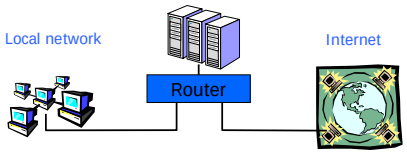
CS 155 May 24, 2005

Firewalls and Intrusion Detection

John Mitchell

Common network devices

- Packet and Application-Layer Firewall
- Network Intrusion Detection
- Virtual Private Network (IPSEC/PPTP/SSL)
- Content Filtering and Virus Scanning
- Bandwidth Management (Traffic Shaping)
- Web caching, other caching

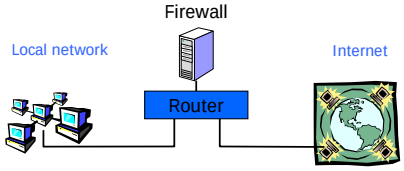


Topics

- Firewalls
  - Packet filter (stateless, stateful)
  - Application-layer gateway
- Traffic Shaping
- Intrusion detection
  - Anomaly and misuse detection
  - Host and network intrusion detection

Basic Firewall Concept

- Separate local area net from internet



All packets between LAN and internet routed through firewall

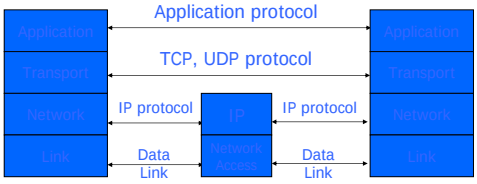
Why firewalls?

- Need to exchange information
  - Education, business, recreation, social and political
- Program bugs
  - All programs contain bugs
  - Larger programs contain more bugs!
  - Network protocols contain;
    - Design weaknesses (SSH CRC)
    - Implementation flaws (SSL, NTP, FTP, SMTP…)
  - Careful (defensive) programming & protocol design is **hard**
- Defense in depth

Two Separable Topics

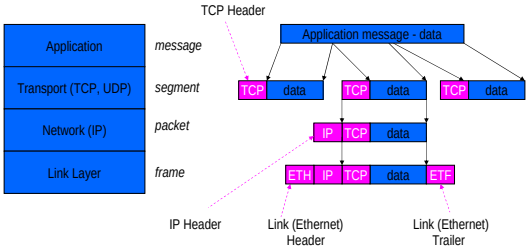
- Arrangement of firewall and routers
  - Several different network configurations
    - Separate internal LAN from external Internet
    - Wall off subnetwork within an organization
      - Test networks, financial records, secret projects
    - Intermediate zone for web server, etc.
  - Personal firewall on end-user machine
- How does the firewall process data
  - Packet filtering router
  - Application-level gateway
    - Proxy for protocols such as ftp, smtp, http, etc.
  - Circuit-level gateway
  - Personal firewall also knows which application
    - E.g., disallow telnet connection from email client

TCP Protocol Stack

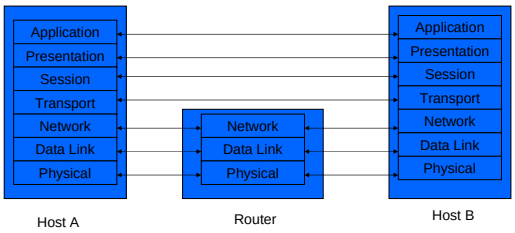


Transport layer provides *ports*, logical channels identified by number

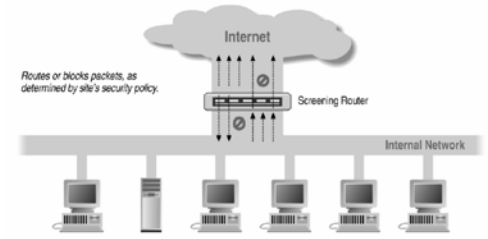
Data Formats



ISO OSI 7 Layer Network Reference Model



Screening router for packet filtering



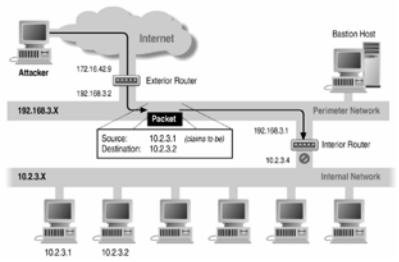
Packet Filtering

- Uses transport-layer information only
  - IP Source Address, Destination Address
  - Protocol/Next Header (TCP, UDP, ICMP, etc)
  - TCP or UDP source & destination ports
  - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - ICMP message type
- Examples
  - DNS uses port 53
    - No incoming port 53 packets except known trusted servers
- Issues
  - Stateful filtering
  - Encapsulation: address translation, other complications
  - Fragmentation

Packet filtering examples

Compare: Tiny Personal Firewall, ZoneAlarm

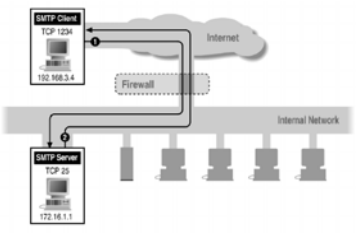
Source/Destination Address Forgery



Port numbering

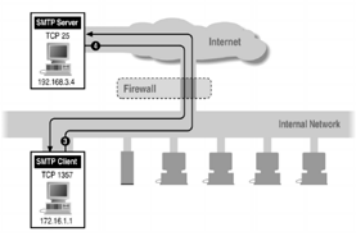
- TCP connection
  - Server port is number less than 1024
  - Client port is number between 1024 and 16383
- Permanent assignment
  - Ports <1024 assigned permanently
    - 20,21 for FTP
    - 23 for Telnet
    - 25 for server SMTP
    - 80 for HTTP
- Variable use
  - Ports >1024 must be available for client to make any connection
  - This presents a limitation for stateless packet filtering
    - If client wants to use port 2048, firewall must allow *incoming* traffic on this port
  - Better: stateful filtering knows outgoing requests
    - Only allow incoming traffic on high port to a machine that has initiated an outgoing request on low port

Filtering Example: Inbound SMTP



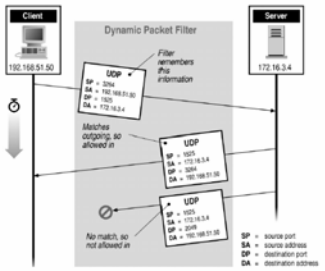
Can block external request to internal server based on port number

Filtering Example: Outbound SMTP

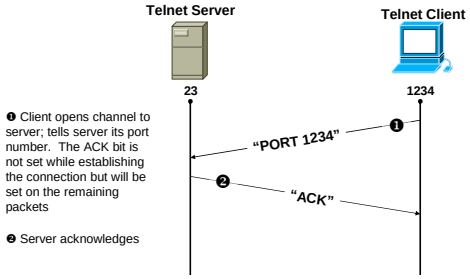


Internal request to external server will use known port out, high port in

Stateful or Dynamic Packet Filtering



Telnet

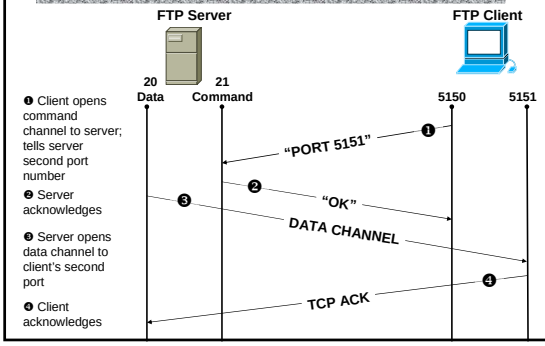


❶ Client opens channel to server; tells server its port number. The ACK bit is not set while establishing the connection but will be set on the remaining packets

❷ Server acknowledges

Stateful filtering can use this pattern to identify legitimate sessions

FTP



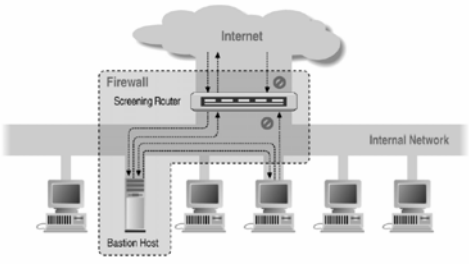
Normal Fragmentation

Abnormal Fragmentation

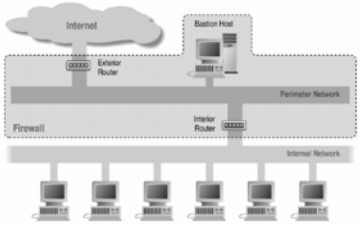
Application-level proxies

- Use "bastion host"
  - Computer running protocol stack
  - Several network locations – see next slides
  - Will interact/accepts data from the Internet
    - Disable all non-required services; keep it simple
    - Install/modify services you want
    - Run security audit to establish baseline
    - Be prepared for the system to be compromised
- Enforce policy for specific protocols
  - E.g., Virus scanning for SMTP
    - Need to understand MIME, encoding, Zip archives

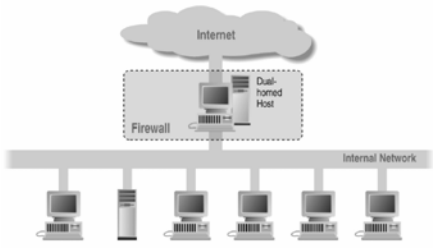
Screened Host Architecture



Screened Subnet Using Two Routers



Dual Homed Host Architecture



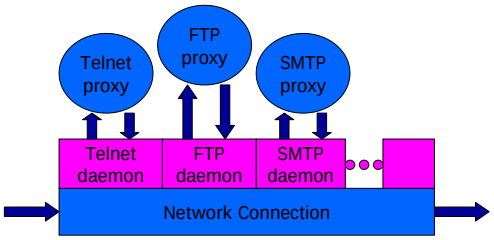
Firewall mechanism

- Firewall runs set of proxy programs
  - Proxies filter incoming, outgoing packets
  - All incoming traffic directed to firewall
  - All outgoing traffic appears to come from firewall
- Policy embedded in proxy programs
- Two kinds of proxies
  - Application-level proxies
    - Tailored to http, ftp, smtp, etc.
  - Circuit-level proxies
    - Decisions based on header information

Proxies

- Application level; dedicated proxy (HTTP)
- Circuit level; generic proxy
  - SOCKS
  - WinSock – almost generic proxy for Microsoft
- Some protocols are natural to proxy
  - SMTP (E-Mail)
  - NNTP (Net news)
  - DNS (Domain Name System)
  - NTP (Network Time Protocol)

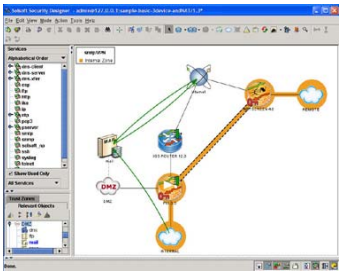
Firewall architecture



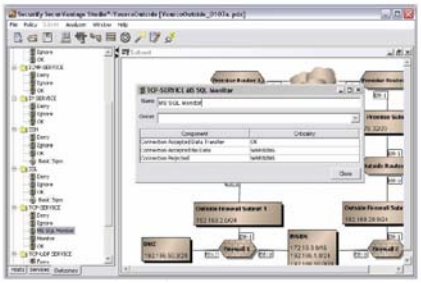
Daemon spawns proxy when communication detected ...

Configuration issues

Solsoft



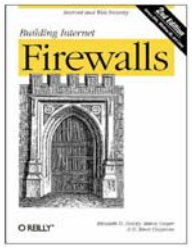
Securify



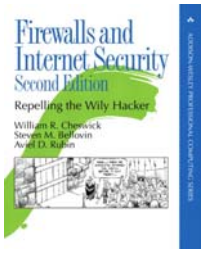
Problems with Firewalls

- Performance
  - Firewalls may interfere with network use
- Limitations
  - They don't solve the real problems
    - Buggy software
    - Bad protocols
  - Generally cannot prevent Denial of Service
  - Do not prevent insider attacks
- Administration
  - Many commercial firewalls permit verycomplex configurations

References



Elizabeth D. Zwicky
Simon Cooper
D. Brent Chapman

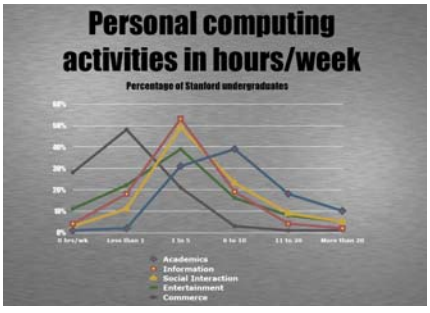


William R Cheswick
Steven M Bellovin
Aviel D Rubin

Traffic Shaping

- Traditional firewall
  - Allow traffic or not
- Traffic shaping
  - Limit certain kinds of traffic
  - Can differentiate by host addr, protocol, etc
  - Multi-Protocol Label Switching (MPLS)
    - Label traffic flows at the edge of the network and let core routers identify the required class of service
- The real issue here on Campus:
  - P2P file sharing takes a lot of bandwidth
  - 1/3 of network bandwidth consumed by BitTorrent
    - And I think you know what BitTorrent, Gnutella, Kazaa, ... are used for

Stanford computer use

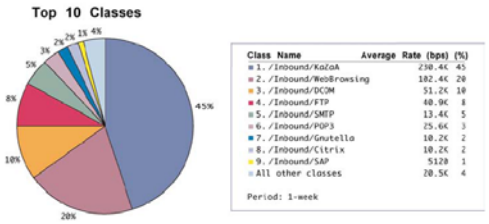


Stanford file-sharing policy?

- ... Stanford University caught heat two years ago when it set up a server to manage requests for music files on the popular Gnutella file-sharing service. The IT department's goal was to cut down on requests leaving the campus by directing queries internally, to PCs in the dorms, thus easing the strain that music files were placing on external links to the Internet. But the MPAA complained that the server effectively handed students a tool to violate copyright laws, and the university shut it down after six months, recalls Richard Holeton, Stanford's head of residential computing.
- Now Stanford relies on traffic shaping alone; the university has no plans to impose additional restrictions. There's "nothing illegal" about using the protocols associated with P2P file sharing, says Holeton, who calls UF's policy draconian.
- "To me, to use any kind of network-management tool to identify somebody who might potentially be doing something is kind of Big Brotherish," Holeston adds. "It's like pulling over everybody on the highway who is driving a certain kind of car that could potentially be breaking the law, and giving them a ticket."
- Says Fred von Lohmann, a senior staff attorney at the Electronic Frontier Foundation: "If John Ashcroft asked us to do this, we'd be crying foul, but the recording industry does it and we roll over."
- A spokesman for the RIAA wouldn't disclose how many universities have been subpoenaed for names of students, but he did say, "Virtually every university has complied."

Feb 19, 2004

Sample

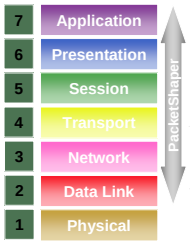


Packeteer white paper example; not Stanford data

Traffic shaping functions

- Classify and analyze traffic
  - Classify by IP address and port number
  - Use application-specific information (layer 7)
- Control traffic
  - Selectively slow certain classes of traffic
- Monitor network performance
  - Collect performance data, used to improve policies
- Network resilience
  - Active traffic management can provide resilience to DoS attacks, at least within the enterprise network

PacketShaper Classification

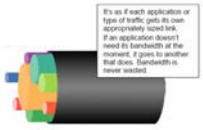


Classify 400+ Apps at OSI Layers 2-7

Peer-to-Peer Apps:
- Aimster
- AudioGalaxy
- CuteMX
- DirectConnect
- Gnutella
- Hotline
- iMesh
- KaZaA/Morpheus
- Napster
- ScourExchange
- Tripnosis….

Some Other Apps:
- H.323
- RTP-I/RTCP-I
- PASV FTP
- HTTP
- Real
- WinMedia
- Shoutcast
- MPEG
- Quicktime
- RTSP
- Chatting Apps
- Games

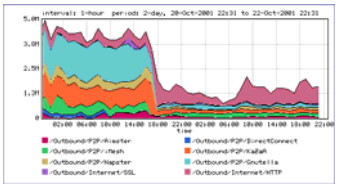
PacketShaper Controls



A partition:
- Creates a virtual pipe within a link for each traffic class
- Provides a min, max bandwidth
- Enables efficient bandwidth use

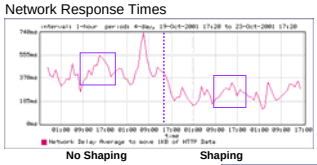
Rate shaped P2P capped at 300kbps

Rate shaped HTTP/SSL to give better performance



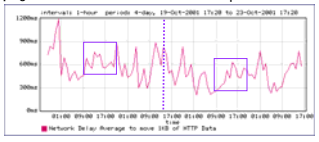
PacketShaper report: HTTP

Outside Web Server Normalized Network Response Times



No Shaping Shaping

Inside Web Server Normalized Network Response Times



No Shaping Shaping

Host and network intrusion detection

- Intrusion prevention
  - Network firewall
    - Restrict flow of packets; cover in another lecture
  - System security
    - Find buffer overflow vulnerabilities and remove them!
- Intrusion detection
  - Discover system modifications
    - Tripwire
  - Look for attack in progress
    - Network traffic patterns
    - System calls, other system events

Tripwire

- Outline of standard attack
  - Gain user access to system
  - Gain root access
  - Replace system binaries to set up backdoor
  - Use backdoor for future activities
- Tripwire detection point: system binaries
  - Compute hash of key system binaries
  - Compare current hash to hash stored earlier
  - Report problem if hash is different
  - Store reference hash codes on read-only medium

Is Tripwire too late?

- Typical attack on server
  - Gain access
  - Install backdoor
    - This can be in memory, not on disk!!
  - Use it
- Tripwire
  - Is a good idea
  - Wont catch attacks that don't change system files
  - Detects a compromise that has happened

Remember: Defense in depth

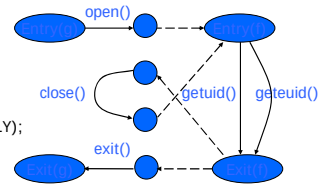
Detect modified binary in memory?

- Can use system-call monitoring techniques
- For example [Wagner, Dean IEEE S&P '01]
  - Build automaton of expected system calls
    - Can be done automatically from source code
  - Monitor system calls from each program
  - Catch violation

Results so far: lots better than not using source code!

Example code and automaton

```
f(int x) {
  x ? getuid() : geteuid();
  x++
}
g() {
  fd = open("foo", O_RDONLY);
  f(0); close(fd); f(1);
  exit(0);
}
```



If code behavior is inconsistent with automaton, something is wrong

General intrusion detection

http://www.snort.org/

- Many intrusion detection systems
  - Close to 100 systems with current web pages
  - Network-based, host-based, or combination
- Two basic models
  - Misuse detection model
    - Maintain data on known attacks
    - Look for activity with corresponding signatures
  - Anomaly detection model
    - Try to figure out what is "normal"
    - Report anomalous behavior
- Fundamental problem: too many false alarms

Misuse example - rootkit

- Rootkit sniffs network for passwords
  - Collection of programs that allow attacker to install and operate a packet sniffer (on Unix machines)
  - Emerged in 1994, has evolved since then
  - 1994 estimate: 100,000 systems compromised
- Rootkit attack
  - Use stolen password or dictionary attack to get user access
  - Get root access using vulnerabilities in rdist, sendmail, /bin/mail, loadmodule, rpc.ypupdated, lpr, or passwd
  - Ftp Rootkit to the host, unpack, compile, and install it
  - Collect more username/password pairs and move on

Rootkit covers its tracks

- Modifies netstat, ps, ls, du, ifconfig, login
  - Modified binaries hide new files used by rootkit
  - Modified login allows attacker to return for passwords
- Rootkit fools simple Tripwire checksum
  - Modified binaries have same checksum
  - But a better hash would be able to detect rootkit

Detecting rootkit on system

- Sad way to find out
  - Disk is full of sniffer logs
- Manual confirmation
  - Reinstall clean ps and see what processes are running
- Automatic detection
  - Rootkit does not alter the data structures normally used by netstat, ps, ls, du, ifconfig
  - Host-based intrusion detection can find rootkit files
    - As long as an update version of Rootkit does not disable your intrusion detection system ...

Detecting network attack (Sept 2003)

- Symantec honeypot running Red Hat Linux 9
- Attack
  - Samba 'call_trans2open' Remote Buffer Overflow (BID 7294)
  - Attacker installed a copy of the SHV4 Rootkit
- Snort NIDS generated alerts, from this signature

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 \
(msg:"NETBIOS SMB trans2open buffer overflow attempt"; \
flow:to_server,established; \
content:"|00|"; offset:0; depth:1; \
content:"|ff|SMB|32|"; offset:4; depth:5;
content:"|00 14|"; offset:60; depth:2; \
...
```

More info: https://tms.symantec.com/members/AnalystReports/030929-Analysis-SHV4Rootkit.pdf

Misuse example - port sweep

- Attacks can be OS specific
  - Bugs in specific implementations
  - Oversights in default configuration
- Attacker sweeps net to find vulnerabilities
  - Port sweep tries many ports on many IP addresses
  - If characteristic behavior detected, mount attack
    - SGI IRIX responds TCPMUX port (TCP port 1)
    - If machine responds, SGI IRIX vulnerabilities can be tested and used to break in
- Port sweep activity can be detected

Anomaly Detection

- Basic idea
  - Monitor network traffic, system calls
  - Compute statistical properties
  - Report errors if statistics outside established range
- Example – IDES (Denning, SRI)
  - For each user, store daily count of certain activities
    - E.g., Fraction of hours spent reading email
  - Maintain list of counts for several days
  - Report anomaly if count is outside weighted norm

Big problem: most unpredictable user is the most important

[Hofmeyr, Somayaji, Forrest]

Anomaly – sys call sequences

- Build traces during normal run of program
  - Example program behavior (sys calls)

    open read write open mmap write fchmod close
  - Sample traces stored in file (4-call sequences)

    open read write open
    read write open mmap
    write open mmap write
    open mmap write fchmod
    mmap write fchmod close
  - Report anomaly if following sequence observed

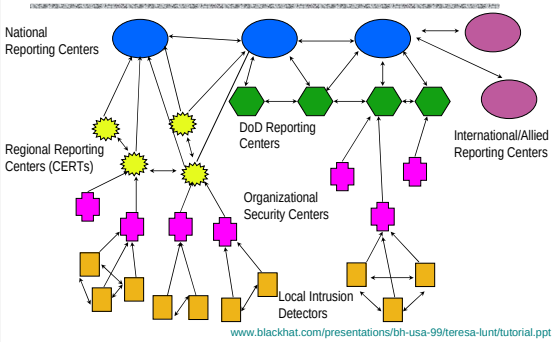
    open read read open mmap write fchmod close

    Compute # of mismatches to get mismatch rate

Difficulties in intrusion detection

- Lack of training data
  - Lots of “normal” network, system call data
  - Little data containing realistic attacks, anomalies
- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally
- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities
- False identifications are very costly
  - Sys Admin spend many hours examining evidence

Strategic Intrusion Assessment [Lunt]



Strategic Intrusion Assessment [Lunt]

- Test over two-week period
  - AFIWC’s intrusion detectors at 100 AFBs alarmed on 2 million sessions
  - Manual review identified 12,000 suspicious events
  - Further manual review => four actual incidents
- Conclusion
  - Most alarms are false positives
  - Most true positives are trivial incidents
  - Of the significant incidents, most are isolated attacks to be dealt with locally

Lecture Topics

- Firewalls
  - Packet filter (stateless, stateful)
  - Application-layer gateway
- Traffic Shaping
- Intrusion detection
  - Anomaly and misuse detection
  - Host and network intrusion detection