Network Security: Prevention, Detection and Mitigation

Dr. Charles Antonelli
Center for Information Technology Integration
School of Information
University of Michigan

June 22, 2006

Content

Some of this material is covered in a three-month security training course I developed for system administrators at U-M

Course contributors:

- Kirk Soluk, U-M IT Security Services
- Matt Bing, U-M IT Security Services
- About a dozen domain expert guest lecturers
- http://www.itss.umich.edu/training/

Work supported by U-M IT Security Services

Agenda

- Security foci
- Prevention
- Detection
- Mitigation

- Linux and Windows environments
- Introduction to building & using tools

Traditional Security Focus

The infrastructure landscape

- Computing hardware
- Operating systems
- Network infrastructure
  - Routers, switches, hubs
  - Protocols, middleboxes
  - VLANs, VPNs
- File systems
- Security infrastructure
  - Identification, Authentication, Authorization
- Middleware
- Applications, libraries

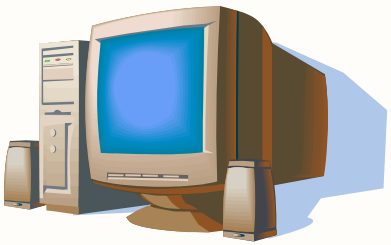
User Security Focus

Navigating around the landscape

- Complex, arcane, layered systems & tools
- Onerous, repetitive authentication procedures
- Hidden network infrastructure
- Malicious software, viruses, worms
- Malicious web sites, services
- Risk of identity, data, asset theft

Secure the network

- Prevention
  - Firewalls
  - Network Scanning
  - Security risk assessment
- Detection
  - Intrusion detection
- Mitigation
  - Attack surface reduction



Secure the user

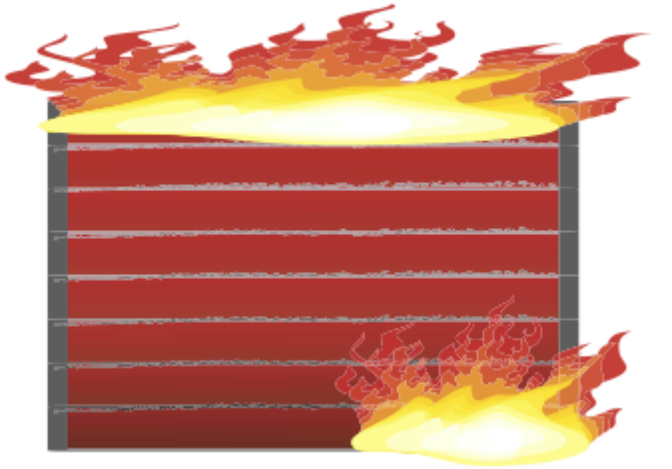
- Prevention
  - Password security
  - Social engineering
  - Secure remote login
  - RunAs User
  - Google Desktop
- Detection
  - Phishing & Pharming
    - Netcraft toolbar
- Analysis
  - Marketscore
- *… not covered further in this talk*

Prevention

Firewalls

Firewalls

- A *firewall* limits the extent to which hosts on different networks can interact with one another
- Not a panacea, but a necessary security component in today's networks



Packet level firewalls

- Firewall inspects incoming network packets
- Blocks packets violating policy rules
- Rules allow blocking based on
  - Source and destination IP address
  - Source and destination port
  - Protocol, flags, type of service, …

Stateless vs. stateful

- Stateless packet level firewalls treat every packet independently
  - Doesn't relate packets to network connections
  - Doesn't keep any history
- Results in coarse-grained control
  - Forces overly liberal or conservative policies
- Solution: firewall keeps state about recent packet flows
  - Decisions based on packet contents plus stored state
  - More fine-grained control
  - Can obviate application-level firewalls
- Problem
  - All that state consumes firewall resources
- Stateful firewalls are *de rigueur*

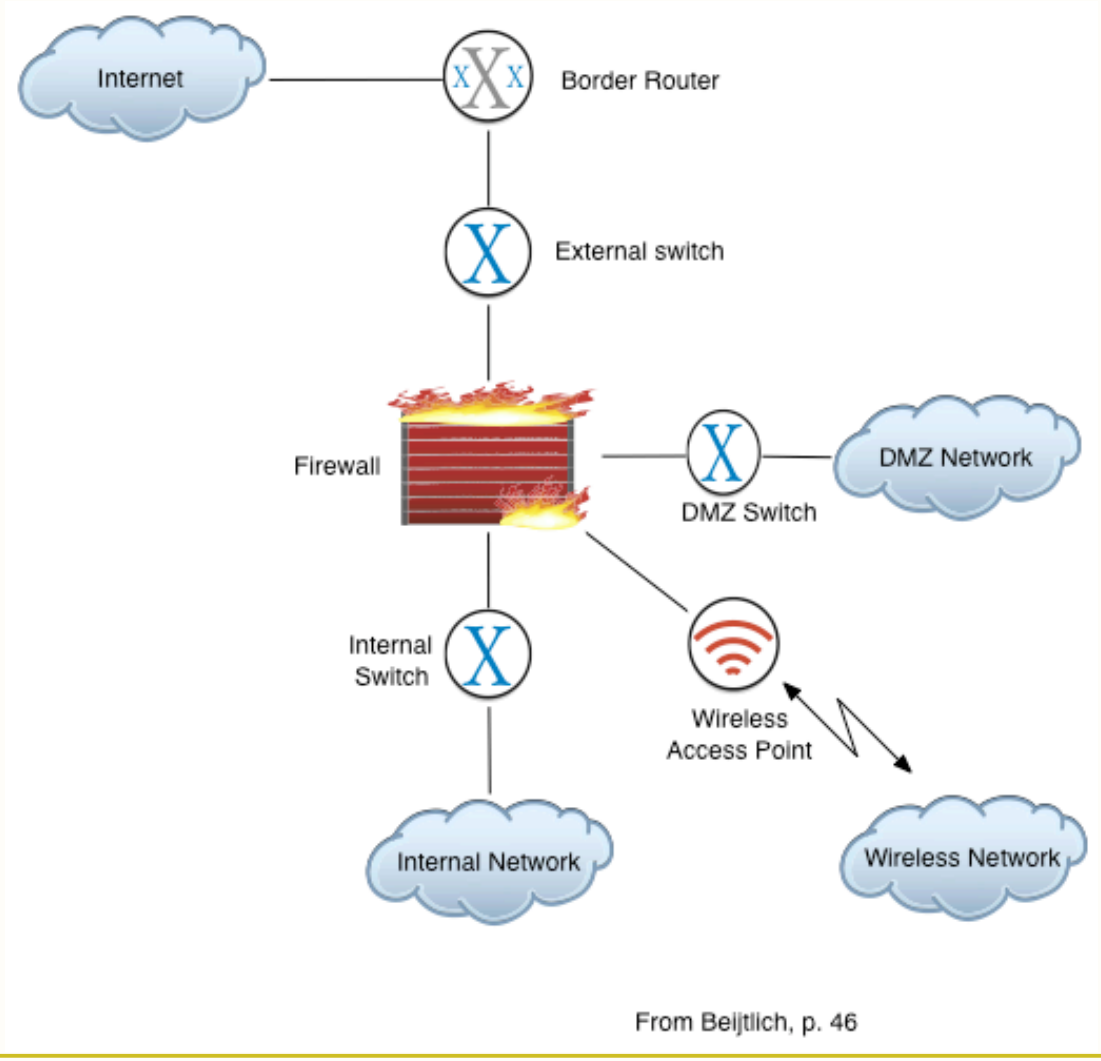
Application-level firewalls

- Application proxy server
  - Accepts client traffic
  - Maintains state, validates traffic
  - Passes validated traffic to server
- Firewall worries about security
  - Obviates security-related server changes
  - Hampers defense-in-depth
- Firewall must understand application protocol
  - Increased complexity
- Stateful packet-level firewalls are an alternative

Host-based firewalls

- Firewall run on individual hosts
- Placed between incoming packets and the host network stack
- Acts like a packet-level firewall
- Each host requires policy management
  - Administration headache
  - Simple default policies in distributions
- Defense-in-depth
- Stateful host-based firewalls are *de rigueur*

Canonical firewalled network



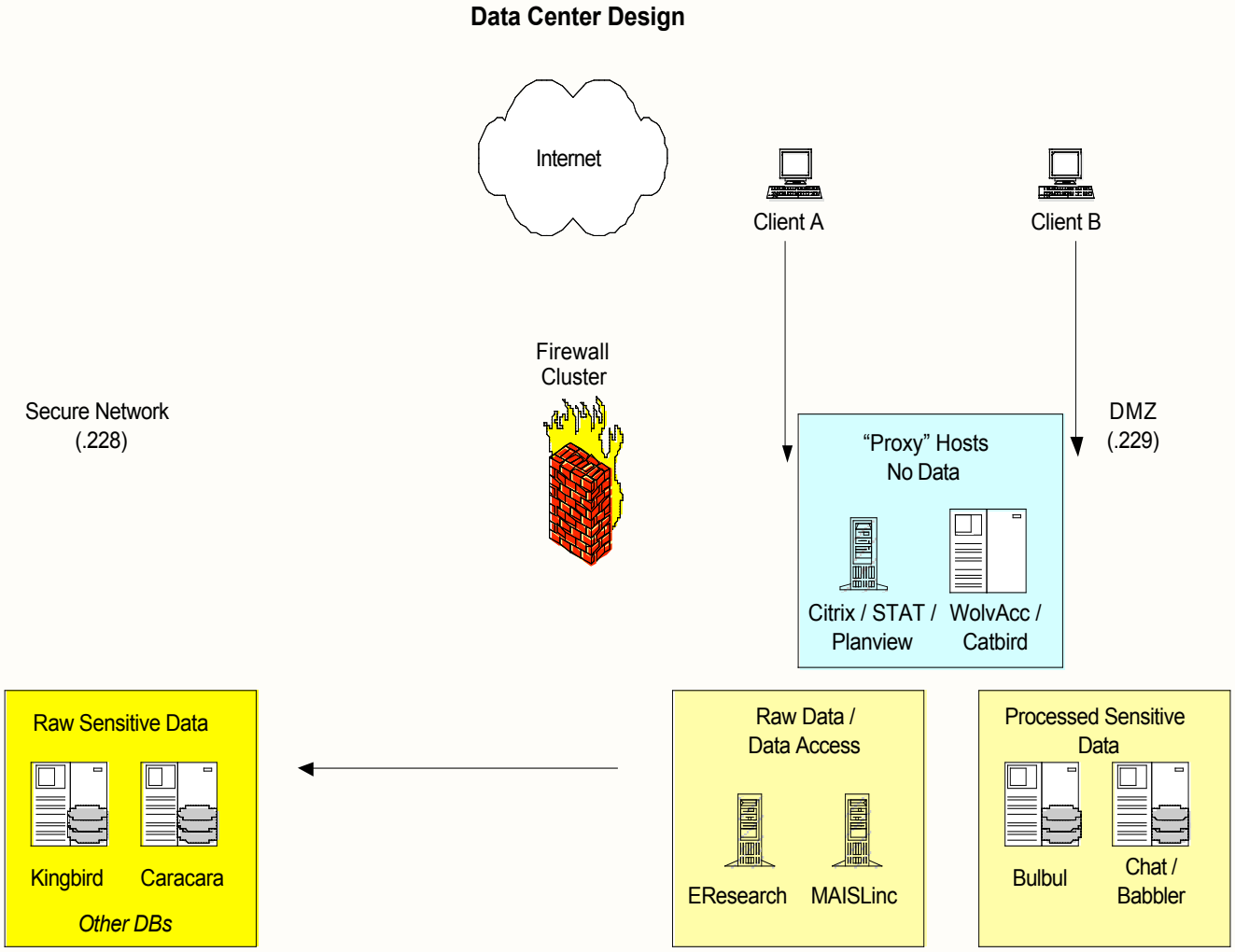
From Beijtlich, p. 46

Canonical Firewall Zones

Collection of networks with specified security properties

- Perimeter: untrusted
- DMZ: semi-trusted
- Intranet: trusted
- Wireless: untrusted!

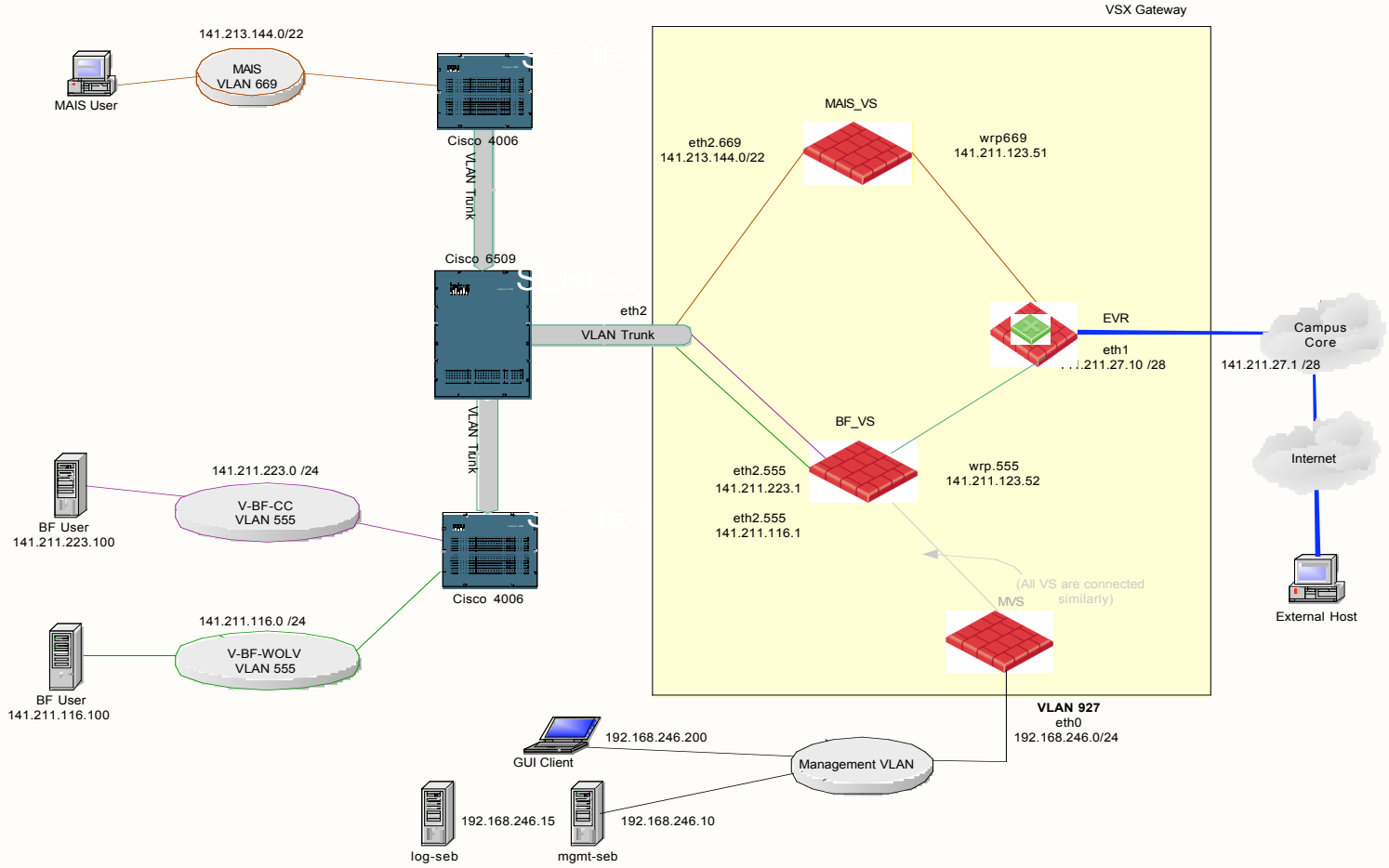
Administrative Computing Data Center Design



Virtual Firewall

- Single firewall can be impractical for a campus
  - Scalability, privacy, compartmentalization, administration
- Solution: virtual firewall
  - Leverages existing VLAN architecture
  - Separate virtual firewall per VLAN
    - Compartmentalizes administration, rule bases
    - Virtual firewalls co-located in physical firewall
  - Requires QoS, VLAN trunking, one subnet per VLAN
- Checkpoint VSX
  - Deployed by Administrative Computing
  - Available to U-M campus units

Virtual Firewall



Linux Firewall

- “IP Tables”
- Packet-level firewall
- Successor to IP Chains
- NAT support
- Extended functionality via modules
- Stateful filter support
- Applications
  - Host based firewall
  - Stateful packet firewall
    - net.ipv4.ip_forward=1 in /etc/sysctl.conf

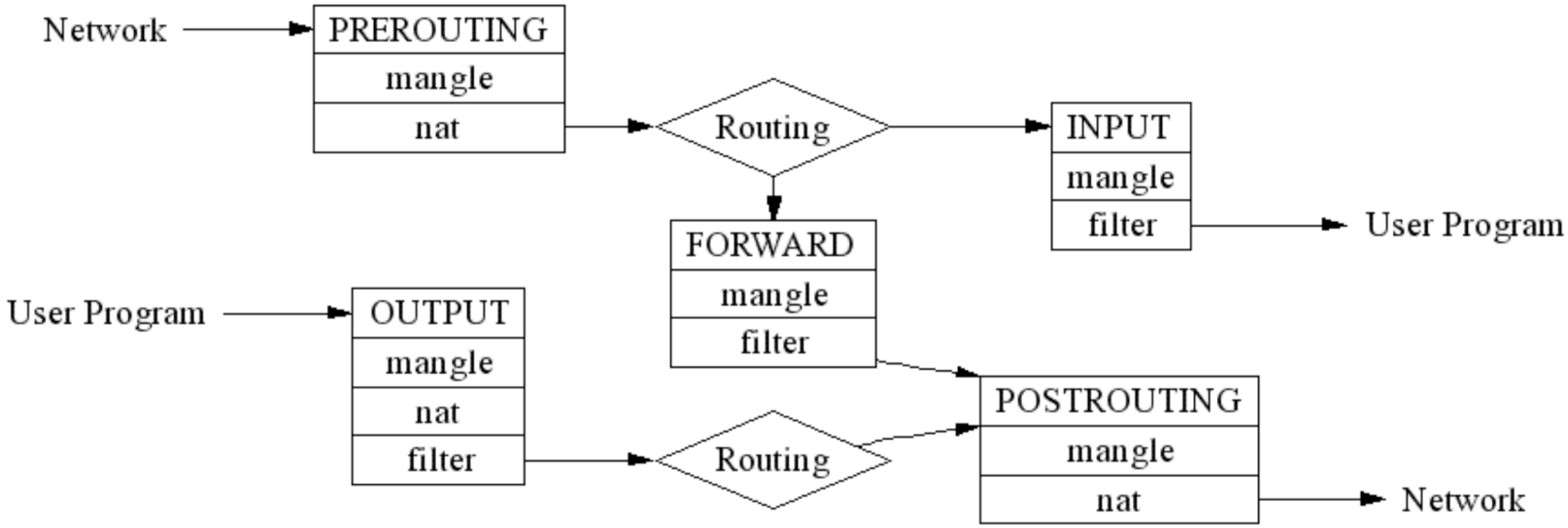
Firewall Rules

- (Standard) *matching criteria*
  - protocol
  - source IP (address/mask)
  - dest IP (address/mask)
  - port (source/destination/both)
  - interface (input/output)
- (Standard) *targets*
  - ACCEPT
  - REJECT
- Plus stateful matching criteria
  - e.g. is packet part of established TCP connection

filter table

- Default table
- Built-in chains
  - INPUT
    - incoming network packets
  - FORWARD
    - packets being routed through the host
  - OUTPUT
    - locally-generated packets output to network
- Other tables: nat and mangle
  - See the man page

Firewall Traversal



Rob Mayoff

IP Tables Example (RHEL4)

```
2SI4#:; iptables -n -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0           0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0           0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:443
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:5443
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```

Interlude: Bastille Linux

- Wizard for locking down Linux
  - Created by a group of security administrators
  - Support for the major Linux distributions
- Categories & step-by-step walkthroughs
  - … including iptables
- Won't apply *any* changes until you've answered *all* the questions
  - Undo feature
- Read-only assessment with risk ratings
- http://www.bastille-linux.org/

Bastille lab

- Install bastille
  - Obtain from Bastille web page or `rpmfind.net`
  - rpm -iv Bastille-3.0.9-1.0.noarch.rpm
  - On RHEL4, you'll also need:
    - rpm -iv perl-Tk-804.027-1.2.el4.rf.i386.rpm
- Run bastille
  - man bastille
  - bastille --assess ("guaranteed" read-only)
  - bastille
- Explore

Windows Firewall

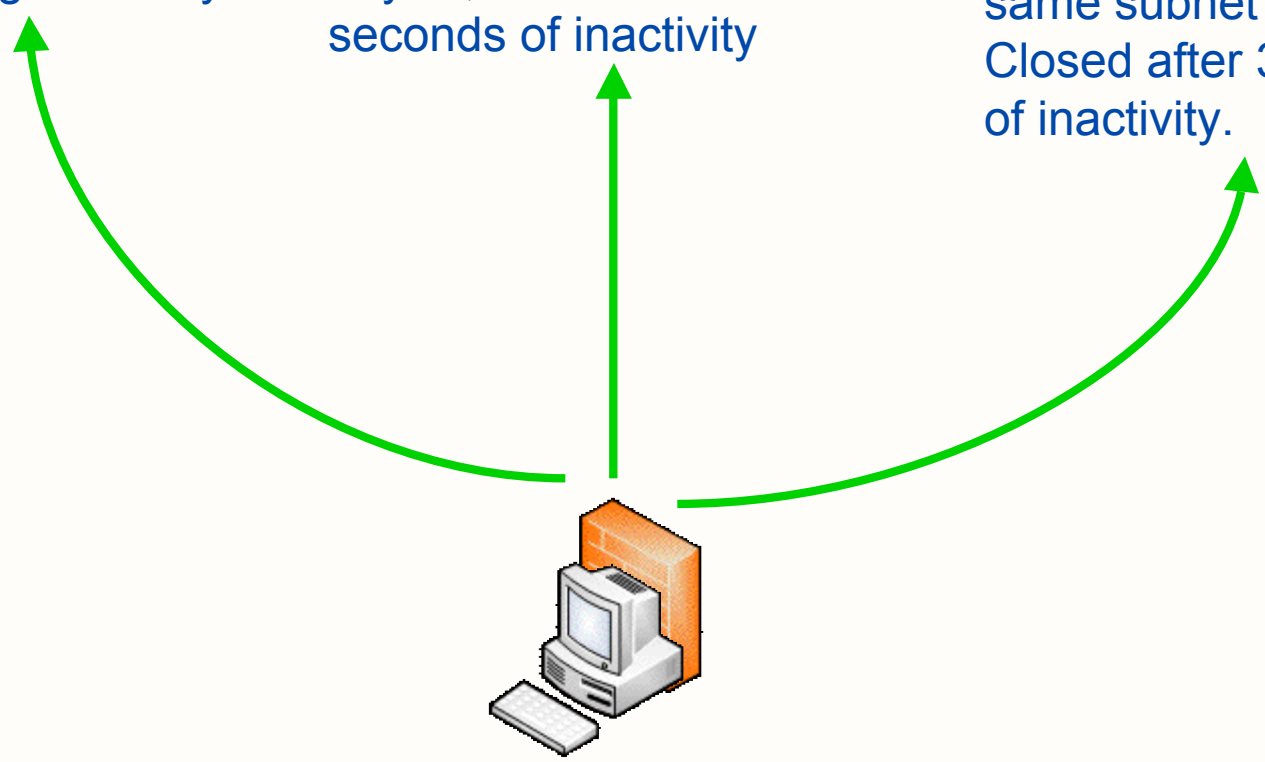
- On by default for all interfaces (XP)
- Stateful
- Supports
  - remote address restrictions
  - port exception
  - program exception
  - ICMP exception
- Can be managed via Group Policy

Windows Firewall Outbound Behavior (Stateful)

Outbound TCP:
Response allowed from target IP only

Outbound UDP:
Response allowed from any IP; closed after 90 seconds of inactivity

Outbound broadcast and multicast:
Response allowed from same subnet only. Closed after 3 seconds of inactivity.



Network Scanning

- Examining host(s) from the network
  - What ports are open
  - What services are running
  - What flaws exist in those services
  - What type of OS is running
  - What kind of filtering is in place
- Attack tool
  - Reconnaissance
- Defensive tool
  - Where are the security risks?

Scanners

- Commercial
  - eEye Retina
  - ISS
  - …
- Open source
  - Nessus
  - Nmap
  - …

nmap

- Network mapping tool
  - Really a network scanner
- Swiss army knife
- Two-step process
  - Identifies hosts on specified network segment(s)
  - Scans specified ports on each host
- Read the man page thoroughly
  - Especially for limitations …
- Generally under-appreciated

nmap

- nmap
  - subnet e.g. 141.211.244.0/26
  - -n don’t map addresses to names
  - -sS TCP SYN port scan
  - -sT TCP connect port scan
  - -sU UDP port scan
  - -sV detect service versions
  - -s… several more advanced scans
  - -O use fingerprinting to guess remote OS
  - -T manually set scan rate
  - -p range range of ports to scan
  - … many more
- Maintained at http://www.insecure.org/nmap/

Nessus

- Was open-source, GPL
  - … Nessus 3.0 closed
- Client/server architecture
  - Server placed on host(s) in network
    - UNIX/Linux, AIX, Mac OS X
  - Client connects to server(s), runs test
    - Windows, UNIX/Linux
- Strong authentication
  - TLS, aka SSL
  - Certificates used to authenticate server

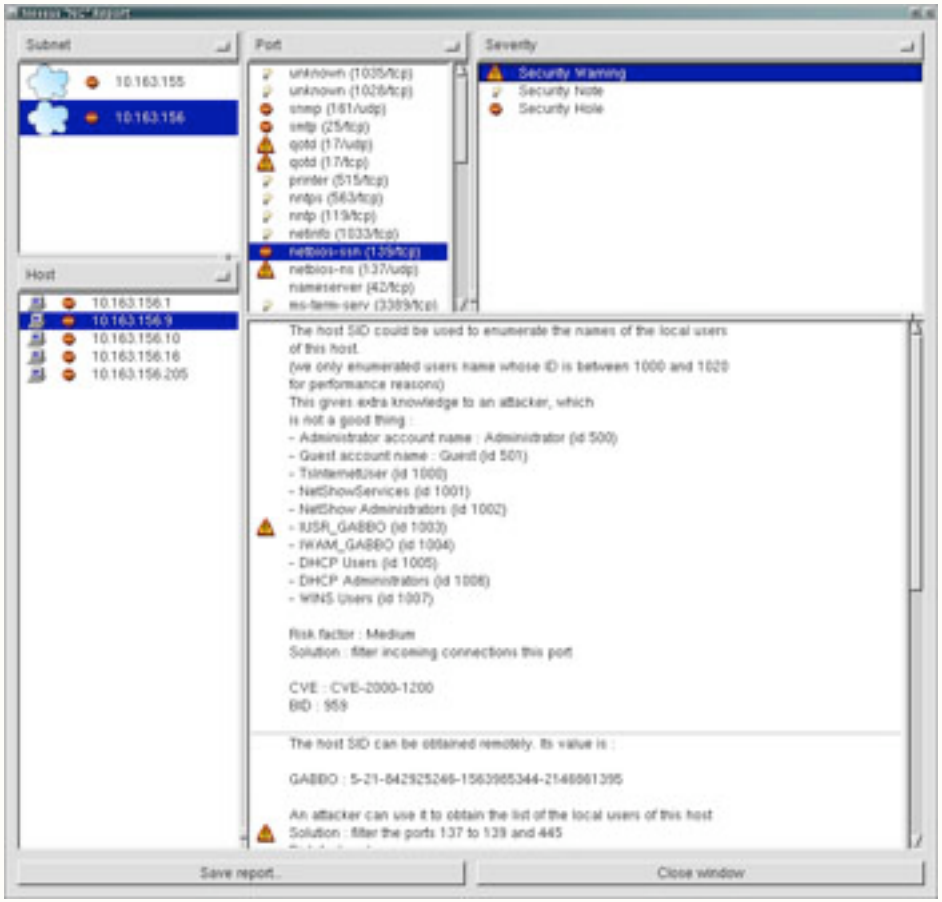
Install Nessus

- http://www.nessus.org/download/
- On RHEL 4, Nessus 3.0 also needs:
  - sharutils-4.2.1-9.i386.rpm
  - freetype-devel-2.1.9-1.i386.rpm
  - fontconfig-devel-2.2.3-7.i386.rpm
  - xorg-x11-devel-6.8.2-1.EL.13.6.i386.rpm
  - glib-devel-1.2.10-11.1.i386.rpm
  - gtk+-devel-1.2.10-33.i386.rpm

Nessus Results

- Subnet -> Host -> Port -> Severity groupings
- Three severity levels
  - Security note - informational
  - Security warning - possible vulnerability
  - Security hole - verified vulnerability
- Detail pane gives description, suggested fixes, references and links
  - Also gives a standardized vulnerability name; see the Common Vulnerability and Exposures list at http://cve.mitre.org/

Nessus Display



Note

As always, you should seek authorization before performing a network scan using any tool

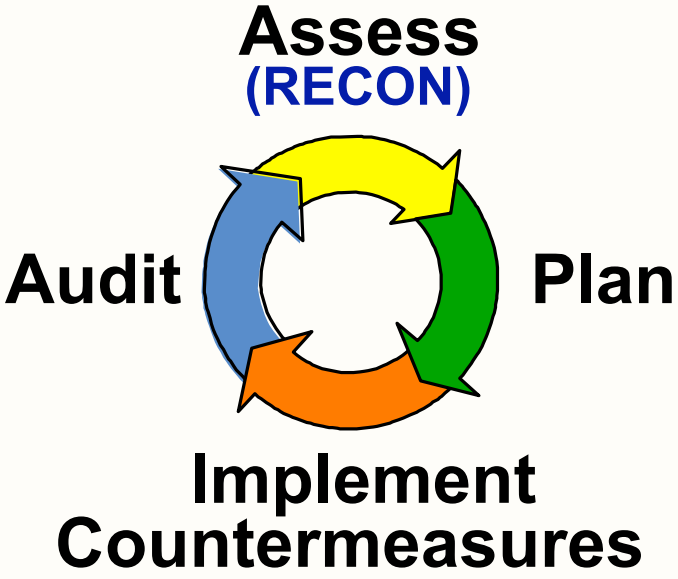
- Scans can trigger intrusion detection systems
- Scans can crash machines
- Scans can print reams of gibberish
- Great way to get on your system administrator's radar

A Note on Penetration Testing

Actively find weaknesses in your systems

- Reconnaissance
  - Google, WHOIS, Web, DNS, Traceroute
  - Newsgroups, discussions, job postings
- Scanning & Enumeration
  - Nmap, Nessus, Retina
  - DumpSec, SQLPing2, Netcat, snmpwalk
- Exploitation
- *Obtaining pen-test authorization is critical!*

Risk Assessment



RECON

Risk Evaluation of Computers and Open Networks

Why Risk Assessment?

- No such thing as perfect security
- Foundation for well-informed decisions that justify IT expenditures
- RECON methodology facilitates consistent decisions across U-M

RECON Background

- Produced and maintained by IT Security Services in collaboration with others
  - University Audits, Administrative Computing, Health System
  - Lessons learned from security course projects
    - 36 units have already conducted RECON-based risk assessments
  - Risk assessment methodology for University-wide GLBA compliance effort
- Standards based
  - ISO 17799 *Code of Practice for information security management (2005)*
  - NIST SP 800-26: *Security Self-Assessment Guide for Information Technology Systems*
- Self-directed
  - Meant to be performed locally by units, schools, and colleges
- Incorporates real-world, hands-on security testing
  - Results based on fact rather than perception.

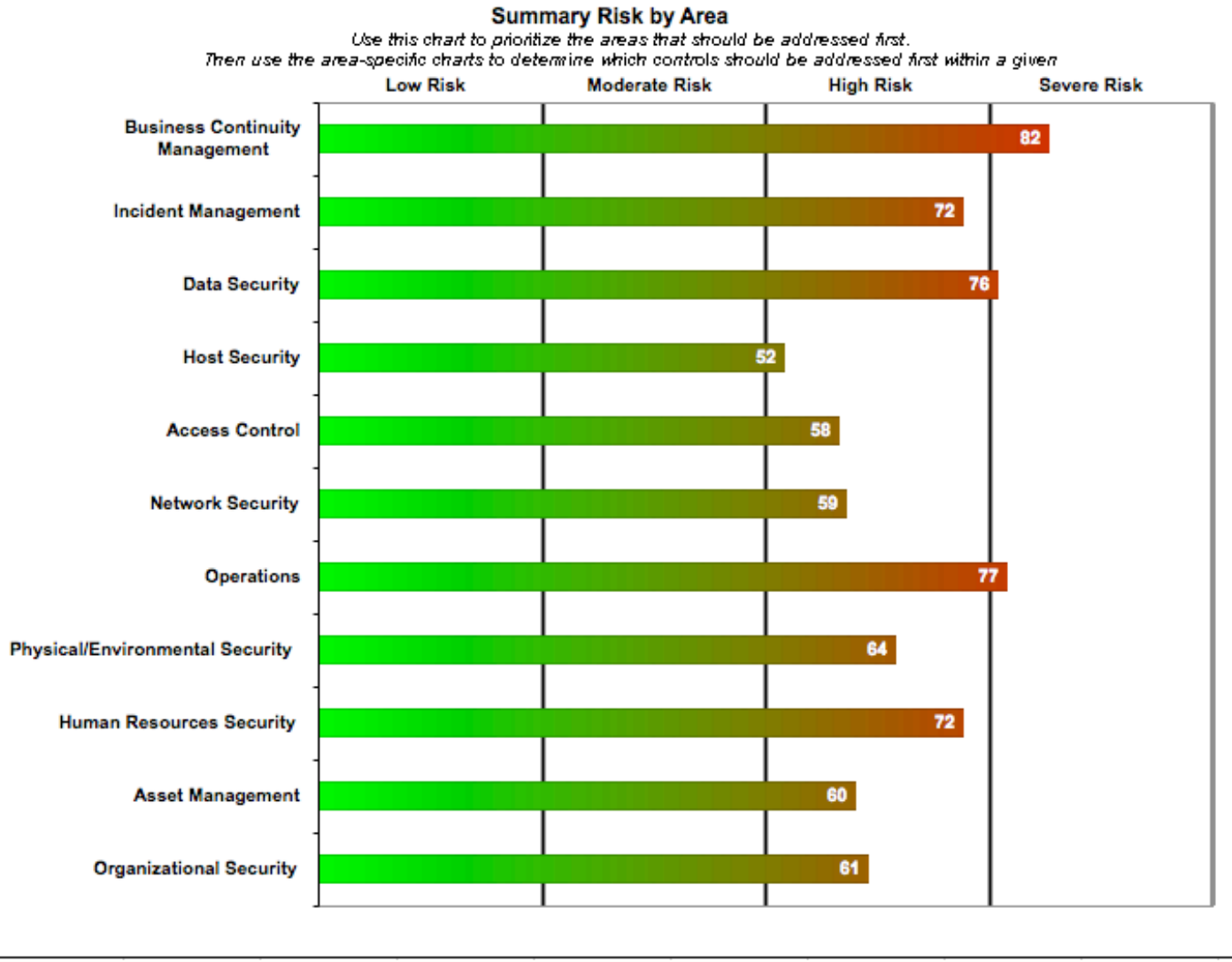
RECON Tangibles

- One Excel Spreadsheet
  - Scope Pages (worksheets)
    - Network Diagram
    - Application Scope
    - System Scope
  - Questionnaire
    - Answers determine level of compliance with ISO 17799 Best Practices
  - Risk Analysis Logic
    - Deviation from best practices represents a risk
  - Built-in Reports
    - Graphically depict prioritized risk areas
- Security Test Document
  - Describes how to perform approximately 15 hands-on security tests
  - Test results used to accurately answer a subset of critical questions

RECON Security Tests

- Attack Surface Enumeration
  - Nmap
  - Service verification
  - Service validation
- Password Audit
  - Default vendor passwords
  - Weak (dictionary) user login and database passwords
- Account Security
  - Unused Accounts
- Firewall Security
  - Nmap again from outside the firewall
- War walking
- RAS Authentication
- Encryption Verification
- Vulnerability Scanning using eEye Retina

RECON Summary Output

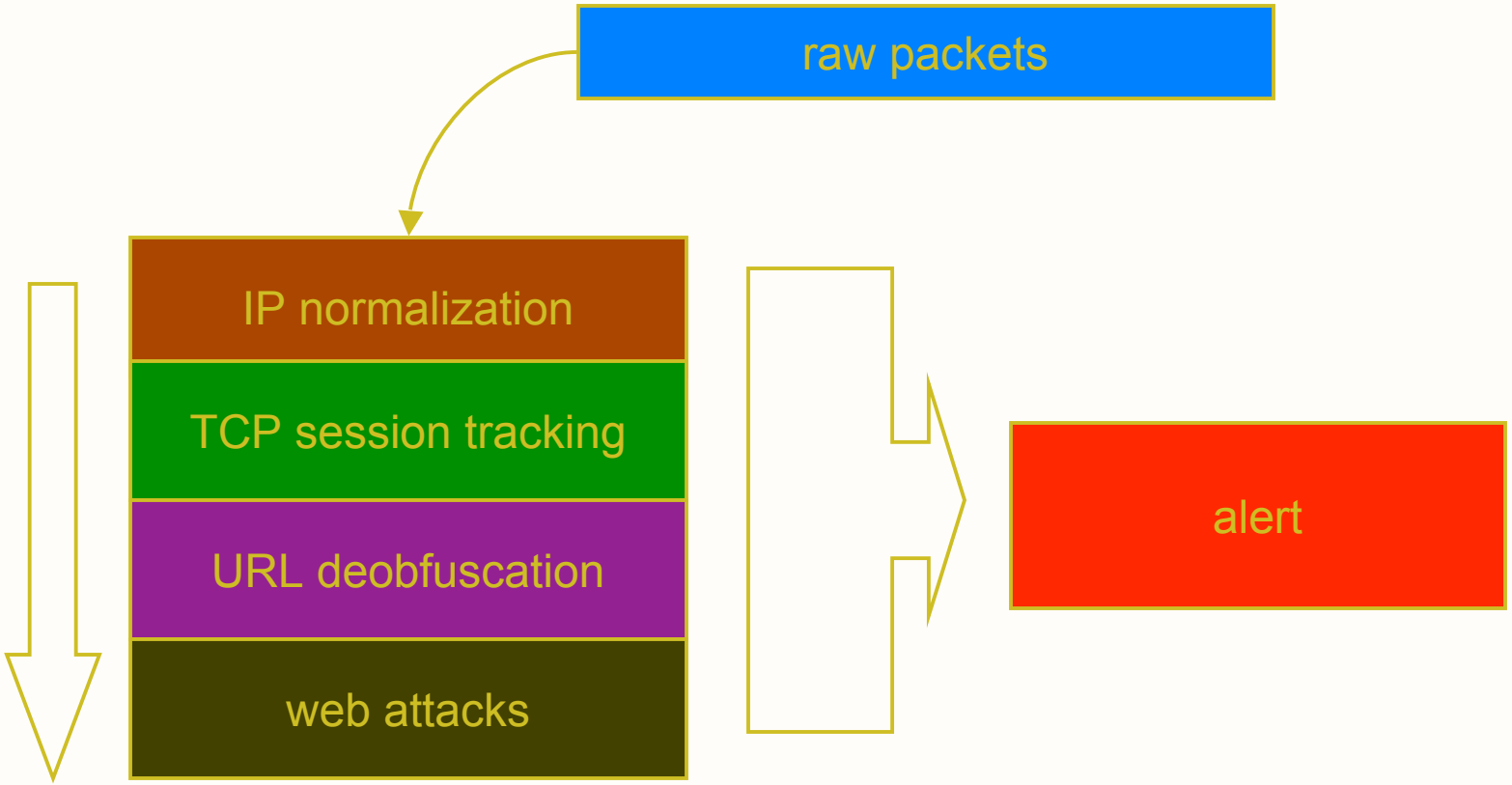


Intrusion Detection

Network Intrusion Detection

- *The goal of the Network Intrusion Detection System (NIDS) is to surmise what the end host will process at each network layer and look for some indication of intrusion*
- A box
  - This is where the magic happens
- Session tracking at each network layer passed up the stack
  - IP defragmentation
  - TCP session reassembly
  - Application layer deobfuscation

Layered detection



Signature vs anomaly

- Signature
  - Does this network traffic match a known, well-formed pattern of a particular attack?
    - HTTP GET /awstats?configdir=|cmd
- Anomaly
  - Does this network traffic differ from the usually observed traffic?
- Writing good rules is an art

NIDS issues

- Packet fragmentation
  - Different OS's reassemble overlapping fragments differently
- Out-of-order packets, low TTL, …
- See Ptacek & Newsham paper
  - … and Dugsong's fragroute for an implementation
- Most network ambiguities are solved
  - Reasonably permissive TCP/IP stack
    - aggressive timeouts to avoid DoS
  - Do not accept data until ACKed by destination
  - Alert on any obvious anomalies
  - UDP remains a problem
    - connection-less

Snort

- Free
- Excellent way to cut your teeth
- Rule based rather than a language
  - One line per rule
  - Syntax supported by most vendors
  - Official rules
  - User contributed rules
    - bleedingsnort.com
    - isc.sans.org
- http://www.snort.org/
- … 0day rules aren't free anymore

Mitigation

Attack surface reduction

Some recommendations

- Scan for existing services
  - Nessus, eEye Retina, nmap
- Run only needed services
  - … and keep them updated!
    - If all you run is sshd, that's where the attacks will come
- RunAs User

Countermeasures

- Manual
  - Block with firewall/router filter rules
- Automated
  - TCP RSTs / UDP port unreachable
    - Race condition with sender
  - Inline blocking
    - Danger, Will Robinson
- These countermeasures are temporary!
  - Buy time to investigate & remediate

IPS

- Intrusion Prevention Systems
  - Inline NIDS
    - “Bump on the wire”
  - Alerts cause traffic to be blocked
    - Drop this packet only
    - Drop packets from this host for some time
  - Has a direct effect on availability

IPS

- You should carefully consider the implications of IPS
  - Attacker spoofs malicious UDP packets from *.root-servers.net
    - Game over

Conclusions

- Practical tools and procedures exist for securing networks
  - For most major platforms and distributions
  - Some good tools are freely available
- Experience is needed to use the tools and interpret the results
  - Don’t let that scare you off
- Securing the infrastructure is a problem different from securing the user

References

- Richard Bejtlich, “The Tao of Network Security Monitoring,” Addison-Wesley, 2004.
- Bob Toxen, “Real World Linux Security: intrusion detection, prevention, and recovery,” 2nd Ed., Prentice-Hall, 2003
- Fyodor’s Top 100 Network Security Tools http://sectools.org
- nmap http://www.insecure.org/nmap/
- Nessus http://www.nessus.org/
- Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection http://www.snort.org/docs/idspaper/
- fragroute http://www.monkey.org/~dugsong/fragroute/
- Snort http://www.snort.org/
- NFR http://www.nfr.net/
- ISS http://www.iss.net/
- RunAs User http://itss.umich.edu/events/download/RunAsUser_sumit_05.pdf
- Google desktop http://safecomputing.umich.edu/tools/download/gd_security.pdf
- Netcraft toolbar http://toolbar.netcraft.com