Security and Networks

Daniel Halperin
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Class updates

- (Short) Homework 3
- Due Wednesday
- Individual assignment

- My office hours this week:
  - **CSE 210**: M, W, F after class. T, Th afternoons
  - others by appointment

- come pick up graded Homework #2
Lab 3

• **Posted on website and on Catalyst.**
  
  • https://catalyst.uw.edu/collectit/assignment/dhalperi/17513/72548

• Hack my privacy!
Lab 3

- Posted on website and on Catalyst.
  - https://catalyst.uw.edu/collectit/assignment/dhalperi/17513/72548
- Hack my privacy!

- This lab is optional
  - Can only help your grade.
  - Lots of opportunity for extra credit.
  - I really think this lab is fun, and encourage you to do it, but we’re not going to require it.
This week

• **Today:** Network security

• **Wednesday:** Potpourri

• **Friday:** Any questions you have
  • Submit to my email, cse484-tas
  • Submit anonymously via the feedback form on the website
Internet Infrastructure

- TCP/IP for packet routing and connections
- Border Gateway Protocol (BGP) for route discovery
- Domain Name System (DNS) for IP address discovery
(Some) Entities

User → Network Admin → Intermediate ISPs → Server
(Some) Goals

User
(Some) Goals

- Service (can get to Internet)
- Privacy (middle-entities shouldn’t know what communicating or with whom)
- Fairness (e.g., get service I paid for)
- Integrity (can’t impersonate me)
- Safety (network shouldn’t attack me)
(Some) Goals

Network Admin
(Some) Goals

- Service (clients can get to Internet)
- Performance (network works well)
- Identity (know what’s on network)
- Safety (no one launching attacks)
- Accountability (can find bad users)
(Some) Goals

Intermediate ISPs
(Some) Goals

- Service (deliver traffic -> earn $$)
- Reliability & Performance (network works well)
- Integrity of delivered traffic (can bill customers properly, you’re not overcharged by providers)
(Some) Goals
(Some) Goals

- Service (deliver traffic -> earn $$)
- Reliability & Performance (network works well)
- Analytics (better delivery)
- Accounting (can bill customers properly)
- Safety (not being attacked)
(Some) Malicious Goals

User → Network Admin → Intermediate ISPs → Server
(Some) Malicious Goals

User
Launch undetectable attacks
Probe for vulnerabilities

Network Admin

Intermediate ISPs

Server
(Some) Malicious Goals

- User
  - Launch undetectable attacks
  - Probe for vulnerabilities

- Network Admin

- Intermediate ISPs
  - Spy on/tamper with traffic
  - Impersonate servers

- Server
(Some) Malicious Goals

User
- Launch undetectable attacks
- Probe for vulnerabilities

Network Admin
- Spy on/tamper with traffic

Intermediate ISPs
- Impersonate servers

Server
- Spy on users
- Identify anonymous users
OSI Protocol Stack

- application
- presentation
- session
- transport
- network
- data link
- physical

- email, Web, NFS
- RPC
- TCP, UDP, ICMP
- IP
- Ethernet
IP (Internet Protocol)

- Connectionless
  - Unreliable, “best-effort” protocol
- Uses numeric addresses for routing
  - Typically several hops in the route

Alice’s computer

Alice’s ISP

128.83.130.239

Packet

<table>
<thead>
<tr>
<th>Source</th>
<th>128.83.130.239</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest</td>
<td>171.64.66.201</td>
</tr>
</tbody>
</table>

Bob’s ISP

Bob’s computer

171.64.66.201
TCP (Transmission Control Protocol)

- **Sender**: break data into packets
  - Sequence number is attached to every packet

- **Receiver**: reassemble packets in correct order
  - Acknowledge receipt; lost packets are re-sent

- **Connection** state maintained on both sides
UDP (User Datagram Protocol)

- **Sender:** break data into packets
  - *Sequence number* - maybe? If Application wants them

- **Receiver:** receive packets
  - No acknowledgement
  - Dropped packets are skipped - no retransmission

Diagram:
- Video frames are sent as packets.
- Each packet is acknowledged by the receiver.
- Stream frames are sent to the application.
ICMP (Control Message Protocol)

- Provides feedback about network operation
  - “Out-of-band” messages carried in IP packets
  - Error reporting, congestion control, reachability, etc.

- Example messages:
  - Destination unreachable
  - Time exceeded
  - Parameter problem
  - Redirect to better gateway
  - Reachability test (echo / echo reply)
  - Message transit delay (timestamp request / reply)
(Some) Malicious Goals

User → Network Admin → Intermediate ISPs → Server
(Some) Malicious Goals

User
Launch undetectable attacks

Network Admin

Intermediate ISPs

Server
Probe for vulnerabilities
(Some) Malicious Goals

User
- Launch undetectable attacks
- Probe for vulnerabilities

Network Admin

Intermediate ISPs
- Spy on/tamper with traffic
- Impersonate servers

Server
(Some) Malicious Goals

Network Admin

Spy on/tamper with traffic

Impersonate servers

ISP

Launch undetectable attacks

Probe for vulnerabilities

User

Intermediate ISPs

Identify anonymous users

Server
Detecting attacks

User

Launch undetectable attacks
Detecting attacks

- **Problem:** IP packets contain source IP address
Detecting attacks

• **Problem:** IP packets contain source IP address

User

Launch undetectable attacks
Detecting attacks

User

Launch undetectable attacks

• **Problem:** IP packets contain source IP address

• **Solution:** Spoof IP address
Inferring DDOS (Moore, Voelker, Savage '01)

SYN packets

Attacker

Victim

Attack

Backscatter
Finding vulnerabilities

User

Probe for vulnerabilities
Finding vulnerabilities

• Many, many tools

User

Probe for vulnerabilities
Finding vulnerabilities

- Many, many tools
- One example: **Nmap**
  - Many services have known TCP/UDP ports
  - These give away what services you’re running
Nmap example (me)

dhalperi@dhp cse484 % nmap dsp.cs.washington.edu

Starting Nmap 5.51 (http://nmap.org) at 2011-12-05 14:05 PST
Nmap scan report for dsp.cs.washington.edu (128.208.4.246)
Host is up (0.0062s latency).
Not shown: 996 closed ports

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>22/tcp</td>
<td>open</td>
<td>ssh</td>
</tr>
<tr>
<td>139/tcp</td>
<td>open</td>
<td>netbios-ssn</td>
</tr>
<tr>
<td>443/tcp</td>
<td>open</td>
<td>https</td>
</tr>
<tr>
<td>445/tcp</td>
<td>open</td>
<td>microsoft-ds</td>
</tr>
</tbody>
</table>

Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
Nmap example (aqua)

dhalperi@dhm cse484 % nmap  aqua.cs.washington.edu

Starting Nmap 5.51 (http://nmap.org) at 2011-12-05 14:06 PST
Nmap scan report for aqua.cs.washington.edu (128.208.4.187)
Host is up (0.0022s latency).
Not shown: 990 filtered ports

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>80/tcp</td>
<td>open</td>
<td>http</td>
</tr>
<tr>
<td>135/tcp</td>
<td>open</td>
<td>msrpc</td>
</tr>
<tr>
<td>139/tcp</td>
<td>open</td>
<td>netbios-ssn</td>
</tr>
<tr>
<td>445/tcp</td>
<td>open</td>
<td>microsoft-ds</td>
</tr>
<tr>
<td>1025/tcp</td>
<td>open</td>
<td>NFS-or-IIS</td>
</tr>
<tr>
<td>1026/tcp</td>
<td>open</td>
<td>LSA-or-nterm</td>
</tr>
<tr>
<td>1027/tcp</td>
<td>open</td>
<td>IIS</td>
</tr>
<tr>
<td>1028/tcp</td>
<td>open</td>
<td>unknown</td>
</tr>
<tr>
<td>1048/tcp</td>
<td>open</td>
<td>neod2</td>
</tr>
<tr>
<td>3389/tcp</td>
<td>open</td>
<td>ms-term-serv</td>
</tr>
</tbody>
</table>

Nmap done: 1 IP address (1 host up) scanned in 5.29 seconds
telnet example
Fingerprinting users

Server

Identify anonymous users
Fingerprinting users

- Browser

Identify anonymous users
Fingerprinting users

• Browser
• Clocks

Identify anonymous users

Server
Fingerprinting users

- Browser
- Clocks
- More

Server

Identify anonymous users
Browser example
http://panopticlick.eff.org/
Clocks
The slope of the points in the trace is short, we cannot always approximate the slope of particular the wide band, means that if the duration of our resolution (that the narrow band corresponds to a device with a higher frequency). We first observe that the large band corresponds to a device with a low frequency (where the TSopt clock has low resolution (ms)). The width of these bands, and in similar contexts (22, 20), variable network delay renders points in the set. Moreover, as Paxson and others have noted for this line are that, for all

\[ \alpha x + \beta y \leq C, \quad \alpha \geq 0, \beta \geq 0, \quad \sum_{i=1}^{T} x_i = 0, \quad \sum_{i=1}^{T} y_i = 0, \quad |T| \right] \]

Approximate the skew of which has as its core Graham's convex hull algorithm on sorted data [12].

Let us consider plots like those in Figure 1 more closely. The linear programming solution outputs the equation of the line that upper-bounds the set of points. Consequently, to approximate the skew using an upper bound because network and host delays are all different. The linear programming solution then minimizes the average vertical distance of all the points in the set. Moreover, as Paxson and others have noted for this line are that, for all

\[ \alpha x + \beta y \leq C, \quad \alpha \geq 0, \beta \geq 0, \quad \sum_{i=1}^{T} x_i = 0, \quad \sum_{i=1}^{T} y_i = 0, \quad |T| \right] \]

The solution must upper-bound all the points in the set. Therefore, the linear programming method for slope estimation, using the linear programming method for slope estimation, arriving at the same solution since we can prove that, when the above set is the skews are clearly visible; e.g., Figure 1.

The fingerprintee's subsequent ICMP Timestamp Reply messages from the fingerprintee. The measurer must Timestamp Requests (ICMP message type 13) to the fingerprintee. The measurer must be capable of issuing ICMP Timestamp Requests (ICMP message type 13) to the fingerprintee. The measurer could be any website with which the fingerprintee is mountable, the primary limitation is that the device must not be behind a NAT or firewall that filters ICMP.
Security Issues in TCP/UDP

- Network packets pass through/by untrusted hosts
  - Eavesdropping (packet sniffing)
  - Modifications
- IP addresses are public
  - Smurf attacks
  - Anonymity?
- TCP connection requires state
  - SYN flooding
- TCP state is easy to guess
  - TCP spoofing and connection hijacking
Smurf Attack

1 ICMP Echo Req
Src: victim’s address
Dest: broadcast address

Looks like a legitimate “Are you alive?” ping request from the victim

Every host on the network generates a ping (ICMP Echo Reply) to victim

Stream of ping replies overwhelms victim

Solution: reject external packets to broadcast addresses
TCP Handshake

C

S

Listening...
TCP Handshake

C

SYN_C

S

Listening...
TCP Handshake

C
SYN
C
Listening...
Store data
(connection state, etc.)

S
TCP Handshake

C

SYN_C

S

SYN_S, ACK_C

Listening...

Store data
(connection state, etc.)
TCP Handshake

C

SYN

S

SYN, ACK

S

Listening...

Store data
(connection state, etc.)

Wait
TCP Handshake

C
SYN_C
SYN_S, ACK_C
ACK_S
S
Listening...
Store data
(connection state, etc.)
Wait
TCP Handshake

Listening...

Store data
(connection state, etc.)

Wait

Connected

C

SYN<sub>c</sub>

SYN<sub>s</sub>, ACK<sub>c</sub>

ACK<sub>s</sub>

S
SYN Flooding Attack

Listening...
Spawn a new thread, store connection data
... and more
... and more
... and more
... and more
SYN Flooding Explained

- Attacker sends many connection requests with spoofed source addresses
- Victim allocates resources for each request
  - Connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service (DoS) attack
  - Common pattern: it costs nothing to TCP initiator to send a connection request, but TCP responder must allocate state for each request (asymmetry!)