lookup secrecy
  – Definition of DNS system says this data isn’t secret
• Threats to DNS information integrity
  – Very important, since everything trusts that this translation is correct
• Threats to DNS availability
  – Potential to disrupt Internet service

What Could Really Go Wrong?

• DNS lookups could be faked
  – Meaning packets go to the wrong place
• The DNS service could be subject to a DoS attack
  – Or could be used to amplify one
• Attackers could “bug” a DNS server to learn what users are looking up

Where Does the Threat Occur?

• Unlike routing, threat can occur in several places
  – At DNS servers
  – But also at DNS clients
    • Which is almost everyone
• Core problem is that DNS responses aren’t authenticated

The DNS Lookup Process

lookup thesiger.cs.ucla.edu

Should result in a ping packet being sent to 131.179.191.144

ping thesiger.cs.ucla.edu

If the answer is wrong, in standard DNS the client is screwed
How Did the DNS Server Perform the Lookup?

- Leaving aside details, it has a table of translations between names and addresses
- It looked up thesiger.cs.ucla.edu in the table
- And replied with whatever the address was

Where Did That Table Come From?

- Ultimately, the table entries are created by those owning the domains
  - On a good day . . .
- And stored at servers that are authoritative for that domain
- In this case, the UCLA Computer Science Department DNS server ultimately stored it
- Other servers use a hierarchical lookup method to find the translation when needed

Doing Hierarchical Translation

Where Can This Go Wrong?

- Someone can spoof the answer from a DNS server
  - Relatively easy, since UDP is used
- One of the DNS servers can lie
- Someone can corrupt the database of one of the DNS servers

The Spoofing Problem

Unfortunately, most DNS stub resolvers will take the first answer

DNS Servers Lying

That wasn’t very nice of him!
**DNS Database Corruption**

- Output: thesiger.cs.ucla.edu
- Answer: 97.22.101.53

**The DNSSEC Solution**

- Sign the translations
- Who does the signing?
  - The server doing the response?
  - Or the server that “owns” the namespace in question?
- DNSSEC uses the latter solution

**Implications of the DNSSEC Solution**

- DNS databases must store signatures of resource records
- There must be a way of checking the signatures
- The protocol must allow signatures to be returned

**Checking the Signature**

- Basically, use certificates to validate public keys for namespaces
- Who signs the certificates?
  - The entity controlling the higher level namespace
- This implies a hierarchical solution

**An Example**

- Who signs the translation for thesiger.cs.ucla.edu to 131.179.192.144?
- The UCLA CS DNS server
- How does someone know that’s the right server to sign?
- Because the UCLA server says so
  - Securely, with signatures
  - Where do you keep that information?
  - In DNS databases
- Ultimately, hierarchical signatures leading up to ICANN’s attestation of who controls the edu namespace

**Implications for Use**

- To be really secure, you must check signatures yourself
- Next best is to have a really trusted authority check the signatures
  - And to have secure, authenticated communications between trusted authority and you
Some Questions for Discussion

- Partial deployment and interoperability?
- Costs?
- Susceptibility to denial of service?
- Handling negative answers?
- Need also for authenticated communications with server?