Security News

- US Congress (once again) considering hackback legislation
- Gruss et. al publish advanced ROWHAMMER attacks
Hacking the network

• Attacks at all layers of the network
  – Data-link layer
  – Network layer
  – Transport layer
  – Application layer

• Attacks against all of the properties we care about
  – Availability
  – Integrity
  – Confidentiality
Link Layer Attacks
Sniffing and attacks against confidentiality

- Gathering packets from the local network
  - Passive (wired network with a hub or a wireless network)
  - Active (wired network built with a switch)
- Traffic is not encrypted
Active Sniffing

• Fool the switch into sending the packets to the sniffer
• MAC Flooding
  – Send a flood of traffic with random MAC addresses
  – Fill up the switch’s memory
  – Switches will then forward packets to all links on the switch
• ARP spoofing
  – Send fake ARP replies to change the victim’s ARP table
  – Attacker configures his or her system to forward any traffic it receives to the router.
  – Any traffic from the target machine is sent to the attacker’s machine before being transferred to the local network.
Address Resolution Protocol

• A mapping is needed between IP and physical (MAC) address
• Dynamically established using the Address Resolution Protocol (ARP)
  – Broadcast protocol implemented at the link layer
  – Considered to be a layer 2.5 protocol
  – Used by Ethernet, 802.11, many other link layer protocols with IPv4
• Message types:
  – *who-has* requests – “Who has IP 192.168.0.2?”
  – *is-at* replies – “00:01:2D:00:5F:CC has IP 192.168.0.2”
• Problem:
  – No binding between ARP messages and sender identity
  – In other words, no authentication
ARP Spoofing

CLIENT

Alice .10
MAC: 00:0A:E4:2E:9B:11

LAN: 192.168.1.x

Regular traffic

SERVER

Bob .100
MAC: 00:0A:E4:3B:47:7E

Using arp poisoning

Cracker .1
MAC: 00:22:64:34:60:88

gratuitous arp reply
Bob’s IP → Cracker’s MAC
arpspoof 192.168.1.10 192.168.1.100

victim ip
gateway ip

Alice’s IP → Cracker’s MAC
arpspoof 192.168.1.100 192.168.1.10

victim ip
gateway ip
Network layer hacks
IP Spoofing

- IP Spoofing is an attempt by an intruder to send packets from one IP address that appear to originate at another.
- If the server thinks it is receiving messages from the real source after authenticating a session, it could inadvertently behave maliciously.
- There are two basic forms of IP Spoofing:
  - Blind Spoofing
    - Attack from any source
  - Non-Blind Spoofing
    - Attack from the same subnet
IP Spoofing

• Attacks made possible by IP spoofing include
  – Denial of Service (DOS)
  – Session Hijacking
  – Man in the Middle

• To take over a TCP stream, sequence and acknowledgement numbers must be sniffed or predicted.
TCP RST Attack

- Server port is typically known, client port must be guessed
- Older OSs accepted a wide range of plausible seq/ack numbers with RST
- Modern OSs are more conservative
  - Seq. must be “reasonable”
  - Ack. must be within the sender’s window

Okay, connection closed.
TCP Connection Hijacking

• RST attacks enable DoS, but not packet injection

• Attackers can hijack TCP connections by:
  1. Silencing one participant (A)
  2. Sending spoofed packets to the other participant (B)

• If B accepts a spoofed packet, the connection becomes desynchronized

• Why is it useful to silence one participant?
  – A may RST the connection if they observe a desynchronization or unsolicited packets
**rsh** Connection Hijacking Example

- Remote shell (*rsh*) was the predecessor to *ssh*
- Typically allowed connections from a preconfigured list of "trusted" hosts
- Attacker goal: spoof a TCP connection from a trusted host

Finish login, redirect stdout/stderr to 
/dev/null, create a new account for the attacker, etc.

ACK seq=2 ack=guess
Src: 193.54.34.101:666
Dst: 193.54.34.101:514

72.80.02.01

Internet

Server
84.79.0.1:514

Trusted Host
193.54.34.101

SYN
ACK
Modern TCP

• Modern OSs choose ISNs purely at random
  – This makes off-path hijacking attacks extremely difficult
• Does this mean TCP is now secure from spoofing and hijacking?
  – NO!
  – On-path attackers still see everything, no guessing required
    – May drop, modify, or inject packets at will
• This is why we need IPSEC, TLS, etc.
DDoS Attacks
Attacks against Availability: Denial of Service attacks

• An attempt to consume finite resources, exploit weaknesses in software design or implementation, or exploit lack of infrastructure capacity

• Effects the availability and utility of computing and network resources
Simple DoS

Attacker

Victim

Victim

Victim
Distributed DoS

Attacker

Handler

Handler

Agent

Agent

Agent

Agent

Agent

Victim
DoS History

• Locally-induced crash
  – exploit operating system or server software bug
• Local resource consumption
  – fork() bomb, fill disks, deep directory nesting
• Deny service to individual hosts
  – force crash or outage of critical services
• Remotely-induced crash
  – “magic” packets – ping of death, teardrop
• Remote resource consumption
  – syslog, SYN, fragment flood, UDP storm
DoS History (cont.)

• Deny service to an entire network
  – target vulnerable links or critical network infrastructure / information

• Remotely-induced network outage
  – attacks against routers, DNS servers
  – redirected routes – forged routing information

• Remote network congestion
  – forged directed broadcasts – smurf, fraggle
  – remote control of compromised hosts ("zombies") for coordinated flooding – DDoS
Timeline of a DDoS attack

- A large set of machines are compromised
- Attacker identifies exploitable hosts with scanners, or other techniques
- Attacker accesses the system with automated remote exploits, sniffers, password cracking, worms, trojans
- Attacker installs attack tools
- Attacker remotely instructs compromised machines to attack target
Distributed Denial of Service
Blind spoofing and Smurf Denial of Service (DOS) Attacks

<table>
<thead>
<tr>
<th>SRC</th>
<th>DST</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.3.100</td>
<td>2.2.2.255</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SRC</th>
<th>DST</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.2.*</td>
<td>3.3.3.100</td>
</tr>
</tbody>
</table>

ICMP Echo Request

ICMP Echo Replies

Attacker

Target
Example: TCP SYN Flood

Normal sequence for TCP connection establishment (3-way handshake)
Example: TCP SYN Flood (cont.)

**Attacker**
- SYN 141:141
- SYN 241:241
- SYN 341:341
- SYN 441:441
- SYN 541:541
- SYN 641:641
- SYN 741:741

**Server**
- SYN 182:182
- ACK 142
- SYN 282:282
- ACK 242
- SYN 382:382
- ACK 342

**Listen Queue**
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
- SYN_RCVD
Reflection Attack

- Protocols like TCP allow an attacker to:
  1. Spoof the IP address of the victim
  2. Send a packet prompting a response to a 3\textsuperscript{rd} party
  3. 3\textsuperscript{rd} party sends response to victim
- Can also be done with NTP, DNS, or others
- Can cause \textit{amplification}
DNS Attacks
DNS: Domain Name System

People: many identifiers:
- SSN, name, passport #

Internet hosts, routers:
- IP address (32 bit) - used for addressing datagrams
- “name”, e.g., www.eecs.umich.edu - used by humans

Q: map between IP addresses and name?

Domain Name System:
- distributed database implemented in hierarchy of many name servers
- application-layer protocol host, routers, name servers to communicate to resolve names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network’s “edge”
DNS name servers

Why not centralize DNS?
- single point of failure
- traffic volume
- distant centralized database
- Maintenance doesn’t scale!

• no server has all name-to-IP address mappings

local name servers:
- each ISP, company has local (default) name server
- host DNS query first goes to local name server

authoritative name server:
- for a host: stores that host’s IP address, name
- can perform name/address translation for that host’s name

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Million Queries</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,220</td>
<td>70.4%</td>
</tr>
<tr>
<td>AAAA</td>
<td>206</td>
<td>11.9%</td>
</tr>
<tr>
<td>MX</td>
<td>152</td>
<td>8.8%</td>
</tr>
<tr>
<td>DS</td>
<td>69</td>
<td>4.0%</td>
</tr>
<tr>
<td>NS</td>
<td>25</td>
<td>1.4%</td>
</tr>
<tr>
<td>ANY</td>
<td>19</td>
<td>1.1%</td>
</tr>
<tr>
<td>TXT</td>
<td>18</td>
<td>1.0%</td>
</tr>
<tr>
<td>SOA</td>
<td>6</td>
<td>0.4%</td>
</tr>
<tr>
<td>A6</td>
<td>5</td>
<td>0.3%</td>
</tr>
<tr>
<td>SPF</td>
<td>4</td>
<td>0.3%</td>
</tr>
<tr>
<td>Other</td>
<td>8</td>
<td>0.5%</td>
</tr>
<tr>
<td>Total</td>
<td>1,732</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Top 10 Resource Records Requested
DNS Hierarchical Name Space

unnamed root

generic domains
.com .edu .org

bar
west east
foo my

my.east.bar.edu

country domains
.ac .uk .zw

.ac .uk .zw

ac cam usr

usr.cam.ac.uk

.in-addr

12 34 56

12.34.56.0/24

Top-Level Domain (TLD)
Iterative Name Resolution

Local Name Server

Application

Resolver

cache

google.com

Resolver

cache

Resolver

cache

.(root)

Resolver

cache

.com

Resolver

cache

query

answer

query

answer

query

answer

query

answer
DNS Caching

Step 1: query yourdomain.org

Step 2: receive reply and cache at local NS and host
DNS Caching (con'd)

Step 3: use cached results rather than querying the ANS

Step 4: Evict cache entries upon ttl expiration
Pharming: DNS Hijacking

- Changing IP associated with a server maliciously:

  ![Diagram showing normal DNS, Pharming attack, and phishing where the different websites look the same](image)
DNS spoofing

• Problem
  – No authentication of responses
  – Any DNS response is generally believed.
  – No attempt to distinguish valid data from invalid.
  – Responses can contain entries that should not be trusted but are
  – Responses are cached
  – Just one false root server could disrupt the entire DNS.

• Attacks
  – Inject bogus DNS responses
  – Attach additional bogus entries in valid DNS responses (especially for internal names)
DNS spoofing

Easy to observe UDP DNS query sent to well known server on well known port.

First response wins. Second response is silently dropped on the floor.
DNS Cache Poisoning

• Basic idea: give DNS servers false records and get them cached

• DNS uses a 16-bit request identifier to pair queries with answers

• Cache may be poisoned when a name server:
  – Disregards identifiers
  – Has predictable ids
  – Accepts unsolicited DNS records
DNS cache poisoning

Bell Labs Caching Server

Response
- www.attacker.com A 128.9.128.127
- attacker.com NS ns.attacker.com
- attacker.com NS www.google.com
- ns.attacker.com A 128.9.128.2
- www.google.com A 128.9.128.127

www.google.com = 128.9.128.127

Query
- www.google.com

Any Bell Labs Laptop

Remote attacker
Routing Attacks
Internet inter-AS routing: BGP

• BGP (Border Gateway Protocol): *the* de facto standard

• BGP provides each AS a means to:
  1. Obtain subnet reachability information from neighboring ASs.
  2. Propagate the reachability information to all routers internal to the AS.
  3. Determine “good” routes to subnets based on reachability information and policy.

• Allows a subnet to advertise its existence to rest of the Internet: “I am here”
BGP basics

- Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: **BGP sessions**
- Note that BGP sessions do not correspond to physical links.
- When AS2 advertises a prefix to AS1, AS2 is *promising* it will forward any datagrams destined to that prefix towards the prefix.
  - AS2 can aggregate prefixes in its advertisement
How to Hijack a Prefix

• The hijacking AS has
  – Router with eBGP session(s)
  – Configured to originate the prefix

• Getting access to the router
  – Network operator makes configuration mistake
  – Disgruntled operator launches an attack
  – Outsider breaks in to the router and reconfigures

• Getting other ASes to believe bogus route
  – Neighbor ASes not filtering the routes
  – ... e.g., by allowing only expected prefixes
  – But, specifying filters on peering links is hard
Corrigendum- Most Urgent

GOVERNMENT OF PAKISTAN
PAKISTAN TELECOMMUNICATION AUTHORITY
ZONAL OFFICE PESHAWAR
Plot-11, Sector A-3, Phase-V, Hayatabad, Peshawar.
Ph: 091-9217279- 5829177 Fax: 091-9217254
www.pta.gov.pk

NWFP-33-16 (BW)/06/PTA

February , 2008

Subject: Blocking of Offensive Website

Reference: This office letter of even number dated 22.02.2008.

I am directed to request all ISPs to immediately block access to the following website

URL: http://www.youtube.com/watch?v=o3s8jtvvg00

IPs: 208.65.153.238, 208.65.153.253, 208.65.153.251

Compliance report should reach this office through return fax or at email peshawar@pta.gov.pk today please.
Pakistan Telecom: Sub-prefix hijack

Here’s what should have happened....

“The Internet”

I’m YouTube: IP 208.65.153.0 / 22

Hijack + drop packets going to YouTube

Pakistan Telecom

Telnor Pakistan

Aga Khan University

Multinet Pakistan

Block your own customers.
Pakistan Telecom: Sub-prefix hijack

But here’s what Pakistan ended up doing...

I’m YouTube: IP 208.65.153.0 / 22

No, I’m YouTube! IP 208.65.153.0 / 24
China Telecom: Interception

Paths chosen based on cost and length.
April 2010: China Telecom intercepts traffic

ChinaTel path is shorter

ChinaTel 66.174.161.0/24

ISP 1

Level3, VZW, 22394 66.174.161.0/24

Level 3

China Telecom

Verizon Wireless

This prefix and 50K others were announced by China Telecom

Traffic for some prefixes was possibly intercepted
Canadian Bitcoin hijack (2/3/2014)

• Bitcoin Background
• Miners solve cryptographic puzzles using their computing power
  – Once the puzzle is solved new bitcoins are created and the miner gets some reward
• Mining is generic: mining pool dictates currency
  – E.g., dogecoin, bitcoin etc.
• Miners communicate with pool server using a protocol called ‘stratum’
  – Mining server can redirect user to a different server (e.g., for load balancing)

Canadian Bitcoin hijack (2/3/2014)

- Hijacked users got directed to a mining server IP that was under the control of the hijacker and redirects them to a malicious mining pool.
- Miners continue to receive tasks and solve puzzles but don’t get compensated. 😞

Canadian Bitcoin hijack (2/3/2014)
Canadian Bitcoin hijack (2/3/2014)
Canadian Bitcoin hijack (2/3/2014)
Canadian Bitcoin hijack (2/3/2014)

Estimated loss of $83,000

Approximately 8000 Dogecoins lost due to mining on the hijacker's pool