Network Protocols and Vulnerabilities

Dan Boneh
Outline

- Basic Networking:
  - How things work now plus some problems

- Some network attacks
  - Attacking host-to-host datagram protocols
    - TCP Spoofing, ...
  - Attacking network infrastructure
    - Routing
    - Domain Name System
Internet Infrastructure

- Local and interdomain routing
  - TCP/IP for routing, connections
  - BGP for routing announcements
- Domain Name System
  - Find IP address from symbolic name (www.cs.stanford.edu)
TCP Protocol Stack

- Application
- Transport
- Network
- Link

Application protocol

TCP protocol

IP protocol

Data
Link

Network Access

Application
Transport
Network
Link
Data Formats

- Application
- Transport (TCP, UDP)
- Network (IP)
- Link Layer

TCP Header

Application message - data

TCP data

IP data

ETH ETF

IP Header

Link (Ethernet) Header

Link (Ethernet) Trailer
Internet Protocol

- **Connectionless**
  - Unreliable
  - Best effort

- **Notes:**
  - src and dest **ports** not parts of IP hdr

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td></td>
</tr>
<tr>
<td>Header Length</td>
<td></td>
</tr>
<tr>
<td>Type of Service</td>
<td></td>
</tr>
<tr>
<td>Total Length</td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>Time to Live</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>Source Address of Originating Host</td>
<td></td>
</tr>
<tr>
<td>Destination Address of Target Host</td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>Padding</td>
<td></td>
</tr>
<tr>
<td>IP Data</td>
<td></td>
</tr>
</tbody>
</table>
Internet routing uses numeric IP address
Typical route uses several hops
IP Protocol Functions (Summary)

- **Routing**
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks

- **Fragmentation and reassembly**
  - If max-packet-size less than the user-data-size

- **Error reporting**
  - ICMP packet to source if packet is dropped

- **TTL field**
  - Decremented after every hop
  - Packet dropped if TTL=0. Prevents infinite loops.
Problem: no src IP authentication

- Client is trusted to embed correct source IP
  - Easy to override using raw sockets
  - **Libnet**: a library for formatting raw packets with arbitrary IP headers

- Anyone who owns their machine can send packets with arbitrary source IP
  - ... response will be sent back to forged source IP

- **Implications**: (solutions in DDoS lecture)
  - Anonymous DoS attacks;
  - Anonymous infection attacks (e.g. slammer worm)
User Datagram Protocol

Unreliable transport on top of IP:
- No acknowledgment
- No congestion control
- No message continuation

IP Header

UDP Header
Transmission Control Protocol

- Connection-oriented, preserves order
  - **Sender**
    - Break data into packets
    - Attach packet numbers
  - **Receiver**
    - Acknowledge receipt; lost packets are resent
    - Reassemble packets in correct order

Diagram:

- Book
  - Mail each page
  - Reassemble book
TCP Header

TCP Header:
- Source Port
- Dest port
- SEQ Number
- ACK Number
- URG
- ACK
- PSH
- SYN
- FIN
- Other stuff

IP Header:
- Source IP address
- Destination IP address

32 bits
Review: TCP Handshake

Received packets with SN too far out of window are dropped.
Basic Security Problems

1. Network packets pass by untrusted hosts
   - Eavesdropping, packet snifffing
   - Especially easy when attacker controls a machine close to victim

2. TCP state can be easy to guess
   - Enables spoofing and session hijacking

3. Denial of Service (DoS) vulnerabilities
   - DDoS lecture
1. Packet Sniffing

- Promiscuous NIC reads all packets
  - Read all unencrypted data (e.g., "wireshark")
  - ftp, telnet (and POP, IMAP) send passwords in clear!

**Sweet Hall attack** installed sniffer on local machine

Prevention: Encryption  (next lecture: IPSec)
2. TCP Connection Spoofing

Why random initial sequence numbers? \((S_{NC}, S_{NS})\)

Suppose init. sequence numbers are predictable
- Attacker can create TCP session on behalf of forged source IP
  - Breaks IP-based authentication (e.g. SPF, /etc/hosts)

```
TCP SYN
srcIP=victim

ACK
srcIP=victim
AN=predicted \(S_{NS}\)

command

SYN/ACK
dstIP=victim
SN=server \(S_{NS}\)
```

Server thinks command is from victim IP addr
Example DoS vulnerability [Watson’04]

- Suppose attacker can guess seq. number for an existing connection:
  - Attacker can send Reset packet to close connection. Results in DoS.
  - Naively, success prob. is \(1/2^{32}\) (32-bit seq. #’s).
  - Most systems allow for a large window of acceptable seq. #’s
    - Much higher success probability.

- Attack is most effective against long lived connections, e.g. BGP
Random initial TCP SNs

- Unpredictable SNs prevent basic packet injection
  - ... but attacker can inject packets after eavesdropping to obtain current SN

- Most TCP stacks now generate random SNs
  - Random generator should be unpredictable
  - GPR’06: Linux RNG for generating SNs is predictable
    - Attacker repeatedly connects to server
    - Obtains sequence of SNs
    - Can predict next SN
    - Attacker can now do TCP spoofing (create TCP session with forged source IP)
Routing Vulnerabilities
Routing Vulnerabilities

- **Common attack:** advertise false routes
  - Causes traffic to go through compromised hosts

- **ARP (addr resolution protocol):** IP addr -> eth addr
  - Node A can confuse gateway into sending traffic for B
  - By proxying traffic, attacker A can easily inject packets into B’s session (e.g. WiFi networks)

- **OSPF:** used for routing within an AS

- **BGP:** routing between ASs
  - Attacker can cause entire Internet to send traffic for a victim IP to attacker’s address.
  - Example: Youtube mishap (see DDoS lecture)
Interdomain Routing

connected group of one or more Internet Protocol prefixes under a single routing policy (aka domain)

earthlink.net

Stanford.edu

BGP

Autonomous System

OSPF
BGP overview

Iterative path announcement
- Path announcements grow from destination to source
- Packets flow in reverse direction

Protocol specification
- Announcements can be shortest path
- Not obligated to use announced path
BGP example

Transit: 2 provides transit for 7
Algorithm seems to work OK in practice
- BGP is does not respond well to frequent node outages
Issues

Security problems
- Potential for disruptive attacks
- BGP packets are un-authenticated
  - Attacker can advertise arbitrary routes
  - Advertisement will propagate everywhere
  - Used for DoS and spam (detailed example in DDoS lecture)

Incentive for dishonesty
- ISP pays for some routes, others free
Domain Name System
Domain Name System

Hierarchical Name Space
DNS Root Name Servers

**Hierarchical service**
- Root name servers for top-level domains
- Authoritative name servers for subdomains
- Local name resolvers contact authoritative servers when they do not know a name
DNS Lookup Example

DNS record types (partial list):
- NS: name server (points to other server)
- A: address record (contains IP address)
- MX: address in charge of handling email
- TXT: generic text (e.g. used to distribute site public keys (DKIM))
Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains

- DNS negative queries are cached
  - Save time for nonexistent sites, e.g. misspelling

- Cached data periodically times out
  - Lifetime (TTL) of data controlled by owner of data
  - TTL passed with every record
DNS Packet

Query ID:
- 16 bit random value
- Links response to query

(from Steve Friedl)
Resolver to NS request

IP

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>identification</th>
<th>flg</th>
<th>fragment offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TTL</th>
<th>protocol</th>
<th>header cksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

src IP = 68.94.156.1
dst IP = 192.26.92.30
dnsr1.sbcglobal.net
c.gtld-servers.net

UDP

<table>
<thead>
<tr>
<th>src port</th>
<th>dst port</th>
</tr>
</thead>
<tbody>
<tr>
<td>5798</td>
<td>53</td>
</tr>
</tbody>
</table>

UDP length

<table>
<thead>
<tr>
<th>QID</th>
<th>0</th>
<th>Op=0</th>
<th>1</th>
<th>Z</th>
<th>rcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>43561</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question count = 1
Answer count = 0
Authority count = 0
Additional Record count = 0

Qu: What is A record for www.unixwiz.net?
Response to resolver

Response contains IP addr of next NS server (called “glue”)

Response ignored if unrecognized QueryID
**Authoritative response to resolver**

**bailiwick checking:**
response is cached if it is within the same domain of query (i.e. **a.com** cannot set NS for **b.com**)

---

<table>
<thead>
<tr>
<th>IP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>src IP = 64.170.162.98</td>
<td>dst IP = 68.94.156.1</td>
</tr>
<tr>
<td>src port = 53</td>
<td>dst port = 5798</td>
</tr>
</tbody>
</table>

**QID = 43562**
- Question count = 1
- Authority count = 2
- Addl. Record count = 2

**Qu:** What is A record for www.unixwiz.net?

**An:**
- www.unixwiz.net A = 8.7.25.94 1 hr
- unixwiz.net NS = linux.unixwiz.net 2 dy
- unixwiz.net NS = cs.unixwiz.net 2 dy
- linux.unixwiz.net A = 64.170.162.98 1 hr
- cs.unixwiz.net A = 8.7.25.94 1 hr

**QR=1 - this is a response**
**AA=1 - Authoritative!**
**RA=0 - recursion unavailable**
Basic DNS Vulnerabilities

- Users/hosts trust the host-address mapping provided by DNS:
  - Used as basis for many security policies:
    - Browser same origin policy, URL address bar

- Obvious problems
  - Interception of requests or compromise of DNS servers can result in incorrect or malicious responses
    - e.g.: hijack BGP route to spoof DNS
  - Solution – authenticated requests/responses
    - Provided by DNSsec but no one uses DNSsec
DNS cache poisoning (a la Kaminsky’08)

- Victim machine visits attacker’s web site, downloads Javascript

```
user
browser
```

```
Query:
a.bank.com
```

```
local
DNS
resolver
```

```
a.bank.com
QID=x_1
```

```
ns.bank.com
```

```
attacker
```

```
256 responses:
Random QID y_1, y_2, ...
NS bank.com=ns.bank.com
A ns.bank.com=attackerIP
```

attacker wins if \( \exists j: x_1 = y_j \) if response is cached and attacker owns bank.com
If at first you don’t succeed …

- Victim machine visits attacker’s web site, downloads Javascript

**Diagram**

- User browser queries `b.bank.com`
- Local DNS resolver with QID=`x_2`
- `b.bank.com` resolves to `attackerIP`
- `ns.bank.com` with IP address

**Attacker wins if**

\[ \exists j : x_2 = y_j \]

response is cached and attacker owns `bank.com`

**Success after** \( \approx 256 \) tries (few minutes)
Defenses

- Increase Query ID size. How?
  
a. Randomize src port, additional 11 bits
    Now attack takes several hours
  
b. Ask every DNS query twice:
    - Attacker has to guess QueryID correctly twice (32 bits)
    - Apparently DNS system cannot handle the load
Pharming

- DNS poisoning attack (less common than phishing)
  - Change IP addresses to redirect URLs to fraudulent sites
  - Potentially more dangerous than phishing attacks
  - No email solicitation is required

- DNS poisoning attacks have occurred:
  - January 2005, the domain name for a large New York ISP, Panix, was hijacked to a site in Australia.
  - In November 2004, Google and Amazon users were sent to Med Network Inc., an online pharmacy.
  - In March 2003, a group dubbed the "Freedom Cyber Force Militia" hijacked visitors to the Al-Jazeera Web site and presented them with the message "God Bless Our Troops"
DNS Rebinding Attack

Read permitted: it’s the “same origin”
DNS Rebinding Defenses

**Browser mitigation: DNS Pinning**
- Refuse to switch to a new IP
- Interacts poorly with proxies, VPN, dynamic DNS, ...
- Not consistently implemented in any browser

**Server-side defenses**
- Check Host header for unrecognized domains
- Authenticate users with something other than IP

**Firewall defenses**
- External names can’t resolve to internal addresses
- Protects browsers inside the organization
Summary

Core protocols not designed for security
- Eavesdropping, Packet injection, Route stealing, DNS poisoning
- Patched over time to prevent basic attacks
  (e.g. random TCP SN)

More secure variants exist (next lecture):
- IP -> IPsec
- DNS -> DNSsec
- BGP -> SBGP