Securing Critical Internet Technologies
Computer Security
Peter Reiher
March 9, 2021

Outline

- Routing security
- DNS security

Routing Security

- Routing protocols control how packets flow through the Internet
- If they aren't protected, attackers can alter packet flows at their whim
- Most routing protocols were not built with security in mind

Routing Protocol Security Threats

- Threats to routing data secrecy
  – Usually not critical
- Threats to routing protocol integrity
  – Very important, since tampering with routing integrity can be bad
- Threats to routing protocol availability
  – Potential to disrupt Internet service

What Could Really Go Wrong?

- Packets could be routed through an attacker
- Packets could be dropped
  - Routing loops, blackhole routing, etc.
- Some users' service could be degraded
- The Internet's overall effectiveness could be degraded
  - Slow response to failures
  - Total overload of some links
- Many types of defenses against other attacks presume correct routing

Where Does the Threat Occur?

- At routers, mostly
- Most routers are well-protected
  – But . . .
  – Several vulnerabilities have been found in routers
- Also, should we always trust those running routers?

Different Types of Routing Protocols

- Link state
  – Tell everyone the state of your links
- Distance vector
  – Tell nodes how far away things are
- Path vector
  – Tell nodes the complete path between various points
- On demand protocols
  – Figure out routing once you know you two nodes need to communicate

Popular Routing Protocols

- BGP
  - Path vector protocol used in core Internet routing
  - Arguably most important protocol to secure
- RIP
  - Distance vector protocol for small networks
- OSPF
- ISIS
- Ad hoc routing protocols

Fundamental Operations To Be Protected

- One router tells another router something about routing
  - A path, a distance, contents of local routing table, etc.
- A router updates its routing information
- A router gathers information to decide on routing

Protecting BGP

- BGP is probably the most important protocol to protect
- Handles basic Internet routing
- Works at autonomous system (AS) level
  - Rather than router level

BGP Issues

- BGP is spoken (mostly) between routers in autonomous systems
- On direct network links to their partner
- Over TCP sessions that are established with known partners
  - Easily encrypted, if desired
- Isn't that enough to give reasonable security?

A Counterexample

- Pakistan became upset with YouTube over posting of “blasphemous” video (2008)
- Responded by injecting a BGP update that sent all traffic to YouTube to a site in Pakistan
  - Which probably dropped it all
- Rendered YouTube unavailable worldwide (well, 2/3s of world)
  - Probably due to error, not malice

How Did This Happen?

- Pakistan injected a BGP update advertising a path to YouTube
  – Which they had no right to do
- It got automatically propagated by BGP
- Everyone knows YouTube isn’t in Pakistan
- But the routing protocol didn’t
- Security required to prevent other future incidents

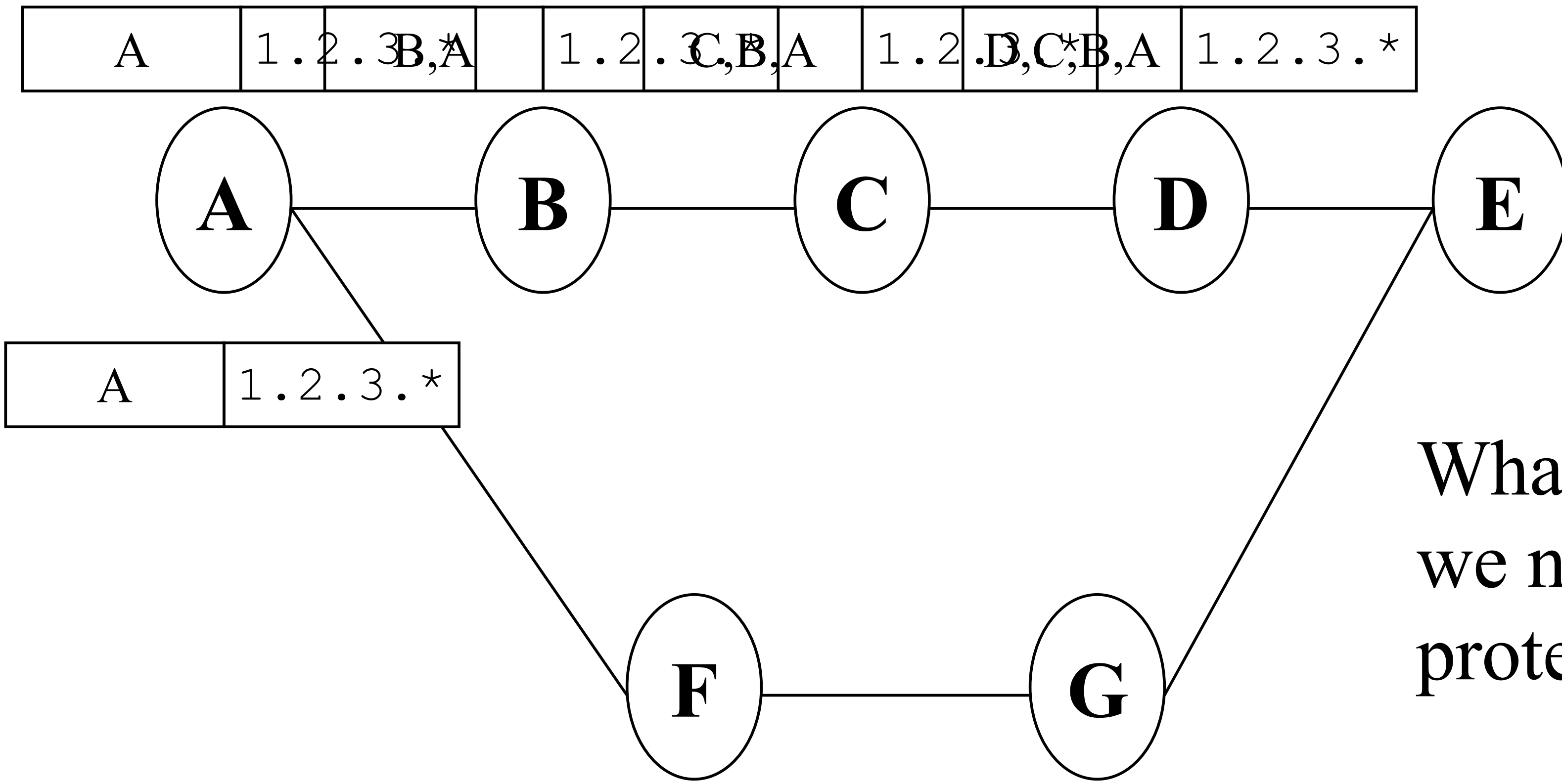
Another Example

- In 2010, China rerouted a lot of US traffic through its servers
  - Traffic purely internal to the US
  - Lots of military, government, commercial traffic
- Based on bogus BGP route advertisements
- Possibly errors, not attacks, but . . .

A Side Issues on This Story

- Much Internet design assumes major parties play by the rules
- Pakistan didn't
- Not desirable to base Internet's security on this assumption
- Though sometimes not many other choices

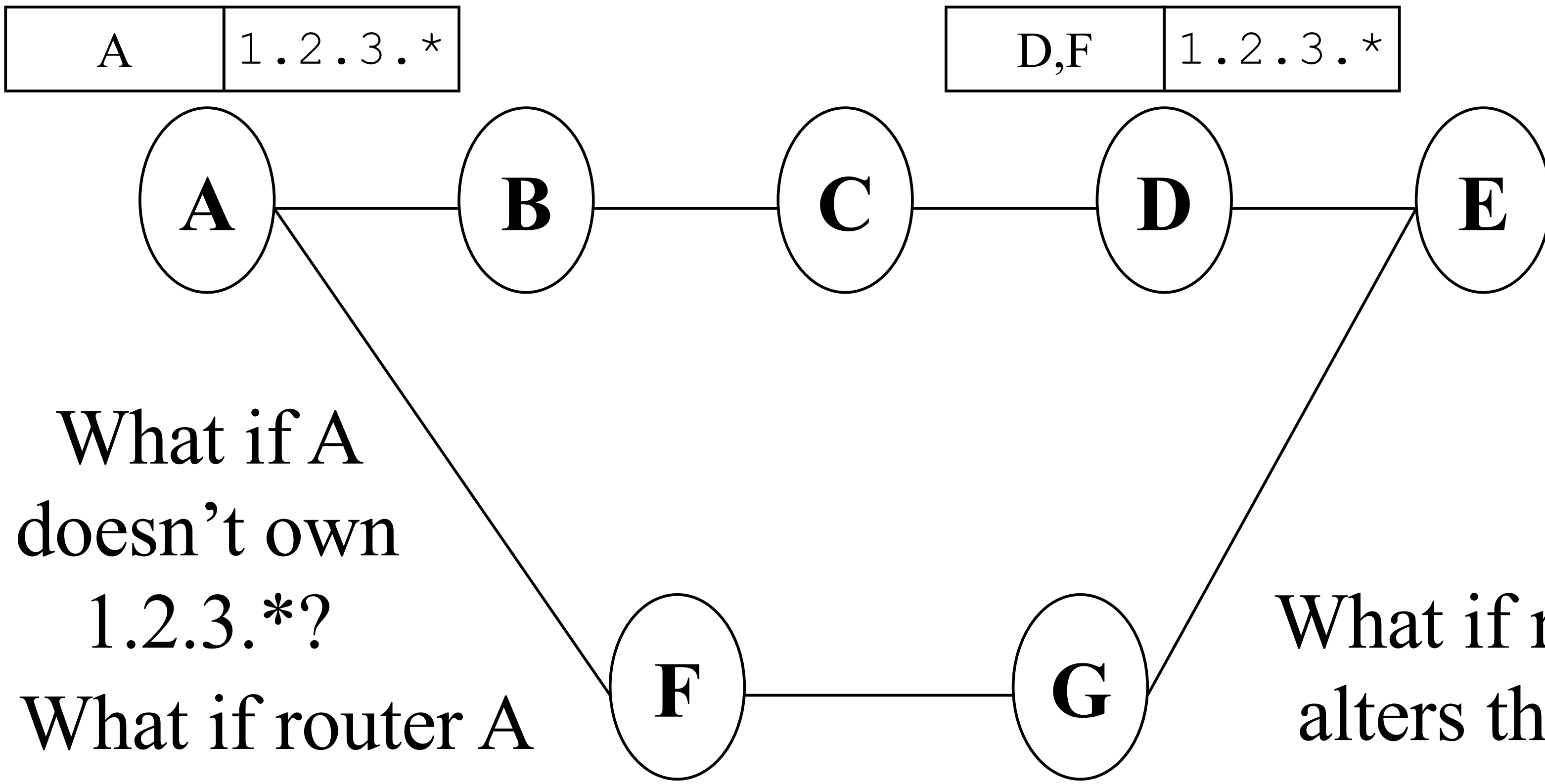
Basic BGP Security Issue



What do we need to protect?

A wants to tell everyone how to get to 1.2.3.*

Well, What Could Go Wrong?



What if A doesn't own 1.2.3.*?

What if router A isn't authorized to advertise 1.2.3.*?

What if router D alters the path?

Two Sub-Problems

- Security of Origin (SOA)
  - Who is allowed to advertise a path to an IP prefix?
- Path Validation (PV)
  - Is the path someone gives to me indeed a correct path?

How Do We Solve These Problems?

- SOA - Advertising routers must prove prefix ownership
  - And right to advertise paths to that prefix
- PV - Paths must be signed by routers on them
  - Must avoid cut-and-paste and replay attacks

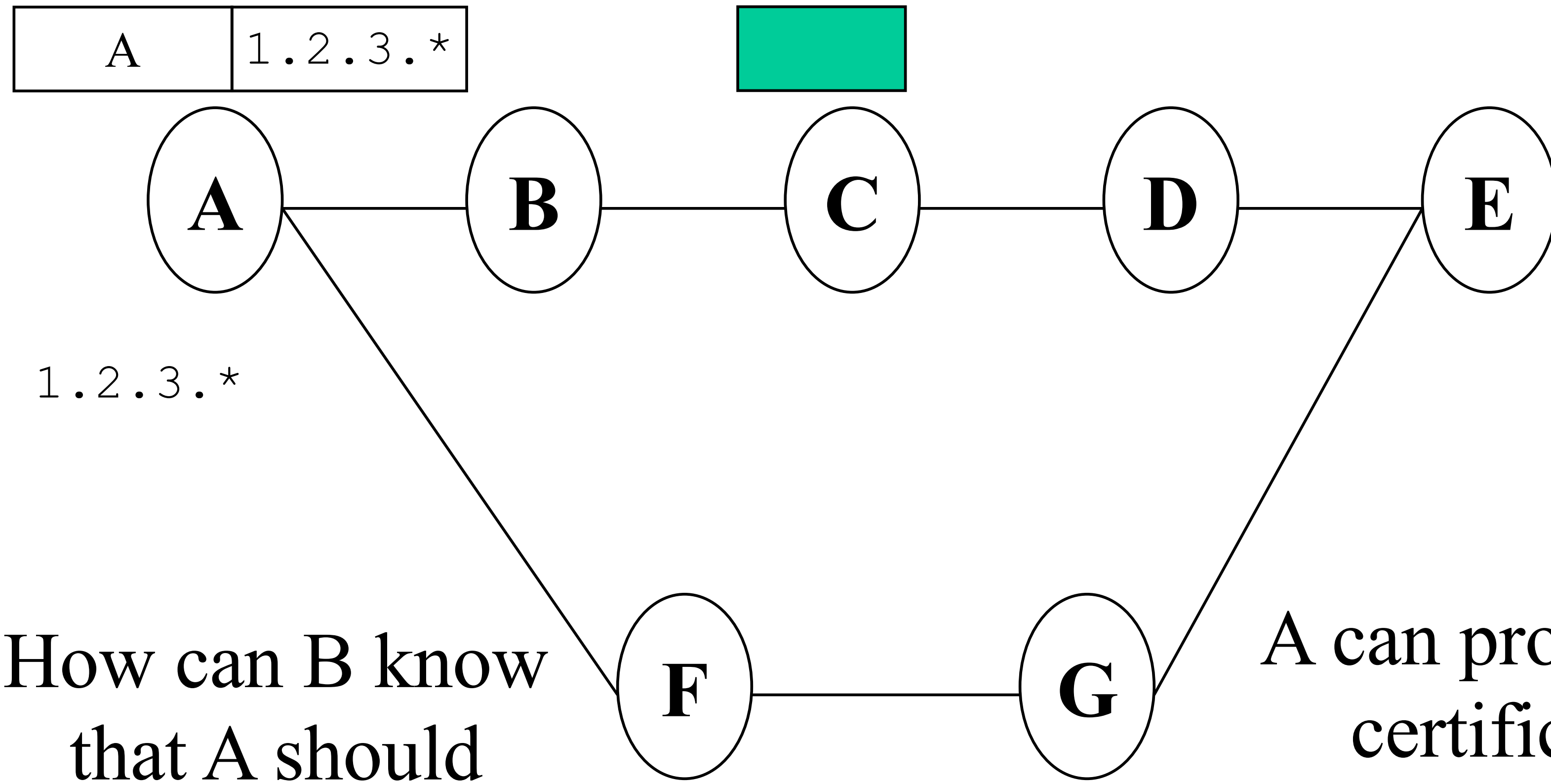
S-BGP

- One example solution
- A protocol designed to solve most of the routing security issues for BGP
- Intended to be workable with existing BGP protocol
- Key idea is to tie updates to those who are allowed to make them
  – And to those who build them

Some S-BGP Constraints

- Can't change BGP protocol
  - Or packet format
- Can't have messages larger than max BGP size
- Must be deployable in reasonable way

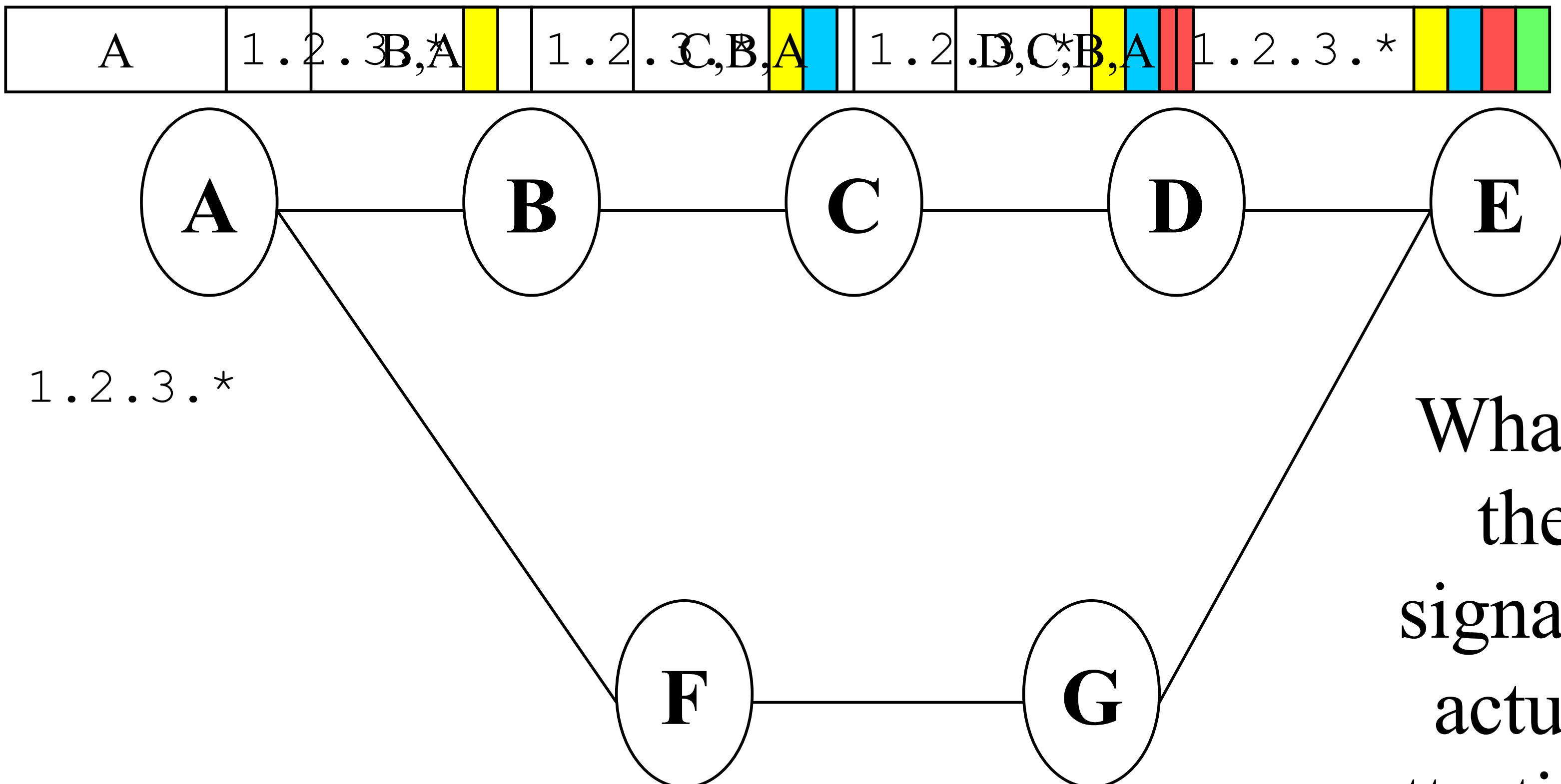
An S-BGP Example



How can B know that A should advertise 1.2.3.*?

A can provide a certificate proving ownership

Securing BGP Updates



What are these signatures actually attesting to?

A wants to tell everyone how to get to 1.2.3.*

Who Needs To Prove What?

- A needs to prove (to B-E) that he owns the prefix
- B needs to prove (to C-E) that A wants the prefix path to go through B
- C needs to prove (to D-E) the same
- D needs to prove (to E) the same

So What Does A Sign?

- A clearly must provide proof he owns the prefix
- He also must prove he originated the update
- And only A can prove that he intended the path to go through B
- So he has to sign for all of that

Address Attestations in S-BGP

- These are used to prove ownership of IP prefix spaces
- IP prefix owner provides attestation that a particular AS can originate its BGP updates
- That AS includes attestation in updates

Route Attestations

- To prove that path for a prefix should go through an AS
- The previous AS on the path makes this attestation
  - E.g., B attests that C is the next AS hop

How Are These Signatures Done?

- Via public key cryptography
- Certificates issued by proper authorities
  - ICANN at the top
  - Hierarchical below ICANN
- Certificates not carried with updates
  - Otherwise, messages would be too big
  - Off-line delivery method proposed

S-BGP and IPSec

- S-BGP generates the attestations itself
- But it uses IPSec to deliver the BGP messages
- Doing so prevents injections of replayed messages
- Also helps with some TCP-based attacks
  - E.g., SYN floods

S-BGP Status

- Not getting traction in networking community
- Probably not going to be the ultimate solution
- IETF working group is looking at various protocols with similar approaches
  - BGPsec, for example

Other BGP Security Approaches

- Filter BGP updates from your neighbors
  – Don’t accept advertisements for prefixes they don’t own
  – Requires authoritative knowledge of who owns prefixes
- Use Resource PKI to distribute certificates on who owns what prefixes
- Sanity check routes
- Continuous monitoring of routing system

DNS Security

- 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?

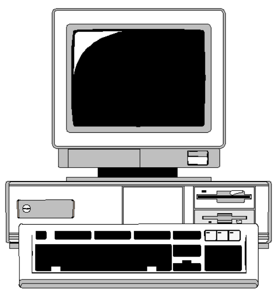
- DNS lookups could be faked
  - Meaning packets go to the wrong place
- The DNS service could be subject to a DoS attack
  - Or could be used to amplify one
- Attackers could "bug" a DNS server to learn what users are looking up

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

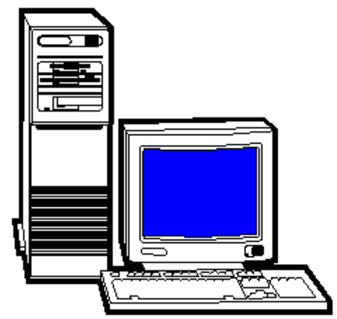
`lookup thesiger.cs.ucla.edu`



`ping thesiger.cs.ucla.edu`

Should result in a ping packet being sent to 131.179.191.144

`answer  131.179.191.144`



If the answer is wrong, in standard DNS the client is screwed

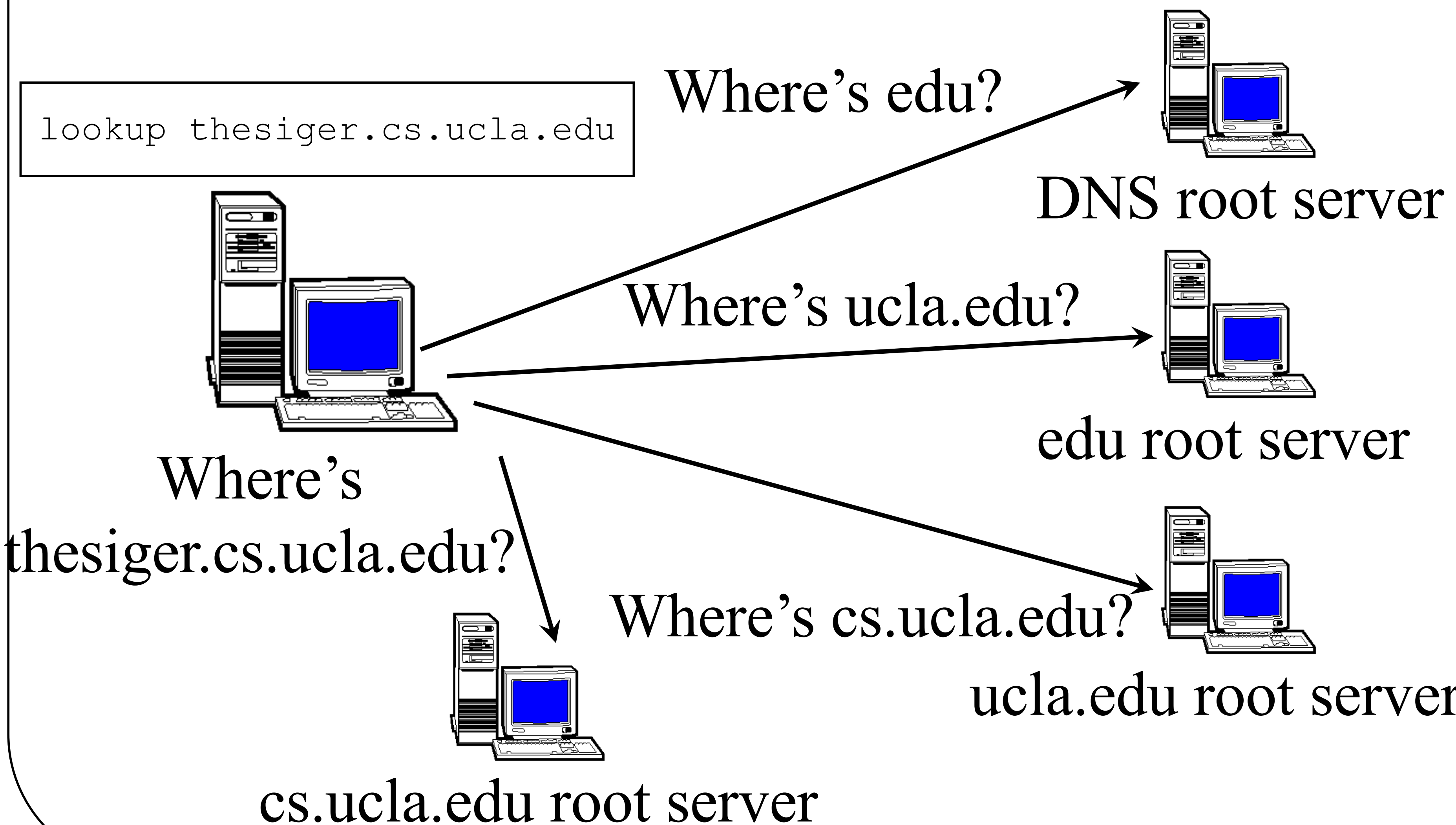
How Did the DNS Server Perform the Lookup?

- 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



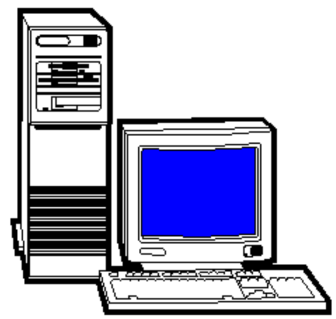
Where Can This Go Wrong?

- 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

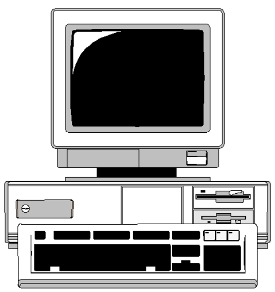
The Spoofing Problem

`lookup thesiger.cs.ucla.edu`

`answer  131.179.191.144`



Unfortunately, most DNS stub resolvers will take the first answer

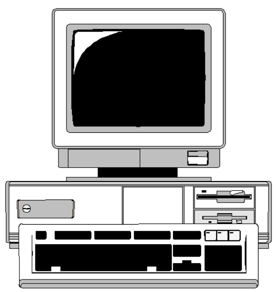


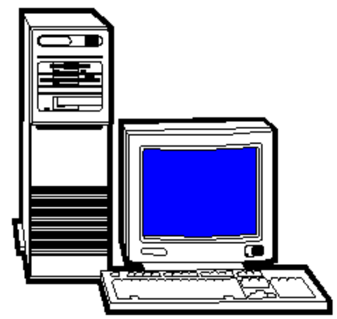
`answer  97.22.101.53`

DNS Servers Lying

lookup thesiger.cs.ucla.edu

answer 97.22.101.53



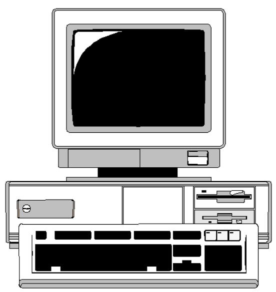


. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
thesiger.cs.ucla.edu	131.178.192.144
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

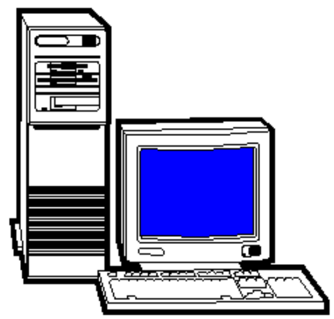
That wasn't very nice of him!

DNS Cache Poisoning

```
lookup thesiger.cs.ucla.edu
```



```
answer  97.22.101.53
```



. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
thesiger.cs.ucla.edu	97.22.101.53
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .
. . .	. . .

Unless the server is authoritative for the name, the lookup is in a server cache

The attacker "poisoned" the DNS

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

The DNSSEC Signing Hierarchy

- In principle, ICANN signs for itself and for top level domains (TLDs)
  - Like .com, .edu, country codes, etc.
- Each TLD signs for domains under it
- Those domains sign for domains below them
- And so on down

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
- The edu server verifies the UCLA server's signature
- Ultimately, hierarchical signatures leading up to ICANN's attestation of who controls the edu namespace
- Where do you keep that information?
  - In DNS databases

Using DNSSEC

- 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

A Major Issue

- When you look up something like cs.ucla.edu, you get back a signed record
- What if you look up a name that doesn't exist?
- How can you get a signed record for every possible non-existent name?

The DNSSEC Solution

- Names are alphabetically orderable
- Between any two names that exist, there are a bunch of names that don't
- Sign the whole range of non-existent names
- If someone looks one up, give them the range signature

For Example,

lasr.cs.ucla.edu	131.179.192.136	
last.cs.ucla.edu / pelican.cs.ucla.edu	NOT ASSIGNED / 131.179.128.17	
pd.cs.ucla.edu / toucan.cs.ucla.edu	131.179.128.16	
pelican.cs.ucla.edu	131.179.128.17	
toucan.cs.ucla.edu	131.179.128.16	

You get authoritative information that the name isn't assigned

Foils spoofing attacks

```
> host last.cs.ucla.edu
```

Status of DNSSEC

- Working implementations available
- In use in some places
- Heavily promoted
  - First by DARPA
  - Now by DHS
- Beginning to get out there

Status of DNSSEC Deployment

- ICANN has signed the root
  - Over 1300 TLDs have signed
  - Including .com, .gov, .edu, .org, .net
  - Not everyone below has signed, though
- Many “islands” of DNSSEC signatures
  - Signing for themselves and those below them
  - In most cases, just for themselves
- Utility depends on end machines checking signatures

Using DNSSEC

- Actually installing and using DNSSEC not quite as easy as it sounds
- Lots of complexities down in the weeds
- Particularly hard for domains with lots of churn in their namespace
  - Every new name requires big changes to what gets signed

Other DNS Security Solutions

- Encrypt communications with DNS servers
  - Prevents DNS cache poisoning
  - But assumes that DNS server already has right record
- Ask multiple servers
  - Majority rules or require consensus
- Use packet sequence number randomization to make it hard to poison a cache

Conclusion

- Correct Internet behavior depends on a few key technologies
  - Especially routing and DNS
- Initial (still popular) implementations of those technologies are not secure
- Work is ongoing on improving their security