Domain Name System (DNS)

Bhaskaran Raman

Department of CSE, IIT Kanpur

Domain Name System (DNS)

- Require a name to IP mapping
- History: HOSTS.TXT file used around 1982
- Problems:
  - File became large
  - Higher rate of change
  - Centralized control

DNS Design Goals

- Distributed ownership
- No obvious size limits
- Independent of network topology, capable of other name spaces as well
- OS/Architecture independent

Design Principles

- Hierarchy
  - In name space
  - In management
- Caching
  - Lean service instead of a generic distributed database
  - Digression: **C**onsistency, **A**vailability, Resilience to **P**artitions – chose any two (CAP principle)

DNS Architecture

- Name servers and resolvers
- Variable depth name space: case insensitive
- Decouple tree structure from any semantics
  - For example, in-addr.arpa for reverse lookup
  - .com, .edu, .in semantics not known to DNS
- Data attached to names
  - Resource Records (RR)
  - Data type for each record (A record, MX record)

Hierarchy and Caching

- DNS zones:
  - Contiguous regions in name space
  - A zone is controlled by an organization
  - Can fork off a portion of a zone to a sub-zone
- A name server can support multiple zones
- Caching:
  - Implemented using Time-To-Live (TTL)
  - Trade-off between consistency and update overhead

Some Remarks

- Root server is replicated for availability
- Datagram-based access
- Additional section processing: add unsolicited information to queries
- Beware of misconfigured entries!

Using DNS for System Break-in

- IP source spoofing attacks possible
- But, authentication is based on host-name, not IP-address
  - E.g., **/etc/hosts.equiv** or **~/.rhosts** used by rsh, rlogin
  - Easy break-in, no need for any IP-address spoofing

Attacking rlogin/rsh

- How do rlogin/rsh work?
  - Incoming connection from **1.2.3.4**
  - DNS (reverse) lookup **1.2.3.4**
  - Check against **~/.rhosts** or **/etc/hosts.equiv**
- Attack!
  - Return trusted host's name from hacker's domain!
  - Need to know target host, trusted host, user name
  - Easy: use finger, SNMP, email
  - Or, use DNS itself!

How to Fix?

- Have rsh/rlogin do forward lookup as well
- Still easy to break: can **poison** DNS cache!
  - Attacker sends unsolicited **A** record along with **PTR** record
- Fix by rejecting **A** record which arrives along with **PTR** record?
  - Still, can force a DNS query to hacker's domain

Concluding Remarks

- DNS is the backbone of Internet applications: WWW, Email
-
- DNS authentication required
- But, DNS not necessarily the issue!
- Cryptographic authentication required in rlogin/rsh or similar applications
- Logs would be useful