CNS Lecture 12

Network defenses

firewalls

intrusion detection systems (IDS)

Forensics

Where to encrypt?

link layer
- encrypting modem, NIC (wireless)
- transparent,fast
- suitable for private net
- protects only one link (pt-to-pt)
- info may be exposed in OS

network/transport layer
- swIPe, IPv6(IPsec)
- transparent
- selectable (policy)
- appl./host/net keying
- works over public net
- virtual private network (VPN)
- system layer: encrypting file systems (EFS/CFS)

application layer
- end-to-end over public net
- custom applications (PGP, ssh, ssl)
- intrusive, but flexible
- API for application development

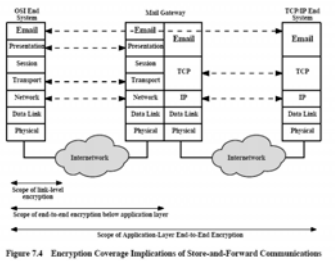


Figure 7.4 Encryption Coverage Implications of Store-and-Forward Communications

Client layout with the BIO API (TCP wrapper)

assignment 9

```
SSL_library_init()
SSL_load_error_strings()
ctx = SSL_CTX_new(SSLv23_method( ))
```
• fetch CA cert's and client cert and private key
```
conn = BIO_new_connect("host:port")
BIO_do_connect(conn)
ssl = SSL_new(ctx);
SSL_set_bio(ssl, conn, conn);
SSL_connect(ssl)
```
• Do any post connection checks on certificate

• Do real work: `SSL_write (ssl, &outpkt, sizeof(outpkt))`
`SSL_read (ssl, &inpkt, sizeof(inpkt));`

SSL on the wire

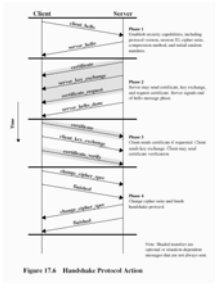
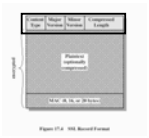
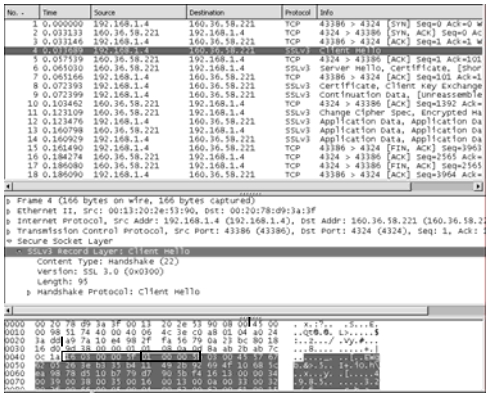


Figure 17.6 Handshake Protocol Action



```
        content type
#define SSL3_RT_CHANGE_CIPHER_SPEC      20  0x14
#define SSL3_RT_ALERT                   21  15
#define SSL3_RT_HANDSHAKE               22  16
#define SSL3_RT_APPLICATION_DATA        23  17
        Handshake
#define SSL3_MT_HELLO_REQUEST                0
#define SSL3_MT_CLIENT_HELLO                 1
#define SSL3_MT_SERVER_HELLO                 2
#define SSL3_MT_CERTIFICATE                 11
#define SSL3_MT_SERVER_KEY_EXCHANGE         12
#define SSL3_MT_CERTIFICATE_REQUEST         13
#define SSL3_MT_SERVER_DONE                 14
#define SSL3_MT_CERTIFICATE_VERIFY          15
#define SSL3_MT_CLIENT_KEY_EXCHANGE         16
#define SSL3_MT_FINISHED                    20
```

You are here ...

Attacks & Defenses
- Risk assessement✓
- Viruses✓
- Unix security✓
- authentication✓
- Network security
  Firewalls,vpn,IPsec,IDS
- Forensics

Cryptography
- Random numbers✓
- Hash functions✓
  MD5, SHA,RIPEMD
- Classical + stego✓
- Number theory✓
- Symmetric key✓
  DES, Rijndael, RC5
- Public key✓
  RSA, DSA, D-H,ECC

Applied crypto
- SSH ✓
- PGP ✓
- S/Mime ✓
- SSL ✓
- Kerberos
- IPsec
- Crypto APIs
- Secure Coding

IP vulnerabilities summary

- denial of service
  - ICMP smurf, redirects, unreachable
  - SYN flooding
  - frag, teardrop
- impersonation
  - host rename (LAN)
  - DNS/ARP cache poisoning
  - source routing
- Session capture
  - TCP seq number guessing
  - TCP hijacking
- server attacks
  - application flooding (ftp,mail,echo)
  - buffer overflows
  - Software bugs

Net attacker MO

- find active hosts (DNS, ICMP broadcasts)
- scan ports (Nessus, nmap, idlescan, SATAN)
- determine OS (nmap/queso/telnet)
  - OS's handle strange packets often in unique ways …
- try exploits (guest/stolen accounts/stack overflows)
- exploit (root shell, shell service to inetd.conf, modify /etc/passwd)
- install hacking tools (root kit)
- clean up logs
- install trojans/sniffer/bot
- review sniffer logs, get accounts/passwords to other systems
- Market your botnet to the bad guys
- tell the world

SANS top 10 ports

Network defenses

Defense in depth

- on a hill
- moat
- outer wall
- archer towers
- inner wall

- disable
- configure properly
- xinetd, tcpwrappers
  - filters (allow, deny)
  - audit and alarm
- filtering portmap
- application filtering (securelib)
- patches
- scanners (Nessus, SATAN, ISS)
- firewalls
- intrusion detection & response
- encryption, IPsec, virtual private networks (VPNs)

Assess your attack surface

Scanners

ISS, Nessus,nmap -- probe and report network hosts and services

- point, click, scan a net
- port probes (nmap)
- OS type probes (nmap)
- portmap probes
- X and NFS attempts
- sendmail checks
- NIS probes

Nessus demo

Network countermeasures

- host-based (wrappers, personal firewalls)
- router based (filters)
- firewalls
- Intrusion Detection Systems (IDS/IPS)
- authentication/encryption (IPsec/VPNs)

Host network services "wrappers"

- network/host access control lists
- re-write applications with filters (securelib)
- replace inetd with filtering version (xinetd)
- use tcp wrappers
  - free, no changes to application
  - inetd services only
  - allow/deny
  - double DNS lookups
  - audit and alarm
  - API for new app's

```
/etc/inetd.conf
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  wu.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
shell   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rshd -L
login   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind

/etc/hosts.deny
in.rlogind: ALL
in.telnetd: ALL
in.rshd: ALL
wu.ftpd: ALL

/etc/hosts.allow
in.rlogind: 128.219., 134.167., 127.
in.telnetd: 128.219., 134.167., 127.
wu.ftpd: 128.219., 134.167., 127.
in.rshd: 128.219., 134.167.
```

firewalls

- NO connection -- best ☺
- toolkits, personal firewalls (Linux, PC)
- filtering/screening routers
- dual-homed gateways (bastion host)
- screened host gateway
- screened subnet (NAT)
- commercial solutions (enterprise firewalls)

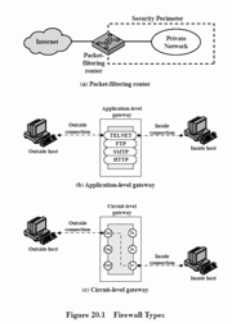


Figure 20.1 Firewall Types

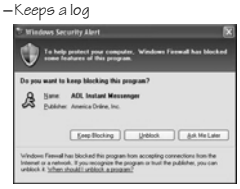
Personal (host) firewalls

- Network access control lists
  - –Which hosts/nets you permit/deny
  - –Which services you permit/deny
  - –Make your host invisible to net (ping/port scans)
- PC/Windows – XP firewall (ICF), ZoneAlarm, Netice
- Linux – iptables
- MAC – ipfw

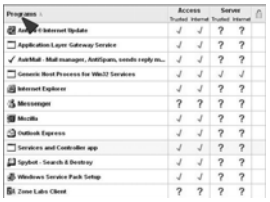
Difficult to configure and EVERY host needs to do it.
If bad guy gets in to your host, he'll disable your host's firewall.

Windows XP firewall

- SP1, ICF (properties of LAN connection)
  http://www.itc.virginia.edu/netsys/security/firewalls/xp-firewall.html
- SP2, Security Center (firewall, auto updates, viruses)
  - –Blocks outside requests
  - –Alerts if program attempts to use Internet
  - –Add exceptions (program or port)
  - –Keeps a log



zonealarm



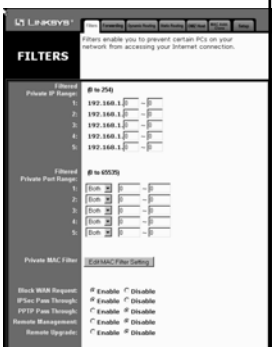
Linux firewalls

- ipfwadm begat ipchains begat iptables
- accept/reject rules (tables) + logging
- RedHat select security (high, medium, none)
- provides Network Address Translation (NAT), masquerading
  - –IP forwarding (private nets 10.0.0.0, 172.16.0.0,192.168.0.0)

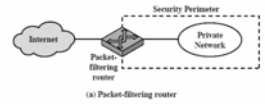
```
iptables -F iptables -A INPUT -i lo -p all -j ACCEPT - Allow self access by loopback interface
iptables -A OUTPUT -o lo -p all -j ACCEPT
iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT - Accept established connections
iptables -A INPUT -p tcp --tcp-option ! 2 -j REJECT --reject-with tcp-reset iptables -A INPUT -p tcp -i eth0 --
dport 21 -j ACCEPT - Open ftp port
iptables -A INPUT -p udp -i eth0 --dport 21 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 22 -j ACCEPT - Open secure shell port
iptables -A INPUT -p udp -i eth0 --dport 22 -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT - Open HTTP port
iptables -A INPUT -p udp -i eth0 --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --syn -s 192.168.10.0/24 --destination-port 139 -j ACCEPT - Accept local network Samba
connection
iptables -A INPUT -p tcp --syn -s trancas --destination-port 139 -j ACCEPT
iptables -P INPUT DROP - Drop all other connection attempts. Only connections defined above are allowed.
```


Home protection

- Personal PC firewalls (ZoneAlarm, iptables)
- DSL/Cable
  - –Inexpensive router, NAT, firewall
  - –Home network with perimeter protection
- Wireless
  - –Enable 128-bit WEP key
  - –Accept only designated ether addresses (MAC filter)
  - –Disable SSID broadcast
  - –Use ssh or VPN
- Review logs



Screening routers



(a) Packet-filtering router

- router's job is to forward packets (fast)
- add filters (ACL's) for each interface
- can block IP address spoofing of internal addresses
- should permit out only legit. local addresses
- may deny/restrict specific services (ports)
- weaknesses
  - complicated filter expressions
  - may fail to the open mode
  - limited logging
  - no authentication
  - DNS spoofing

Port deny list:

portmap, tftp, snmp, syslog, telnet

Restrict http to designated servers

Screening routers -- rules

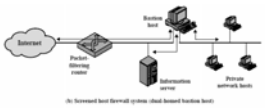
```
! access list  102 specifies what addresses are allowed out
access-list 102 deny   ip 128.219.250.0 0.0.1.255 0.0.0.0 255.255.255.255
! no snmp out
access-list 102 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 162
access-list 102 permit ip 128.219.0.0 0.0.255.255 0.0.0.0 255.255.255.255
access-list 102 permit ip 134.167.0.0 0.0.255.255 0.0.0.0 255.255.255.255
access-list 102 permit ip 192.12.68.0 0.0.0.255 0.0.0.0 255.255.255.255

! access list  112 denies local addresses from the outside
access-list 112 deny ip 128.219.0.0   0.0.255.255 0.0.0.0 255.255.255.255
access-list 112 deny ip 134.167.0.0   0.0.255.255 0.0.0.0 255.255.255.255
access-list 112 deny ip 192.12.68.0   0.0.0.255   0.0.0.0 255.255.255.255
! block a known bad guy
access-list 112 deny ip 130.225.220.16 0.0.0.0 0.0.0.0 255.255.255.255
! deny remote SNMP's and tftp's
access-list 112 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 161
access-list 112 deny   udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 69
! special internal hosts
access-list 112 deny   ip 0.0.0.0 255.255.255.255 128.219.250.0 0.0.1.255
```

other rules for what routes are advertised

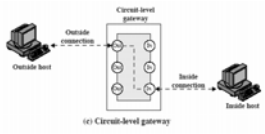
Bastion host

dual-homed gateway

- host with two network interfaces
- IP forwarding disabled
- reachable from either side, but packets do not flow from one side to the other
- user must login to bastion host, then to other side
- supplement with application gateway software (email, ssh)
- strong authentication (Securid), logging (hardened host)
- limited services (restricted shell), wrappers
- custom mail programs
- hides enterprise network (private IP addresses)



(b) Screened host firewall system (dual-homed bastion host)

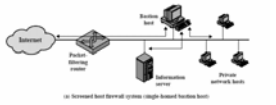


(c) Circuit-level gateway

Screened host/net

Screened host

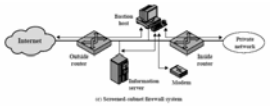
- common implementation
- traffic to/from Internet allowed only to bastion host, though can let internal hosts access some Internet services (ssh, ftp, www)
- bastion host acts as application gateway



(a) Screened host firewall system (single-homed bastion host)

Screened subnet (DMZ)

- two screening routers
- one or more bastion hosts on subnet
- internal net can be private (invisible), network
- address translation (NAT)
- place some servers on DMZ (www, anon ftp)
- place intrusion detectors, traps on DMZ
- place external DNS on DMZ

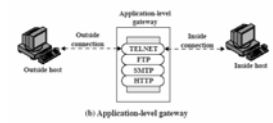


(c) Screened-subnet firewall system

Application gateways

proxy services on the bastion host

- run minimal services (trusted OS?)
- no compilers, linkers
- use wrappers
- no local logins
- custom servers (minimal pkt forwarders, logging, ACLs)
- connections from outside
  - strong authentication (skey, securID)
  - encrypted (ssh, stel)
  - user then connects to internal host and logs in again
- 2-part mail forwarder/scanner (IPS)
  - Remove evil attachments
  - Block spam

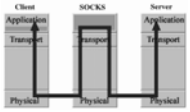


(b) Application-level gateway

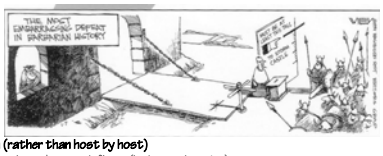
Proxy servers

internal hosts accessing the outside ("relay")

- need socks-ified local applications
  - #define connect Sconnect
- proxy on bastion host (tn-gw, rlogin-gw, ftp-gw, x-gw, http-gw)
- servers are simple packet forwarders with ACL, e.g., telnet and an itelnet
- some services support proxies (netscape, gopher)
- socks library for building your own local apps



Enterprise firewalls



router with an attitude
- establish a perimeter
- single point of protection (rather than host by host)
  - Controls inbound and outbound network flows (by hosts/service)
  - logs
- principle of layering, reference monitor
  - always invoked
  - tamper resistant
  - small and simple (understandable)

Enclaves

use firewalls and VLANs to create internal protected subnets (e.g., business subnet, medical subnet,)

establish a policy
- What's not denied is allowed
- What's not allowed is denied -- best
- Complicated rules

security vs convenience

Firewall limitations

What they don't do?
- don't do UDP very well
- don't prevent session hijacking
- don't provide privacy
- don't protect against virusses
- don't protect against insider (need internal firewalls/enclaves)
- don't prevent backdoors (modems, VPNs/tunnels)
- don't log/alarm like an IDS
- don't improve throughput!

Watch out for tunneling through "permitted" ports (trojan horse)

Selecting an enterprise firewall

- commercial, consultant, kit
- filters for both in and out
- filter granularity (stateful, ftp support)
- IP fragment management
- filter language and user interface
- proxy applications, clients, extensible
- authentication mechanisms
- network address translation (NAT) and VPN
- integration with intrusion detection (IPS)
- IPv6
- logging and audit tools
- ease of install and use
- performance
- cost

Intrusions

Prevention
- host based -- wrappers, patches, strong authentication
- net based -- filtering routers, firewalls
  an intrusion may still occur

Detection & Response
- detect quickly, limit damage
- detection information can be used to strengthen prevention
- based on assumption that behavior of intruder differs from norm -- e.g., credit card mis-usage

Safes are rated by "time to crack".

You can buy a cheaper safe, if you alarm, and your police are quick.

Intrusion Detection

Intrusion –
any set of actions that attempts to compromise integrity, privacy, or availability of a resource

- objective: detect and report/alarm
- techniques
  - –real-time/batch
  - –statistical profiles (user, system, network) –deviations from the norm
  - –attack signatures -- known to be bad
- host-based, net-based
- components
  - –sensors
  - –knowledge base (rules, norms)
  - –audit logs for establishing norms and detecting ab-norms, accurate timestamps (NTP)
  - –decision modules
  - –reporting/action modules
  - –adaptive -- update knowledge base

Statistical profiles

detecting anomalous behavior
- establish a norm
- train the detector
- establish alarm thresholds
- pattern matching, AI, neural networks
- system logs, special logging
- continuous learning

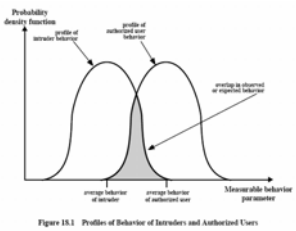


Figure 18.1 Profiles of Behavior of Intruders and Authorized Users

Statistical profiles

user profiles

- login times, durations
- login location
- resource usage (files, CPU time)
- process usage (mail, compile)

host profile

- load average/time-of-day
- process count
- logins/hour
- process usage (mail, compile), time-of-day
- process profiles (system calls, file accesses)

IDS norms

Table 18.1 Measures That May Be Used for Intrusion Detection

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
File access activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.

Attack signatures

batch and real-time

- virus scanners (actively remove malware attachments)
- login/su failures
- application logs
- file/file-mode changes (tripwire)
- sendmail, fingerd, tftp,snmp attacks
- host impersonation
- Rule-based intrusion events:
  - Copying system files
  - Reading from devices (rather than files)
  - Concurrent logins
  - Reading other people's directories

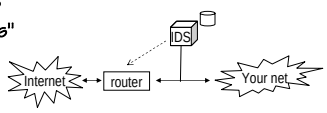
IDS host tools

- logs (lastlogin, messages, syslog, wrappers, NT event logs)
- process accounting
- maybe ACLs and logs
- log watchers (watcher, swatch)
- recent attacks (CERT)
- cpm, tripwire
- Anti-virus software, mail checker
- periodic checksum of home page

need them on <u>every</u> host!

Network IDS

- LAN or perimeter detectors
  - Sniffers
  - Passive or optional router control
  - Login banner to avoid privacy violations
- use in conjunction with host detectors
- collect packet traffic summaries
  - Forensics: reconstruct attack from logs
- construct norms
- detect "ab-norms"



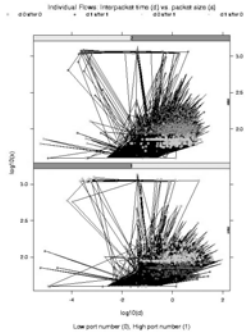
Statistical profiles

network profile

- typical load (pkts/sec/time-of-day)
- most active hosts
- typical services (volume, time-of-day)
- host interconnect patterns (who, when, how long)
- host service/port profiles
- Flow statistics (packet size, rate, interarrival, bursts,duration)
  - Not looking at contents of packet (because of volume or encryption)
  - Detect type of flow (interactive, web, email, streaming, chat…)

Flow characterization

Firewalls allow only certain services to flow. Hackers often will trojan an allowed service, e.g., use port 80 to carry ssh traffic



Two flows from a compromised host

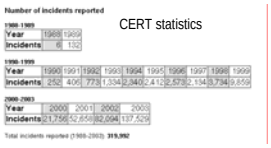
Can you characterize a flow (mail, telnet/ssh, www, chat) based on flow stats (interarrival rate packet size, volume, duration)?

Attack signatures

batch and real-time

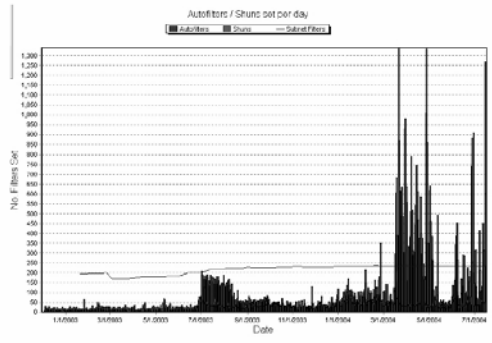
- denial of service
- host impersonation
- port scans
- "known" backdoor ports
- source routing
- "big" packets (buffer overflows)
- content scanning -- keystrokes, viruses

CERT statistics



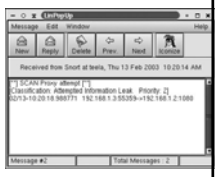
There is a lot of "door-knob rattling" – hundreds of probes per day …

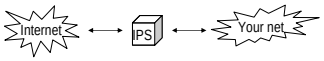
ORNL intrusion prevention



Network IDS tools

- logs (wrappers)
- Alarms (alarmed ports)
- honeypots (honeynet.org)
  - host with alarmed services/ports
  - present fake net vulnerabilities
  - hold their attention
  - deception toolkit
  - How long before a new host is probed?
- tcpdump, sniffers, Ethereal
- RealSecure, NIDS, NetRanger, Bro, Snort
- interface with firewalls/filtering routers for IPS
- IPS – also modify/delete content (enterprise mail)





IDS limitations

- expensive, complex
- must be secure
- knowledge of vulnerabilities, expert systems
- site specific (internal IDS?)
- no common audit/log format
- need for aggregating reports
- labor intensive – you must have a response component (IPS)
  "If you bring a Ranger with you, it is well to pay attention to him."
  -- J.R.R. Tolkein, The Fellowship of the Ring
- Auto-response may result in 2nd order denial of service
- large files (processing time, archiving)
- can be fooled (out of order packets, but TCP sequence numbers re-order)
- Can be overloaded (DoS)
- False alrms vs missing intrusions
- difficult to test and compare

IDS research

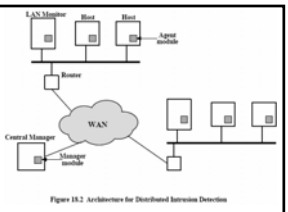


Figure 18.2 Architecture for Distributed Intrusion Detection

distributed IDS
- standard report format
- sensors (host,net), autonomous agents
- central manager
- enterprise aggregation
- country aggregation

- higher speed nets too fast -- host IDS become more important, parallel, FPGA
- resisting DoS
- backtracking spoofed packets
- providing attack signatures and out-of-band info
- adaptive detectors
- active response (IPS) -- adding filters and blocking, removing attachments, counter-attack?
- shunting decisions – let pass, shunt (black hole, honey pot, neuter & pass)
- content monitoring and privacy – encryption (key escrow?)
- data mining
- detecting covert channels
- evading detection (Ptacek)
- OS/CPU diversity to reduce vulnerability, immunology models

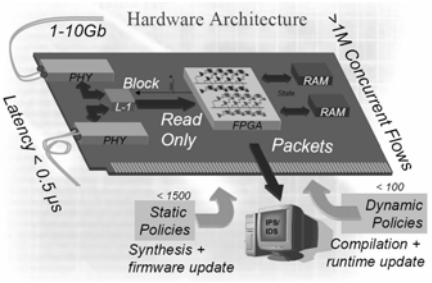
Backtracking spoofed IP address flows

- Spoofed IP source addresses used by Denial of Service and session hijacking
- Perimeter routers SHOULD block spoofed addresses
  - Don't allow internal addresses as source address from external interfaces
  - Only allow packets with valid source addresses out
- For an active attack that is using spoofed IP source addresses
  - Manually check each router along the flow, backtracking
  - Automated program to access routers and backtrack flow and setting filter to block
  - Hard: crosses administrative domains
- Other approaches, marking packets, new ICMP … open research

High speed IDS (10gbs)

Metanetworks.com

- uses SNORT signatures
- process data stream concurrently with FPGA



Pursuit

Response -- following up an intrusion

- policy/procedures (check list)
  - Identify
  - Isolate
  - Evaluate
  - Remediate
  - monitor
- Incident response team
  - Technical staff (network/computer)
  - Security staff
  - Legal, public relations
- Contact
  - attacking host/ISP
  - management
  - FBI/CERT/CIAC
- Collect evidence (forensics)

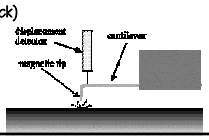
forensics – cyber CSI

- Preserving evidence of the attack
- Determining how, who, when, where, why
- Damage assessment
  - What's been taken, modified, added, deleted
- Forensic tools (for Windows/Mac/Unix)
  - Sleuthkit www.sleuthkit.org
  - TCT (The Coroners toolkit), FTK
  - FIRE fire.dmzs.com
  - Encase
  - Password crackers
  - Net and keyboard sniffers
  - Bootable CD's or remove disks and take to your forensic lab
- Forensics applicable to internal waste/fraud/abuse

Good job opportunity!

Forensics – target site

- preserving evidence
- disconnect from net -- don't reboot? Open debate…
- Analyze/record system status/disks
- Review IDS/firewall logs – attacking host "trail"
- observing chain of custody
  - Mark and log evidence (floppies, dongles, hard drives, etc)
  - Hash and sign digital evidence (use time-stamp notary service too)
- Image-disk copies
  - Maybe keep media, can recover over-written bits !!
- audit trails (accurate clocks? IDS, attacked machine, etc.)
- record of time spent (part of cost of attack)
- getting back online



Network IDS logs for forensics

- Record all packet headers (lots of disk space!)
  - Handy for daily monitoring, statistical profiles
  - Useful for reconstructing an attack
- Follow packet trail of attacking host
  - What protocols did he use
  - Which internal hosts were visited
- Packet trail of attacked host
  - What hosts did it visit before/during/after attack
  - 2nd order, what hosts did those hosts visit ….
- Post attack monitoring
  - Remote hosts visiting attacked host (especially on a backdoor port)
  - Additional traffic from attacking host (or do you have firewall block?)
  - Traffic from attacked host (cleanup incomplete?)

Are system logs admissible evidence?

- Easy to forge computer logs
- Hacker may have tampered with logs
- Computer records are considered *hearsay*
- However *business records* are acceptable
  - So sys logs acceptable if being collected as part of day-to-day ops
  - Must be able to attest to their authenticity (logged to secure machine) (chain of custody, time-stamped MD5)
- Logs started after the attack, probably not admissible, but you may get clues from these logs that leads you to admissible evidence

Forensics – attacker's site

- court orders
  - Wiretap/sniff
  - Keystroke capture (get passwords)
- confiscation of equipment
  - Log evidence collected, maintain chain of custody
  - Look for post-its etc. with passwords
- preservation of evidence (use checklists or call the cops)
- disk analysis
  - Make bit image copies of drives
  - Hash/sign and digital notarize log files and media images
  - Use tookits to look for key words/evidence in files
    - Hidden files (stego?)
    - Deleted files
    - Compressed/encoded files
    - Encrypted files
  - Evaluate executables (disassemblers)
- Check time offset of hacker's PC clock
  - Establish time-line of events, file modifications, system logs
- prosecution and trial

Hacker's guide

- Dial-in (stolen aol) etc.
- Login through a series of compromised machines
- Attack new set of machines
  - Use accounts/passwords
  - Portscan/ OS scan (nmap)
  - Try net exploits (buffer overflows)
- Once you get on:
  - Download exploits and rootkit
  - Get root, install rootkit, cover your tracks
  - Grab passwd/shadow, other interesting stuff (.rhosts, credit card numbers)
  - Check syslog.conf, changed files to see if anyone is watching
  - Install backdoor, sniffer, bots, trojan ssh
- Brag about it on your favorite hacker chat group

"Input validation is for people who can't do forensics."

in situ analysis

- risk in getting on and analyzing the attacked or attacking engine
  - Alert the attacker
  - Altered commands (booby traps) could delete evidence (zero the disk) and crash machine
  - Alter the evidence (Heisenberg principle)
  - Can't trust the data (root kit)
  - mount your CD with your tools and make it only thing in PATH
  - Disconnect from the net
- Can learn a lot
  - State of registry, active network ports, active processes (strace/ltrace), open files, process memory, swap space, who's logged in, shell history files (disk encryption/bitlocker)
  - Keep a record of what you do/find (script)
- At least (safe), if you don't have IDS logs, **start external sniffers** and monitor traffic from suspect engines – though leaving suspect engines running increases risk of more damage
- Information volatility: cache→registers→display frame buffer→RAM (process/kernel/network state) →swap space→spool space→ temp →syslog→disks→printouts

Forensic tools

- Your forensic lab needs PC with various tools and hardware to attach/copy various media (CD/DVD/floppy/USB/SCSI/ATAPI…)
- Bootable CD's (trusted media) alternative for image-copying disks in the field
- Forensic hardware
  - Portable forensic PC
  - hard drive connectors (firewire/usb to ATA/IDE write-block)
  - tape/DVD backup

- Support for various file systems
  - linux, BSD, DOS, Windows, MAC, RAID
  - Examine unallocated blocks, deleted files, slack space, swap space
  - Directory lister (dates), image viewers
  - grep, nm, strings, lsof, ldd, file,find, dd, netcat, md5sum
  - Hash/sign images
  - Dissasemblers, uncompress, decode, decrypt (password crackers)
  - Registry/log file analyzers
- Hashes of "good" versions of executables, libraries, data (html)
  - Hash verifier tool, rpm -Va

Forensic tools

The Coroner's toolkit

- Free, linux-based
- Grave-robber – main data collector
- Lazarus – data reconstructor
- Mactime – file time (M A C) reporter
- pcat – display process memory
- unrm – recover deleted files/blocks
- file – file type checker
- ils/icat – list/cat by inode (deleted)
- Handles windows file systems too

Encase (also see FTK)

- Popular commercial tool
- Disk imaging/hashing/restore
- Parallel search
- Remote diagnosis (servlets)
- Most file systems (linux, windows,…

UNIX forensics

- *in situ* – mount your CD, make it only thing in PATH
- Use dd and netcat to copy disk images to trusted host
  - On trusted host: `nc -l -p 10000 > disk1.img`
  - On suspect host:

```
dd bs=1024 < /dev/ad0s1e | nc 192.168.0.4 10000 -w 3
```

also could `dd /dev/kmem` and `/dev/mem` to forensic host

On trusted host:

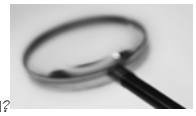
```
md5sum disk1.img > disk1.md5
mount -t ext2 -o ro,loop=/dev/loop0 disk1.img /mnt/badboy
find /mnt/badboy -type f -print0 | xargs -r0 file | grep
 executable
```


Malware analysis

- Mystery executable
- Use strings, nm, ldd to peek inside
- Maybe disassemble & reverse engineer
- Careful – only run it on disposable machine/OS (VMware) , then restore
  - Run it with strace to see system calls
  - Use lsof to see what files/ports it has open
  - Debugger to single-step
  - pcat to dump process memory or /proc/nnn
  - kill -5 to dump core

Assignment 10

Windows malware analysis



- Figure out what a bad .exe does?
  - What files/registry entries does it modify/steal?
  - Capture keystrokes?
  - Talk on the net?
- Be safe -- VMware and/or private net with disposable CPUs
- Tools
  - Regmon, regshot
  - Process explorer, PEiD, PEview
  - UPX
  - Filemon
  - Tcpview, fport
  - Ollydbg or IDA pro
  - Netcat tcpdump/ethereal
  - Google (known malware)

What you really want to know is how they got in?
registry snapshots (XP restore points), logs, IDS logs

UNIX intrusion response

Action	Expertise	Time
Go back to work	None	1+ hours
Minimal work	Anyone who can install	.5 to 1 day
Minimum recommended	Junior sys admin	1 to 2 days
Serious effort	Sys admin	2 days to 2 weeks
Fanaticism	Forensic specialist	Weeks to months $$

Sophistication of attack:
- account/password
- buffer overflow for network daemon
- root access
- root kit (hiding tracks)
- backdoors/trojans/sniffers/bots
- self-re-installing or self-destruct
- physical access (hardware mods, keyboard sniffer)

US security/privacy laws

- Computer fraud & abuse act (CFAA) – computer access
- Gramm-Leach Bliley act (GLBA) – financial data
- Health information portability accountability act (HIPAA)
- Children's online privacy protection act (COPPA)

Legal morass

- federal laws (computer fraud and abuse act)
  - unauthorized access
  - data theft (trade secrets, copyright,passwords)
  - unauthorized modifications
  - porn
  - cyber stalking
  - search & seizure (wiretaps, sniffers)
- some state/local laws
- judiciary not trained in computer crime
- victims reluctant to report crimes
- value of loss (information, service disruption,…)
- jurisdictional/territorial problems
- Defense may be based on inadequate handling of evidence!
  - Cross-examination of the sys admin … "you did what!"
- convincing a jury … digital evidence, tangible loss

Sentencing guidelines

- Potential/actual loss
- Level of sophistication of attack
- For commercial or personal benefit
- Malicious intent
- Messin' with national defense, national security, justice
- Messin' with critical infrastructure
- Threat to people, public health

Detection & response is as important as prevention!

recall

Cost-benefit analysis for the attacker (Clark & Davis '95)

$$M_b + P_b > O_{cp} + O_{cm}P_aP_c$$

M_b monetary benefit to attacker
P_b psychological benefit to attacker
O_{cp} cost of committing the crime
O_{cm} cost of conviction to the attacker
P_a probability of arrest
P_c probability of conviction

Becoming a certified crypto geek

- SANS/GIAC
  security administrator, management, operations, network engineer, sysadmin, legal, audit, security expert
- Certified Information Systems Security Professional (CISSP)
  - 250 question exam
  - security mgt, architecture, access control, application development, operations security, physical security, crypto, network security, continuity planning, ethics/laws, forensics
- CIW security analyst
- Cyber-investigator certification
  Encase, AIS, CCE, CIFI, SSCP, Cisco, RSA

These credentials may qualify you as an "expert" witness

Next time …

IPsec
Kerberos
Trusted systems and secure OS

	Lectures
1.	Risk, viruses
2.	UNIX vulnerabilities
3.	Authentication & hashing
4.	Random #s classical crypto
5.	Block ciphers DES, RC5
6.	AES, stream ciphers RC4, LFSR
7.	**MIDTERM** ☹
8.	Public key crypto RSA, D-H
9.	ECC, PKCS, ssh/pgp
10.	PKI, SSL
11.	Network vulnerabilities
12.	Network defenses, IDS, firewalls
13.	IPsec, VPN, Kerberos, secure OS
14.	Secure coding, crypto APIs
15.	review