University of California
San Francisco

Tutorial S09: Computing Protection in Open HPC Environments

SC06

Tampa, FL

November 12, 2006

Who We Are

Stephen Q. Lau
University of California, San Francisco
Office of Academic and Administrative Information Systems
Stephen.Lau@ucsf.edu

William T.C. Kramer
Lawrence Berkeley National Laboratory
National Energy Research Scientific Computing Center
wtkramer@lbl.gov

Scott Campbell
Lawrence Berkeley National Laboratory
National Energy Research Scientific Computing Center
scampbell@lbl.gov

Brian L. Tierney
Lawrence Berkeley National Laboratory
Distributed Systems Department
bltierney@lbl.gov

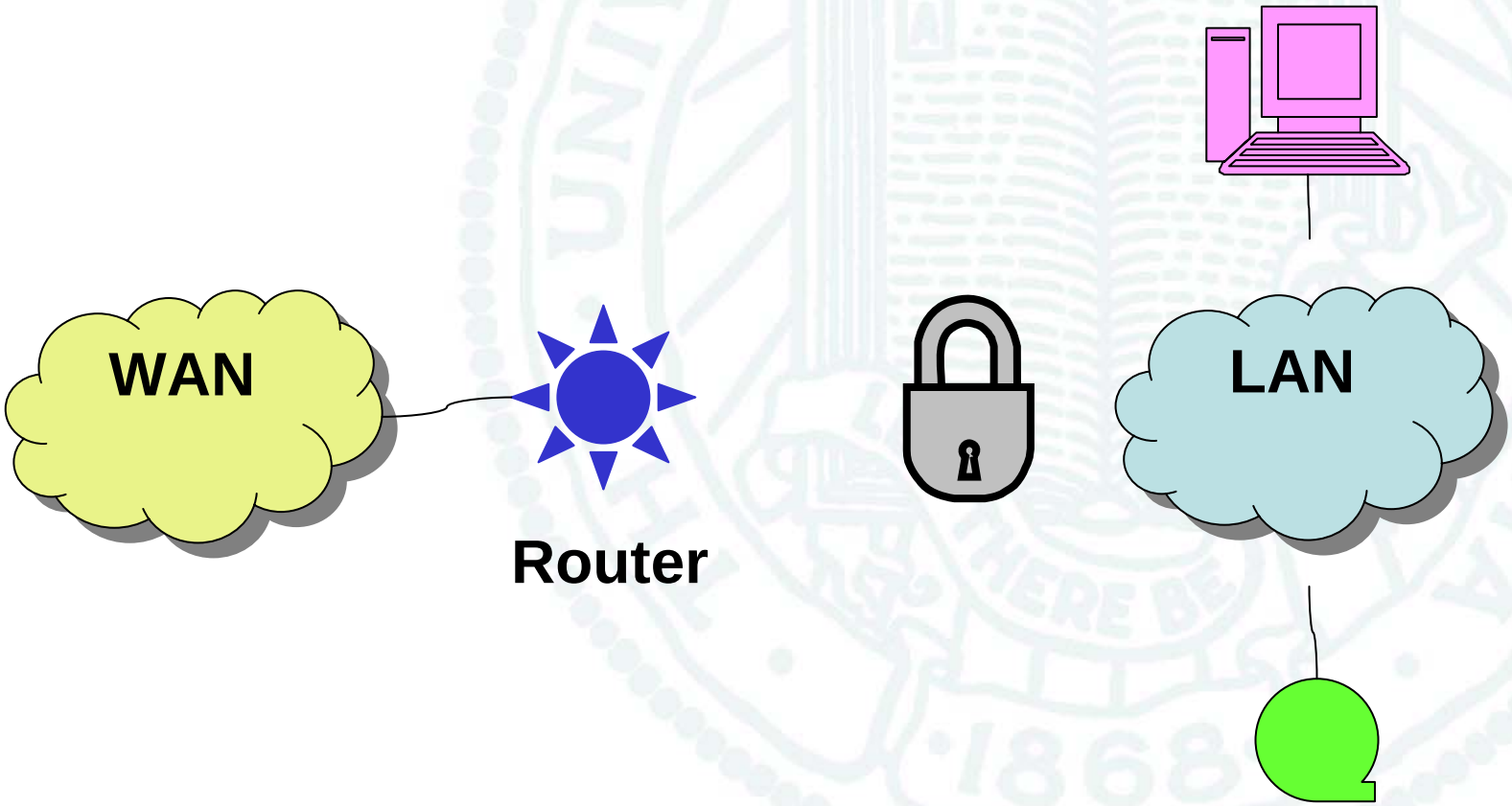
Agenda

• Introduction and Agenda	.25 hour
• HPC Fundamentals	.75 hour
• HPC Security Fundamentals	.5 hour
• Break I	
• HPC Security Protection Fundamentals	.5 hour
• Intrusion Detection Using the Bro System	1.0 hour
• Lunch	
• The Changing Computer Protection Environment	.5 hour
• Keeping One Step Ahead	.75 hour
• Break II	
• Risk Assessment, Mitigation and Compliance	.75 hour
• Components of good protection	.5 hour
• Questions and Answers	.25 hour

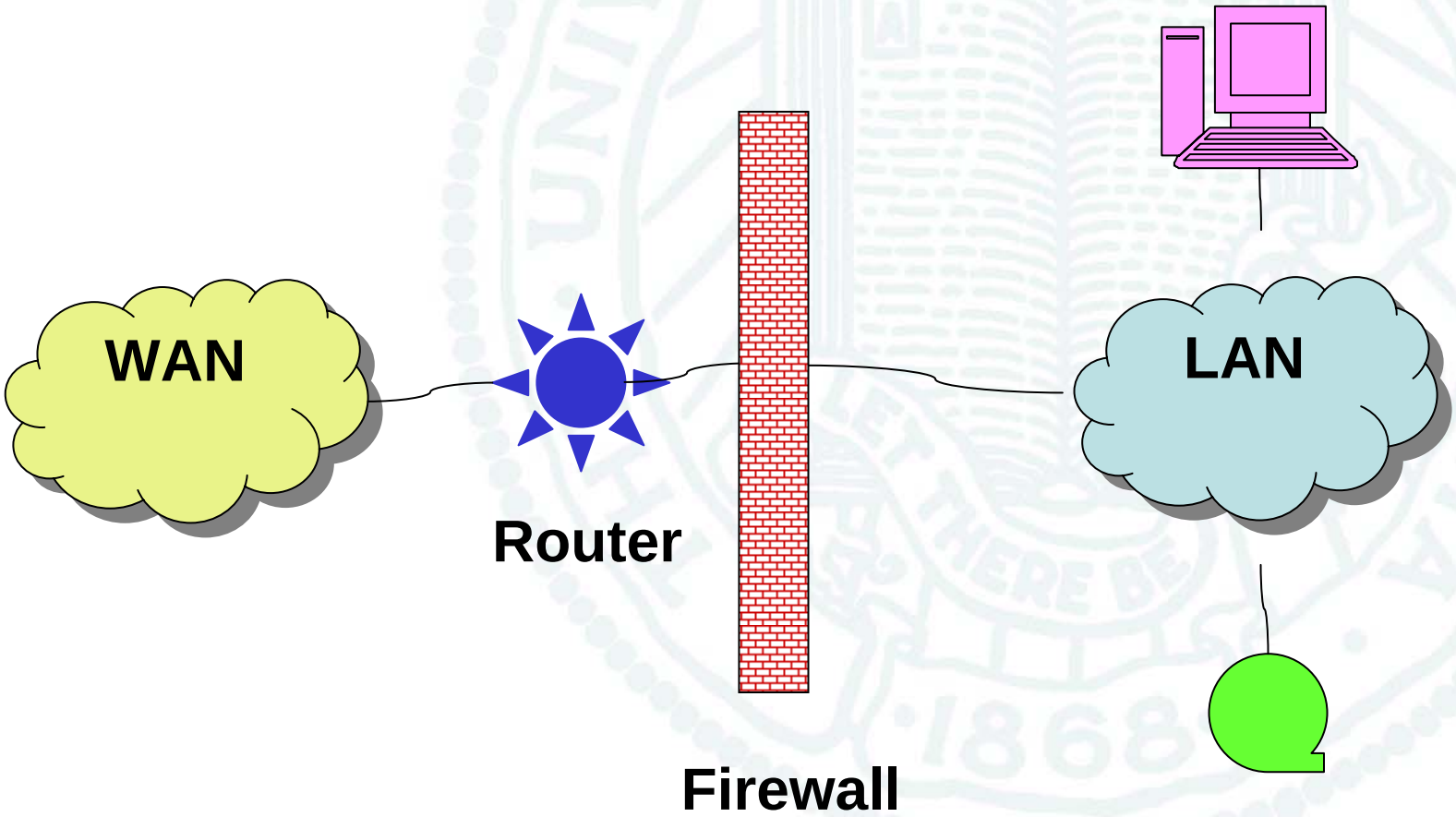
Open Environment

- Unlike enterprise institutions
  - Enterprise oriented computer security techniques typically fail
- Varied and atypical computational infrastructure
  - “We need to support everything under the sun”
- High bandwidth / performance applications
  - Unique applications with unique requirements and traffic patterns
- Varied and distributed resources
- Infrastructure support distributed
  - Support capability also variable
- Multi-institutional collaborations across all levels
  - e.g. LBNL has approximately 4000 collaborations/year

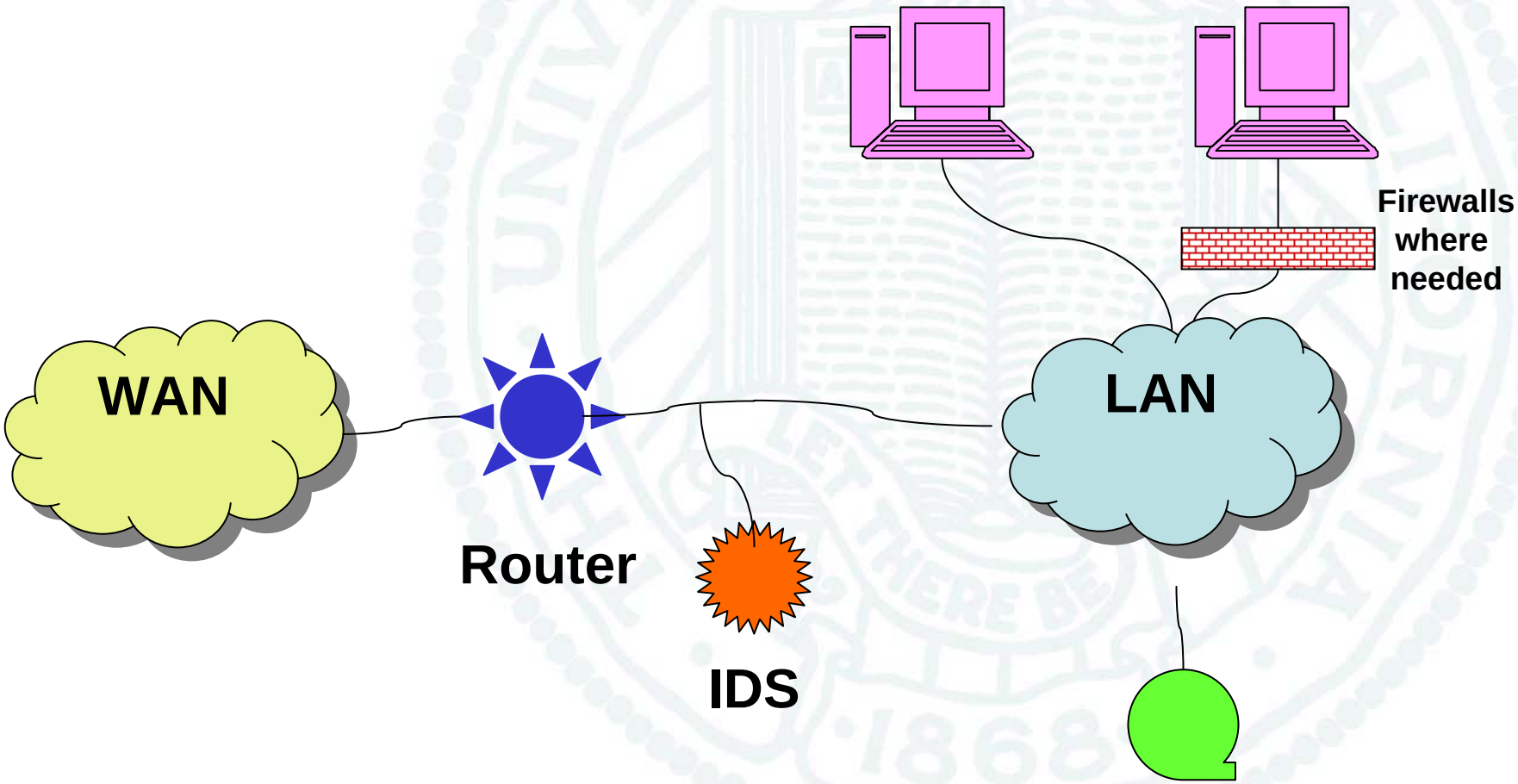
Security Model
Restrictive (Classified/Isolated)



Security Model
Industrial / Enterprise



Security Model
Open Environment



University of California
San Francisco

Office of Academic and
Administrative Information Systems

HPC Fundamentals

Stephen Lau
University of California, San Francisco
Stephen.Lau@ucsf.edu

November 12, 2006

Agenda

- Why Fundamentals?
- Networking basics
  - IP / IPSec / TCP / UDP / ICMP
  - DNS
- System basics
  - Kernels
  - Shared file systems
- Grid basics

Why Fundamentals?

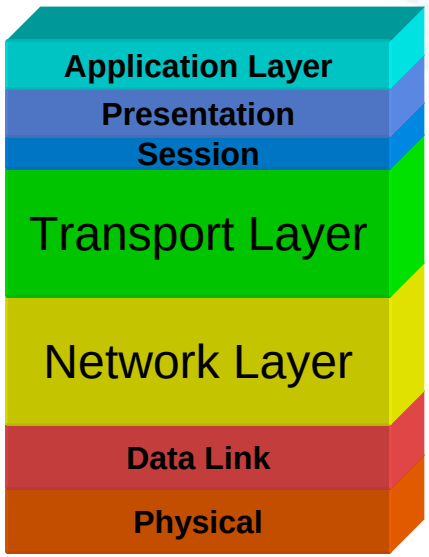
- An understanding of fundamentals is essential to *understanding* security.
  - Efficient differentiation between false positives, “network crud” and security incidents.
  - Ability to effectively deploy security tools.
  - Able to understand and counter new forms of attack.
- De-mystify information security and attacks.
  - You don’t need to wear black or look like you were an extra from the Matrix movies.
- Encourage you to explore further on your own time.
  - This is a Warp Factor 8.95 tour of fundamentals.

Internet Protocol Stack

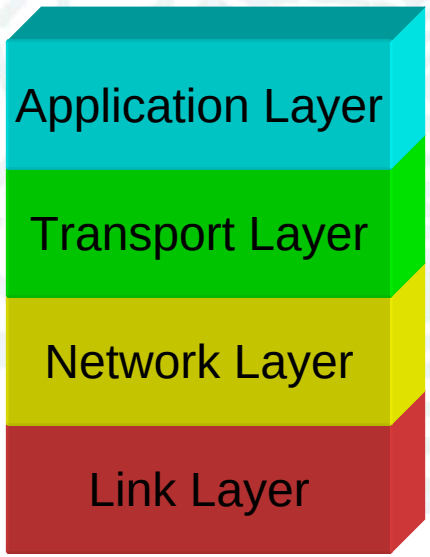
- Data Encapsulation
  - Layered approach to network communication
  - Allows for interchangeability between technologies
  - Easier to develop and maintain

- Predates OSI model of 7 layers
  - Physical, data link, network, transport, session, presentation, application

OSI Model



Internet Protocol Stack



ssh, telnet, ftp, http, etc.

TCP, UDP, etc.

IPv4, IPv6

Ethernet, PPP, ATM etc.

Internet Protocol (IP)

- Underlying Protocol of the Internet
  - IPv4 and IPv6
- Packet based
  - Packets consist of headers and data
- 20 byte header (some of the fields)
  - IP Version
  - Protocol (tcp,udp,icmp)
  - TTL (time to live)
  - Fragment offset
  - Source and Dest addresses

IPSec

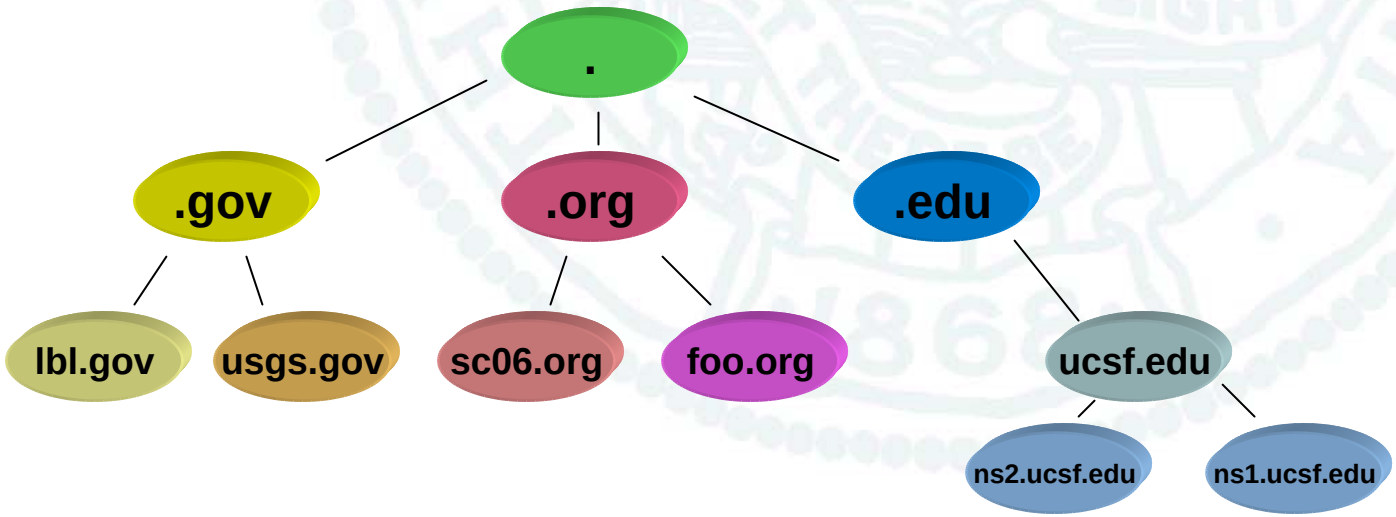
- Secure IP communications
  - Encryption
  - Authentication
- Part of IPv6, optional for IPv4
- Two modes:
  - Transport: Only payload is encrypted
  - Tunnel: entire IP packet encrypted
- Securing Packet flows
  - Authentication Header (AH)
    - Does not offer data confidentiality
  - Encapsulating Security Payload (ESP)
    - Allows for data confidentiality
- Key Management
  - IKE protocol

IP Addresses

- Hosts have IP addresses
  - 32 bit for IPv4 (4,294,967,296 addresses)
    - Denoted as aaa.bbb.ccc.ddd (i.e. 192.168.1.233)
  - 128 bit for IPv6 (340,282,366,920,938,463,463,374,607,431,768,211,456) addresses
    - Denoted in hexadecimal with 0 compression: x:x:x:x:x:x:x:x
      - 2ffe:ffff:101::240:6dff:fe05:d8fe
- CIDR Notation (Classless Inter-Domain Routing)
  - Implemented for more efficient IP address allocation
  - Move away from Class A/B/C allocations
  - 128.218.0.0/16 = 65,536 hosts
  - /12 = 1,048,576 hosts
  - /8 = 16,777,216 hosts
- Private IPv4 addresses space
  - Non-routable (at least they shouldn't be)
  - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16

Domain Name System (DNS)

- Global distributed database that maps IP addresses to names and vice versa.
  - 205.234.170.164 == www.sc06.org
- Hierarchy of DNS Servers
  - 13 current root name servers (www.root-servers.org for more info)
    - [a-m].root-servers.net
- Recursive lookup of hostname through servers.
  - In reality, many entries are cached by local servers.
- Own DNS and you can redirect people wherever you want or cause “denial of service”



Sockets and Ports

- Sockets
  - End point for network communication
  - IP address and Port Number

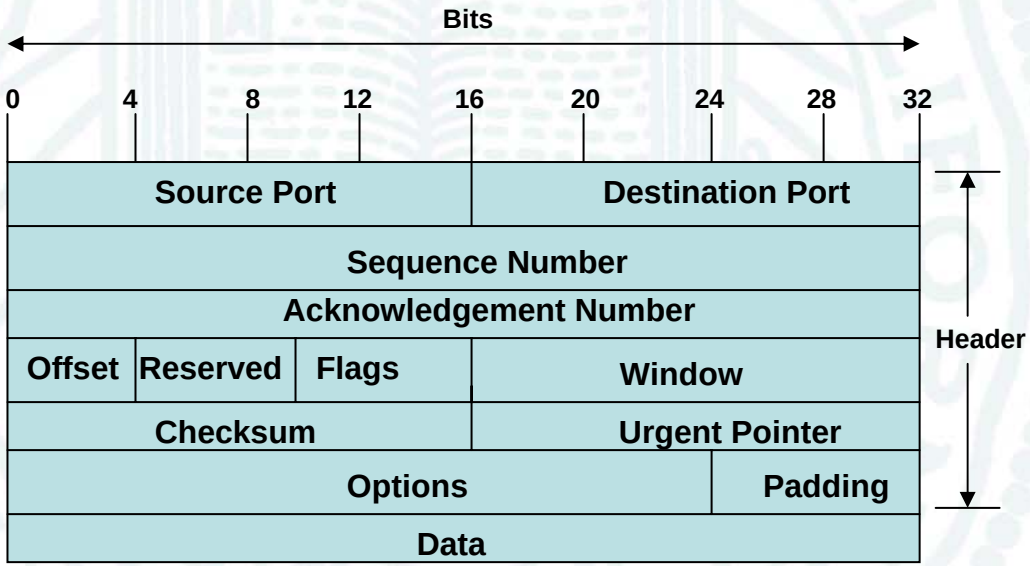
- Ports
  - Used to differentiate between different services/connections
  - They are not COM1, LPT1, etc.
  - TCP/UDP use port numbers
    - Note: TCP port 22 != UDP port 22

- Port numbers (0-65536)
  - See http://www.iana.org/assignments/port-numbers for more information
  - 0-1024 – Common Ports (Well known services)
    - 80 – http
    - 22 – ssh
    - 25 – smtp
  - 1024-49151 Registered Ports
    - 6667 – Internet Relay Chat
    - 5190 – AOL Instant Messenger
  - 49152 – 65535 – Dynamic ports (non-registerable)

Transmission Control Protocol (TCP)

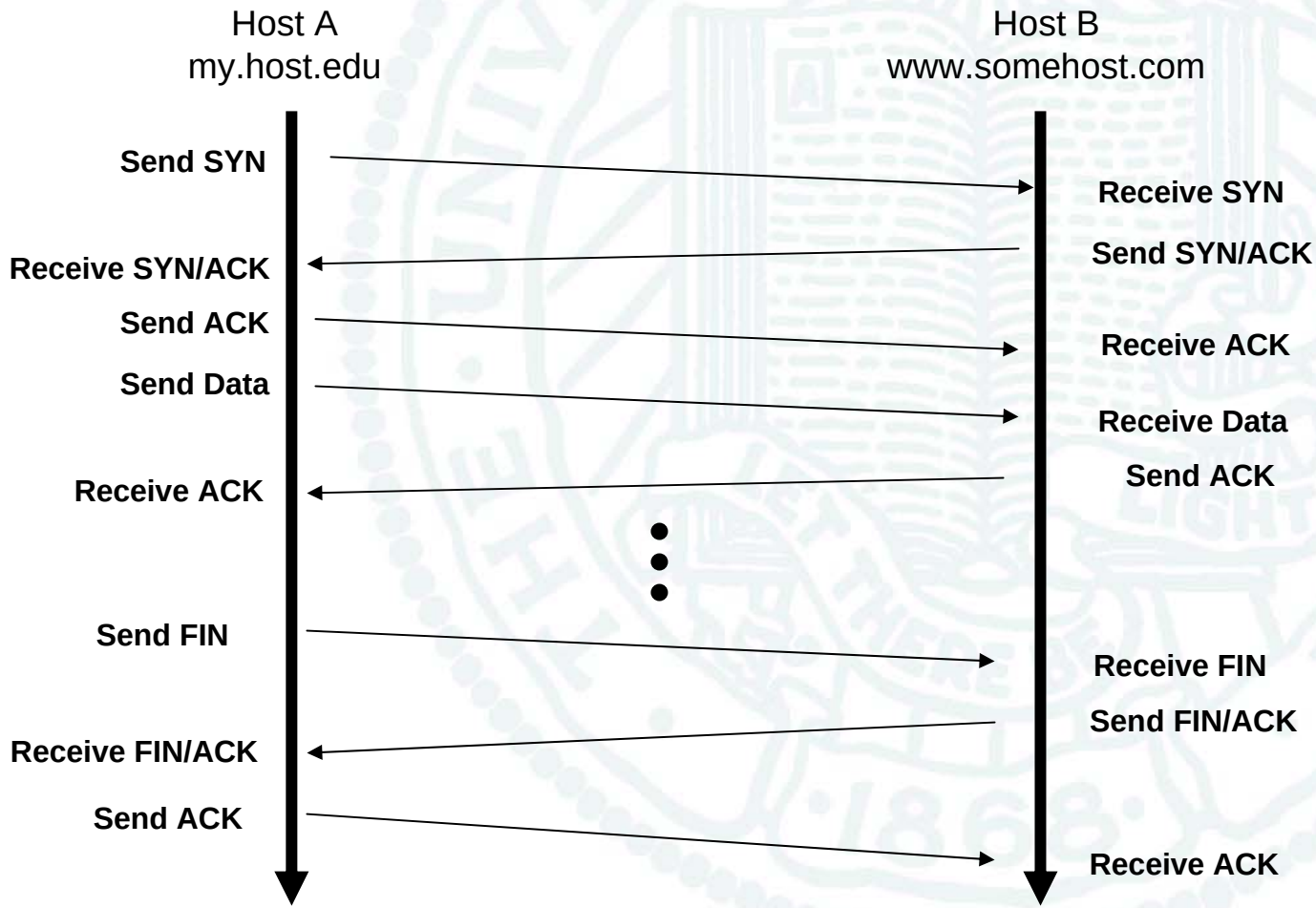
- Reliable connection based packet protocol

- Header
  - 11 fields, 10 required
    - Options field optional
  - Flags
    - URG - Urgent
    - ACK - Acknowledge
    - PSH - Push
    - RST - Reset
    - SYN - Synchronize
    - FIN - Finished



- Three phases to a connection
  - Connection Establishment
  - Data Transfer
  - Connection Termination

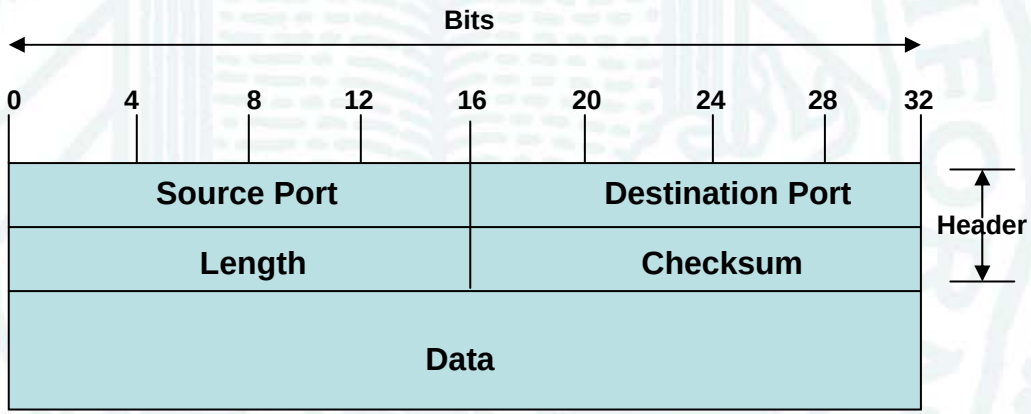
TCP Connection



User Datagram Protocol (UDP)

- Stateless packet based protocol
  - Provides no guarantee for message delivery.

- Header – 4 Fields
  - Source Port
  - Destination Port
  - Length
  - Checksum



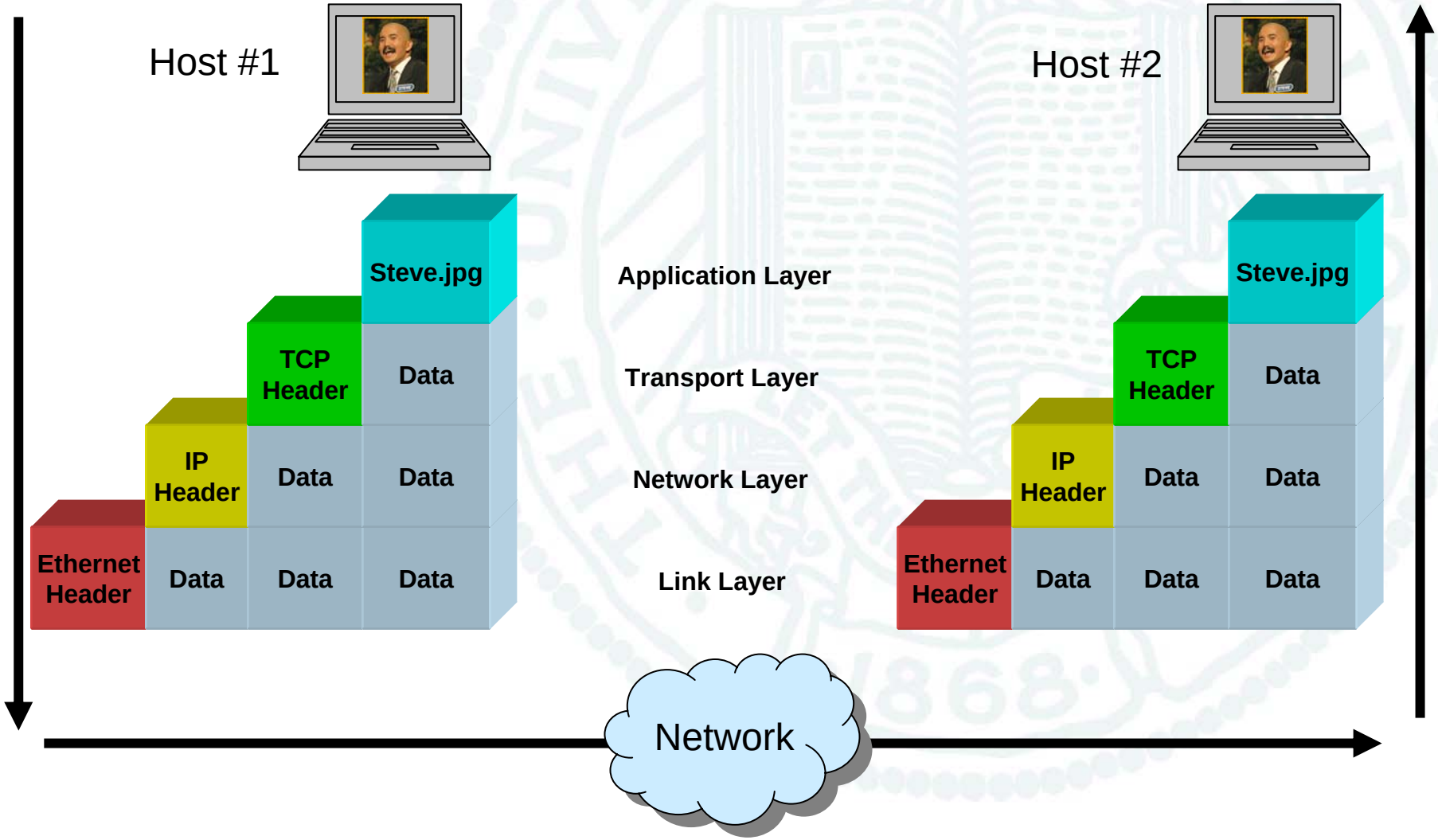
- Common UDP based applications
  - DNS
  - DHCP
  - Streaming media applications

Internet Control Message Protocol (ICMP)

- Designed for out of band network related messages
  - Network Errors
  - Network Congestion
  - Network Troubleshooting
  - Timeouts
- Messages contained within one IP datagram
- No guarantee for delivery
- Does not typically transport data
- Common applications using ICMP
  - traceroute, ping

How it Works

(High Level Conceptualization)



Network Crud

- Not all network devices play nice on the Internet.
  - Lots of broken implementations of networking protocols.

- Causes problems with many protection systems.
  - Most systems were designed in clean environments, not the real Internet.
  - Most consumers don't understand the network.

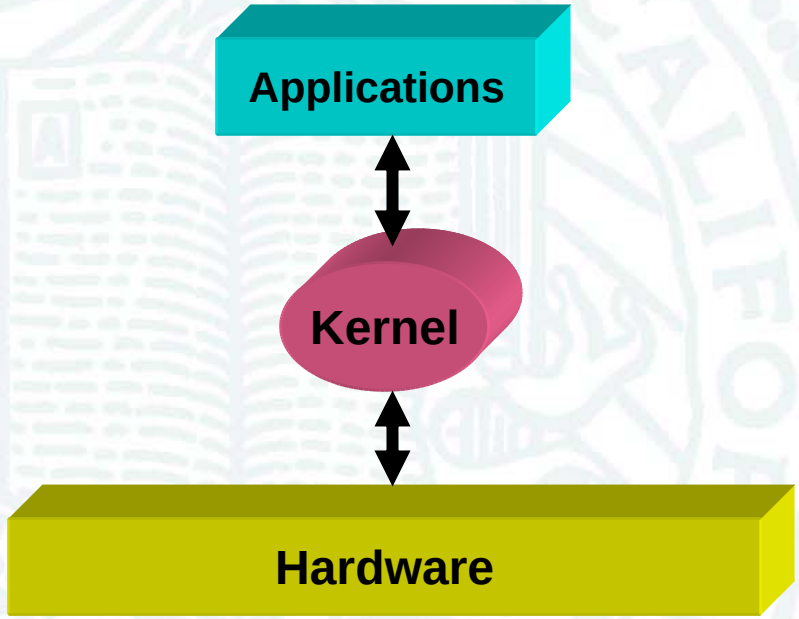
- Examples:
  - Storms of 10,000+ FIN or RST packets, due to TCP bugs.
  - Storms due to foggy days.
  - Private addresses leaking out.

  - Legitimate tiny fragments.
  - Fragments with DF set.
  - Overlapping fragments.

  - TCPs that acknowledge data that was never sent (!).
  - TCPs that retransmit different data than sent the first time (!).

Kernels

- Core part of an Operating System
  - *nix
  - Windows
  - OSX



- Connects hardware to software
  - Process management
  - Memory management
  - Device management

- Kernel Modules allow for modifiable kernels
  - i.e. device drivers
  - Doesn't necessarily require reboots or rebuilds to change kernel

- Obviously, this is a key attack point!
  - Own the kernel, and you can no longer trust what it is reporting.

Distributed File Systems

- Developed to allow people to collaborate
- Unfortunately most weren't developed with security in mind!
  - Most do not provide confidentiality, authentication, auditing, authorization.
- *nix
  - NFS – Network File System
    - Primarily LAN workstations
    - Unencrypted data transmissions
    - No authentication
  - AFS – Andrew File System
    - Utilizes Kerberos for authentication
  - Lustre
    - Large scale cluster computing
  - GPFS – General Parallel File System
- Windows
  - SMB (Server Message Block)
    - Originally designed for NetBIOS / now TCP/UDP enabled.
    - Opensource versions to allow for interoperability with *nix environments.

Grid

- Infrastructure to share resources in a distributed nature
  - Computing power, disk, network
  - Useful in solve large scale problems
- Authentication and Authorization mechanisms
  - PKI (Public Key Infrastructure)
    - Certificate based system
  - myProxy, Kerberos and One Time Passwords
- Many working groups on securing Grid technologies
  - Grid software reliability and security is increasing.
- Most Grids are administered in a distributed fashion
  - Multi-organization / international collaborations
  - Relying on other sites to do proper implementation and system administration
  - Incidents quickly move cross-site
  - Incident resolution requires multi-site coordinated response.

Summary

- Warp 8.95 tour of key fundamentals
  - Encourage you to explore more!
- Not everything covered
  - Routing, permissions, authentication, authorization
  - The list is endless…
- A fundamental understanding of ***what*** you are protecting is essential to protect it!

University of California
San Francisco

Office of Academic and
Administrative Information Systems

HPC Security Fundamentals

Stephen Lau
University of California, San Francisco
Stephen.Lau@ucsf.edu

November 12, 2006

Agenda

- Security Fundamentals Overview
- Buffer Overflows
- Worms
- Viruses
- Scans
- Kernel Rootkits
- Password Cracking
- Background Radiation

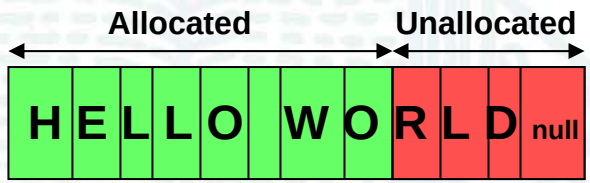
- Covers common security exploits
  - Some have been around for many years
- Threats are constantly changing and becoming more complex
  - See this afternoon's session for more details.
- Strongly advise keeping abreast of computer security.
  - *Not just what vendors or CNN tells you about!*

Heaps, Stacks, etc.

- Memory
  - Addressable
  - When a program is run, it is loaded into memory and executed from memory
- Stack
  - Last in / First Out memory
  - Used to retain state
- Heap
  - Dynamically allocated memory
  - Kernel ensures no collisions between programs.
- Functions
  - Portions of program code that gets executed.
  - When a function is invoked.
    - Current execution address is retained onto stack (usually)
    - Local variables are allocated onto stack
    - Utilizes heap for additional memory
    - Switches execution to memory location of function's code.
  - When function is complete.
    - Deallocates local variables on stack
    - Recovers previous execution address
    - Switches execution back to previous execution address.

Buffer Overflows

- Morris Worm (1988) was a buffer overflow.
- Storing data in memory that has not been allocated.
  - Stack-based
  - Heap-based



"Hello World" requires 13 bytes
Only 8 bytes allocated

- Causes:
  - Poor programming
  - No boundary checking
- Effects:
  - Typically causes program termination or system crash (denial of service)
  - Stack based attack.
    - Serious problems occur if it overwrites return address.
    - Redirect program to execute something else, such as a shell.
  - Heap overflow
    - Rewrite other portions of memory, including system calls.

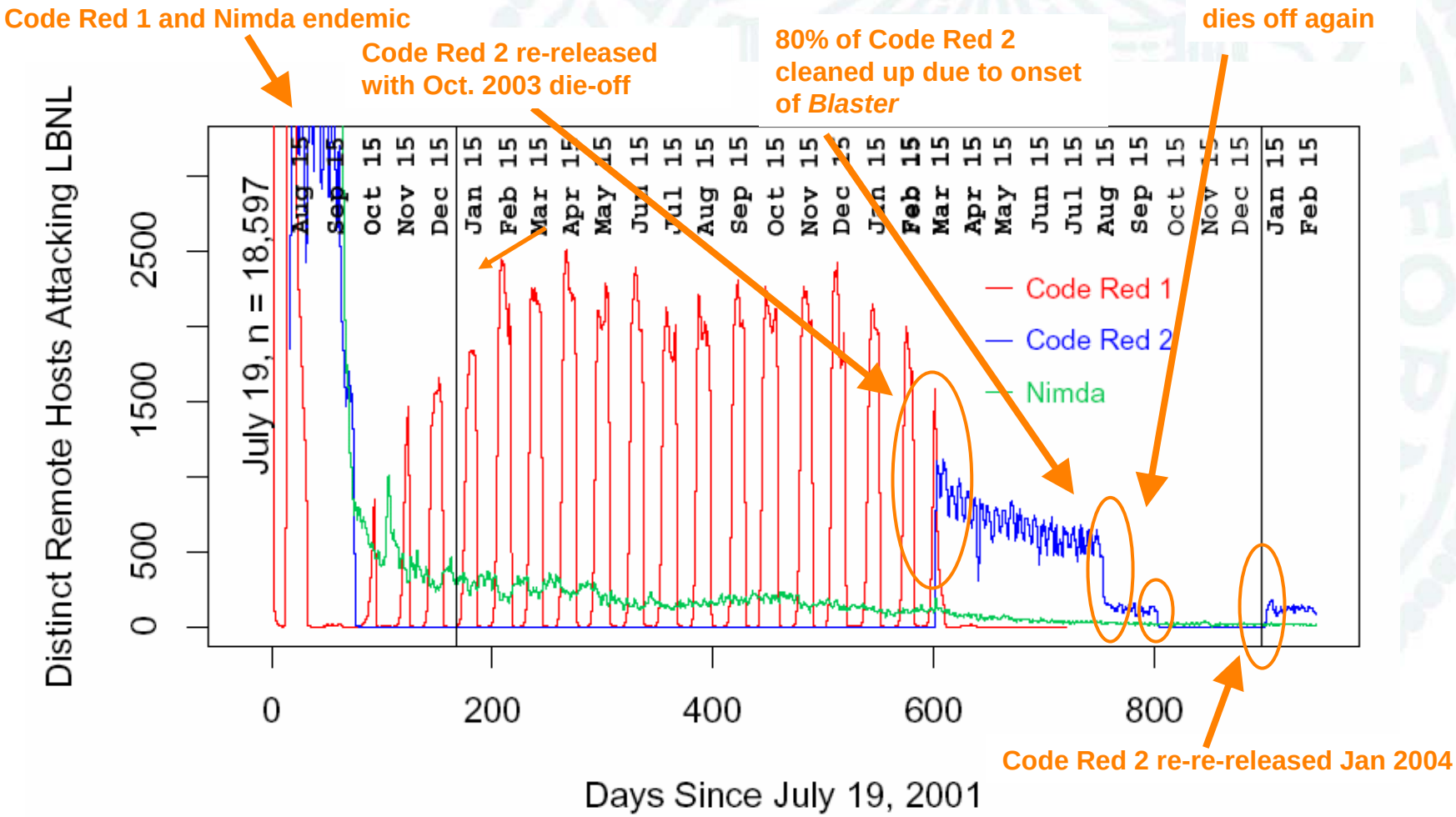
Viruses

- Self-replicating programs embedded in electronic files or in other executable.
  - Electronic File or executable is called a “host” or “carrier”.
- Require execution of program code for “infection”.
  - Either by opening a document or executing the program.
- Anti-virus filters
  - Scan files and executables for patterns
  - Some viruses slightly modify themselves to avoid detection.
- Viruses don’t spontaneously generate
  - Someone wrote it at some time.

Worms

- Self propagating program
  - Doesn't require a "host"
- Scans networks looking for vulnerable systems and infects them.
- Causes:
  - Vulnerable network services enabled
- Effects:
  - Network bandwidth consumption
  - Denial of service
  - Backdoors enabled
    - Allows for additional exploitation later.

Worm Ecosystems



Botnets

- Distributed networks of coordinated compromised systems
- Used for:
  - Denial of service
  - Spam flooding
  - Click fraud
  - Login credential mining
- Typical botnet behavior
  - Systems are compromised via worm or virus.
  - Systems checks in to a botnet server
    - Sometimes log into an Internet Relay Chat (IRC) channel
  - Systems can remain dormant until “activated” or can be continuously reporting information
- Botnets can be HUGE
  - 1.5 million system Botnet used to extort a company
- Most botnets are invisible and pervasive
  - Your site is probably a member of several botnets!
- Distributed nature makes it difficult to eradicate.

Scans

- Scans IP addresses looking for services or hosts.
  - Some scanners will look for a particular service, others for any service.
  - Locate systems running vulnerable services.
  - Sometimes for reconnaissance.

- Behaviors:
  - Attempt to exploit vulnerable services
  - Brute force login attempts
    - SSH scanners.

Rootkits

- Set of bundled software tools to gain privileged access to system and to *hide* it.
  - Ability to control a system without administrator being aware
- Many variant forms of rootkits.
  - Most do not need technical expertise to run.
- Defeats most local monitoring systems
  - You can not trust your operating system
- Essential to rebuild the system.
  - Impossible to clean since you can’t trust the system.

Rootkits

- Many perform process hiding
  - Example from SK Rootkit:

```
bash-2.05$ ps -aux | grep vi
user1 13779 1.6 0.0 5140 2336 pts/2 S 16:54 0:00 vim x
user1 13781 0.0 0.0 1736 576 pts/3 S 16:54 0:00 grep vi

bash-2.05$ ./sk i 13779
/dev/null
Detected version: 1.3b
Pid 13779 is hidden now!

bash-2.05$ ps -aux | grep vi
user1 13785 0.0 0.0 1736 580 pts/3 S 16:54 0:00 grep vi

bash-2.05$
```

Rootkits

- Some rootkits enable remote access

- Example:
  - Rootkit can listen on port 22 for specifically crafted packet to arrive.
  - When it arrives, rootkit establishes outbound connection back to host with root shell.

```
bash-2.05$ ./login -h localhost -d 22
/dev/null
Listening to port 32774
password:
Trying 127.0.0.1:22...
Trying...Et voila
Server connected. Escape character is '^K'
/dev/null
[root@owned .sk12]#
```

Password Cracking

- Attack user credentials
  - Usually requires obtaining password “hash”
- Attack methods
  - Dictionary attack
    - Try permutations against known words/passwords
  - Weak Encryption
    - Attack underlying encryption scheme
  - Brute Force
    - Try all possible combinations
  - Table lookup
    - Precompute hashes and perform table lookup
  - Hybrids
    - Combination of the above
- Example software:
  - LC5 (L0phtCrack)
  - Cain and Abel
  - Crack
  - RainbowCrack
  - John the Ripper
    - Works against LDAP

Background Radiation

- The Internet is a dangerous place.
- Constant barrage of malicious traffic occurring on the Internet.
  - Most of it automated or autonomous.
- Estimate that an unpatched system will be compromised within a few minutes.
- Network Crud
  - Always some network crud out there
- Backscatter
  - Hosts reflecting back RST's to forged IP addresses.
  - Can appear to be attacks
    - Many systems flag these as an attack.

University of California
San Francisco

Office of Academic and
Administrative Information Systems

HPC Security Protection Fundamentals

Stephen Lau
University of California, San Francisco
Stephen.Lau@ucsf.edu

November 12, 2006

Agenda

- Security Protection Fundamentals
- Defense in Depth
- Network Segmentation
- Vulnerability Scanning
- Firewalls
- System Monitoring
- Password Cracking
- Encryption

Security Protection Fundamentals

There is no silver bullet.

- Brief overview of common security protection techniques.
  - Not all techniques are covered.
- Each technique has strengths and weaknesses.
- Your mileage may vary.
  - Some sites will not allow certain techniques, i.e. password cracking.
- Security requires constant updating and changing of techniques to counter new threats.

Security is not a destination, it's a long, twilight struggle.

Confidentiality, Integrity and Availability

- What is the goal of Computer Security?
- Common definition:
  - Confidentiality, Integrity, Availability (CIA)
- Confidentiality
  - Prevent unauthorized disclosure or access to data and information resources.
- Integrity
  - Assure that the data or information resource is accurate and hasn't been changed.
- Availability
  - Assure that the data and information resources are available for use when needed.

Defense in Depth

- Use of multiple tools and techniques leverages off strengths and weaknesses
  - Multiple sensors to detect and prevent intrusions
  - No single points of failure
- No single tool or technique guarantees a problem free environment
- Protects against the “hard outer shell, soft inside” vulnerability
  - Caveat: More resource intensive to implement and maintain, integration difficulties

Defense in Depth Layers

- External Perimeter Defense
  - All points of entry into the network, the “DMZ”
- Internal Network Protection
- Host Level Protection
- User / Staff Protection
  - Education
- Physical Security

Network Segmentation

- Segment network by service and function
  - Physical or Virtual (VLAN)
- Investing in different networks for different functions is worthwhile
- Public and Private Networks
  - Clusters should be built out of private networks
  - There should be a few, well defined and configured access points in the cluster for public networks
  - Never assume your private network is really private
- User and Administrative networks
  - Keep user systems off infrastructure networks
    - i.e. security devices, servers

Firewalls

- Network traffic filters
  - Controls traffic between different zones of policy or trust.
  - Configurable to allow only certain types of traffic.

- Types of firewalls
  - Hardware
    - Blocking routers (ACLs)
    - Hardware appliance
  - Software
    - Host based firewalls
  - Application Layer Firewalls
    - Blocks specific applications.
      - i.e. all “ftp” or http traffic
  - Network Layer Firewalls
    - Blocks IP addresses, src ports, dst ports, etc.

- Standard configuration models
  - Default Deny / Allow some
  - Default Allow / Deny some

Problems with Firewalls

- Performance impact
  - Firewalls can potentially bottleneck
- Modification of traffic
  - Firewalls were designed for enterprises, not high performance computing environments.
  - Vendors bend corners to increase perceived performance
- What do you filter?
  - Academic/HPC network traffic atypical from enterprise traffic
  - End up with “screen door” effect or a locked down site with grumpy researchers.
- Firewalls should be deployed with consideration
  - Not just “because it’s industry standard”, because it’s not!

Vulnerability Scanning

- Use same tools attackers use to locate vulnerable systems and services
  - Run on regular basis to detect changes and new vulnerabilities.
  - Use to enforce minimum standards/compliance.
  - Sweep network for “banned” services: telnet, rlogin, etc.

- Popular scanning systems:
  - Nessus
    - http://www.nessus.org/
  - Nmap
    - http:/insecure.org/nmap/

- ***Danger, Will Robinson!***
  - ***Obtain permission from your system admin first!***
  - ***Scan your own network* FROM *your own network!***
  - ***Scanners* WILL *cause systems or services to crash!***
  - ***You have been warned!***

System Monitoring

- System logging and monitoring can be incredibly useful for forensics.
- Establish site-wide system logging capability.
  - Allow hosts to send logs to centralized logging server.
- Enable verbose logging on system daemons
  - sshd, httpd, ftpd, etc.
- Enable verbose logging under Windows Event Logging.
- Utilize system monitoring for system health.
  - Nagios
    - http://www.nagios.org/

Password Cracking

- Utilize password cracking tools against your user passwords.
- Regularly crack your passwords.
- Look for “test” passwords, default passwords, exceptions, etc.
  - These will show up!
- Example software: (again)
  - LC5 (L0phtCrack)
  - Cain and Abel
  - Crack
  - RainbowCrack
  - John the Ripper
    - Works against LDAP

Ensure that password cracking system is well protected!

Encryption

- Require SSH
  - Just say NO to plain text passwords!
  - Ban and disable telnet, rlogin, rcp, rsh.
- Ban and disable ftp where possible.
  - Use sftp or a VPN solution.
- Require SSL enabled IMAP/POP
- Utilize Virtual Private Network (VPN) technology for access to sites where possible.
- Utilize encrypted Instant Messaging when possible.
- Utilize PGP or GPG for email or encrypting files
  - http://www.gnupg.org/
- Explore whole disk encryption
  - Primarily for workstations, mobile devices.

U.S. Department of Energy

Office of Science

Intrusion Detection Systems

Brian L. Tierney
Lawrence Berkeley National Laboratory

Agenda

- Overview of various types of Intrusion Detection Systems
- Intrusion Detection using NetFlow information
- Intrusion Detection using Snort
- **Bro Intrusion Detection System**
- Intrusion Detection at 10 Gbps

Terminology

- IDS: Intrusion Detection System
- NIDS: Network-based IDS
- HIDS: Host-based IDS
  - IPS: Intrusion Protection System

Types of IDS's

- Network IDS (NIDS)
  - Monitors network traffic
    - Usually at the DMZ or at network boarders
    - Connected to a passive network tap or a switch configured with port mirroring
    - Analyzes traffic and / or traffic patterns
  - Popular Open Source Tools:
    - Snort: http://www.snort.org/
    - Bro: http://www.bro-ids.org/

Types of IDS's (cont.)

- Host IDS (HIDS)
  - Monitors changes of sensitive files on a host
    - Compares the checksum of system files against a database if known good checksums
    - Some HIDS also perform log analysis, process accounting analysis and rootkit detection
  - Popular Open Source Tools:
    - Tripwire: http://sourceforge.net/projects/tripwire
    - OSSEC http://www.ossec.net/

- Ideally, you should deploy both HIDS and NIDS
  - The HIDS might find things that slip past the NIDS

HIDS vs NIDS

- Advantages of Host IDS
  - Can detect insider attacks that don't traverse the network boundary
  - Dedicated hardware is not required

- Disadvantages of Host IDS
  - Can be difficult to deploy / maintain in a highly diverse environment
  - Can't look inside encrypted traffic
  - Really good hackers can hide from a Host IDS

HIDS vs. NIDS

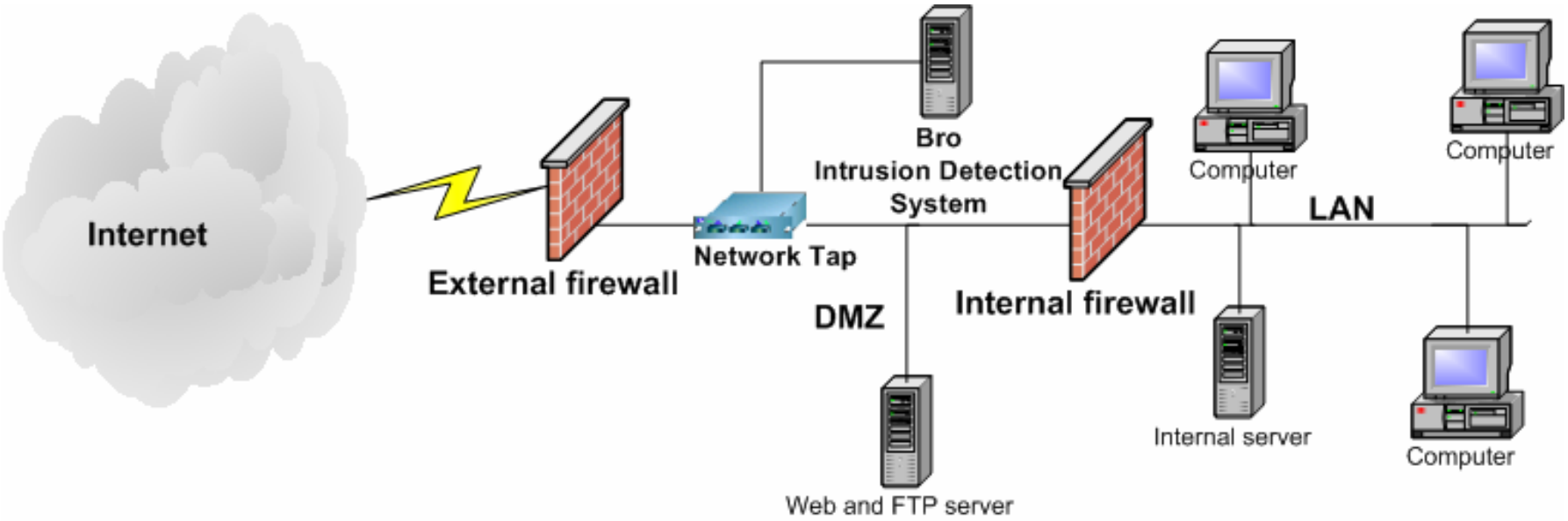
- Advantages of Network IDS
  - Transparent to users
  - No impact on hosts
  - Can detect attacks based on malformed packets
- Disadvantages of Network IDS
  - Hard to do on very high speed links

Passive vs. Active NIDS

- Passive: system generates an alert when it detects a security issue
- Active (or reactive): system responds by:
  - Blocking traffic, or
  - Terminating the connection

- Active NIDS that block the traffic are sometimes call an “Intrusion Prevention System”, or IPS

Sample NIDS Deployment



- One of many possible ways to deploy a NIDS

Styles of Network Intrusion Detection

- Signature Detection: look for specific, known attacks.
  - Example (from Snort):
    - content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
    - msg:"EXPLOIT x86 linux samba overflow"
  - Most commercial systems (e.g.: ISS RealSecure) use signatures

- Anomaly-detection: attacks are peculiar.
  - Approach: build/infer a profile of “normal” use, flag deviations.
  - Very hard to make work in a open science environment
    - Too many false positives

- Activity-based: look for activity that deviates from site policy.
  - Examples:
    - user joe is only allowed to log in from host A.
    - Only hosts A and B are allowed to receive email
  - This is the primary approach used by Bro

Limitations of an NIDS

- Can't see encrypted traffic
  - Must rely on traffic patterns instead
    - Requires that the site security personnel know what "normal" traffic patterns look like
  - Some applications (e.g.: Skype) make this difficult (on purpose)
  - Encapsulated / tunneled traffic hard to analyze

Sample Skype Connection Log

Time	Duration	Dst host	Service	Src port	Dst port	Prot	Sent	Rcv	State
13:44:47	0.320	84-72-129-184.dclient.hispeed.ch	other	18428	5033	udp	31	18	SF
13:44:46	6.1653	d60-65-146-178.col.wideopenwest.com	other	18428	11569	udp	93	18	SF
13:44:56	0.1801	h8441230065.dsl.speedlinq.nl	other	18428	42149	udp	36	18	SF
13:44:15	45.591	12-216-38-190.client.mchsi.com	other	18428	27225	udp	59	712	SF
13:44:15	45.633	d150-160-95.home.cgocable.net	other	18428	52795	udp	59	712	SF
13:45:00	0.107	d-65-175-169-113.metrocast.net	other	18428	2339	udp	30	356	SF
13:45:01	0.095	189-21.26-24.se.res.rr.com	other	18428	59271	udp	32	18	SF
13:45:01	0.105	c-68-34-153-203.hsd1.fl.comcast.net	other	18428	26510	udp	32	356	SF
13:45:01	0.320	cust296.midd.cable.ntl.com	other	18428	44839	udp	32	18	SF
13:45:01	0.249	71-15-124-26.dhcp.ftwo.tx.charter.com	other	18428	29525	udp	32	18	SF
13:46:01	2.387	c529c13f1.cable.wanadoo.nl	other	2949	58730	tcp	112	48	SF
13:46:01	2.528	85.201.151.18	other	2950	42520	tcp	109	83	SF
13:44:47	18.974	88-107-216-238.dynamic.dsl.as9105.com	other	18428	30066	udp	93	18	SF
13:45:50	21.918	dhcp-0-13-d4-35-6a-e5.cpe.cabletv.on.ca	other	2948	31413	tcp	114	54	SF
13:45:50	21.95 1	85.201.151.18	other	2947	42520	tcp	143	67	SF

NetFlow

- NetFlow “flow” records contain:
  - Source IP, Dest IP, port, start/end time, TCP flags, packet count
  - 1 for each direction, so a session will contain 2 (or more) flows
- Netflow connection logs can also be used for security analysis
  - Scan detection
  - Denial of Service attacks
  - Forensics
- Sample tools:
  - Flow-tools: http://www.splintered.net/sw/flow-tools/
    - flow-dscan, flow-scan-report, flow-host-profile
  - NERD (Network Emergency Responder & Detector)
    - http://www.nerdd.org/
  - NFSEN: http://nfsen.sourceforge.net/

Netflow: flow-dscan

- Can use flow-dscan to detect scanning activity
- Reports:
  - Flows with excessive packet count (floods)
  - Sources contacting more than N destination IPs (host scans)
  - Sources contacting more than M destination ports on a single host (port scans)
- Uses source and destination white-lists to eliminate false positives (e.g.: web page ad sites, gaming sites, etc).
- Can run on archived data or live feeds

NetFlow: NERD Example

NERD
Network Emergency Responder & Detector

Alarms | Analysis | Settings

Alarms of Wednesday, September 8th 2004

Previous day | Next day | Search for Alarms

Starttime (GMT+1)	Stoptime (GMT+1)	Rulename	Alarm message	Trigger
Wed, Sept 8th, 11:00	Wed, Sept 8th, 12:31	Flood detection	Source IP address 10.0.0.138 with Destination IP address 192.168.0.2 has 12832 connections in 5 minutes	9000 connections in 5 minutes Analyse
Wed, Sept 8th, 12:50	* Not stopped *	Portscan detection	Source IP address 127.0.0.1 has 20316 destination ports in 5 minutes	15000 destination ports in 5 minutes Analyse

Snort NIDS

- Most popular open-source NIDS
  - Lots of add-on tools for log analysis
    - ACID (Analysis Console for Intrusion Detection), Barnyard, more
  - Support for most database backends
    - mySQL, postgress, Microsoft SQL, Oracle
  - Commercial support (SourceFire)
  - New signatures released daily
  - Rules can contain Perl-compatible regular expressions (PCRE)

- Sample Rule:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
   flow:to_server,established
content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"
msg:"EXPLOIT x86 linux samba overflow"
reference:bugtraq,1816
reference:cve,CVE-1999-0811
classtype:attempted-admin
```

U.S. Department of Energy

Office of Science

Bro Network Intrusion Detection System

Brian L. Tierney
Lawrence Berkeley National Laboratory

Bro's Design Targets Open Research Environments

- 10 years have been invested in optimizing Bro for Open Environments
  - In production use at a number of sites:
    - LBL, NERSC, ESnet, NCSA, UC Davis: Primary IDS
    - Sandia, UCB, UCSD, UCSB, Ohio State, TU Munich, OCCS(ORNL), NOAA, more: Secondary IDS
  - Runs on commodity hardware
  - Provides real-time detection and response
  - Ability to monitor traffic in a very high performance environment
  - Ability to write custom policy analyzers

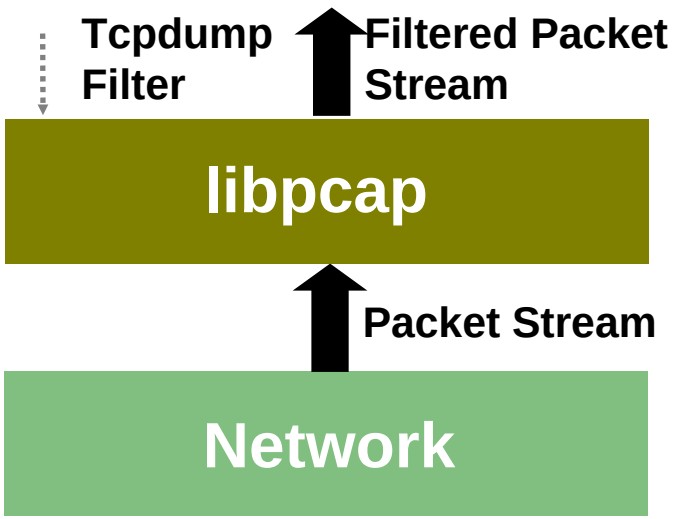
Bro's Use at LBL

- Operational 24×7 since 1996
- Monitors traffic for suspicious behavior or policy violations: incoming/outgoing/internal
- In conjunction with blocking routers, Bro acts as a dynamic and intelligent firewall
  - Blocks access from offending IP addresses
  - Terminates connections and/or sends alarms
  - Blocks known hostile activity
  - Locates site policy violations (e.g.: Kazaa and gnutella)
- Valuable Research Tool
  - Many network characterization studies based on Bro

How Bro Works

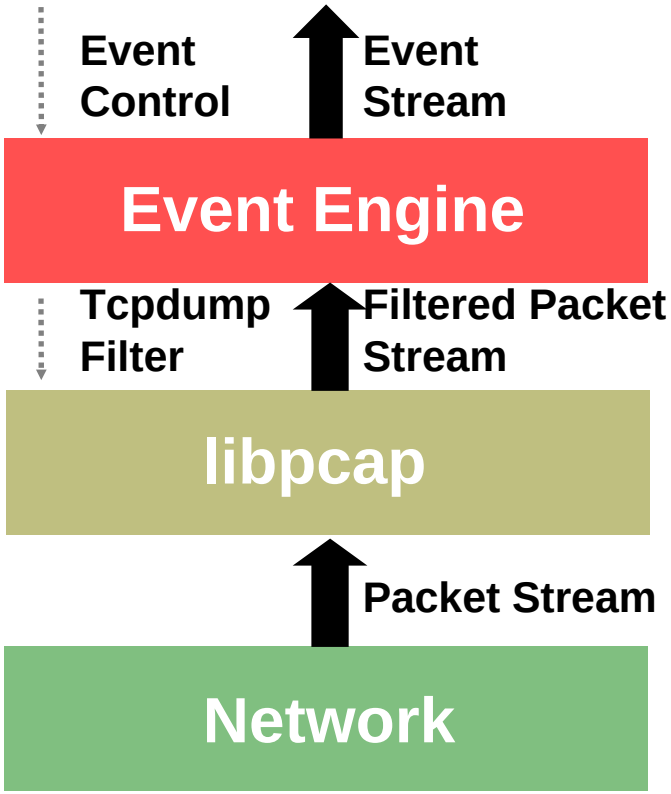
- Taps GigEther fiber link passively, sends up a copy of all network traffic.

How Bro Works



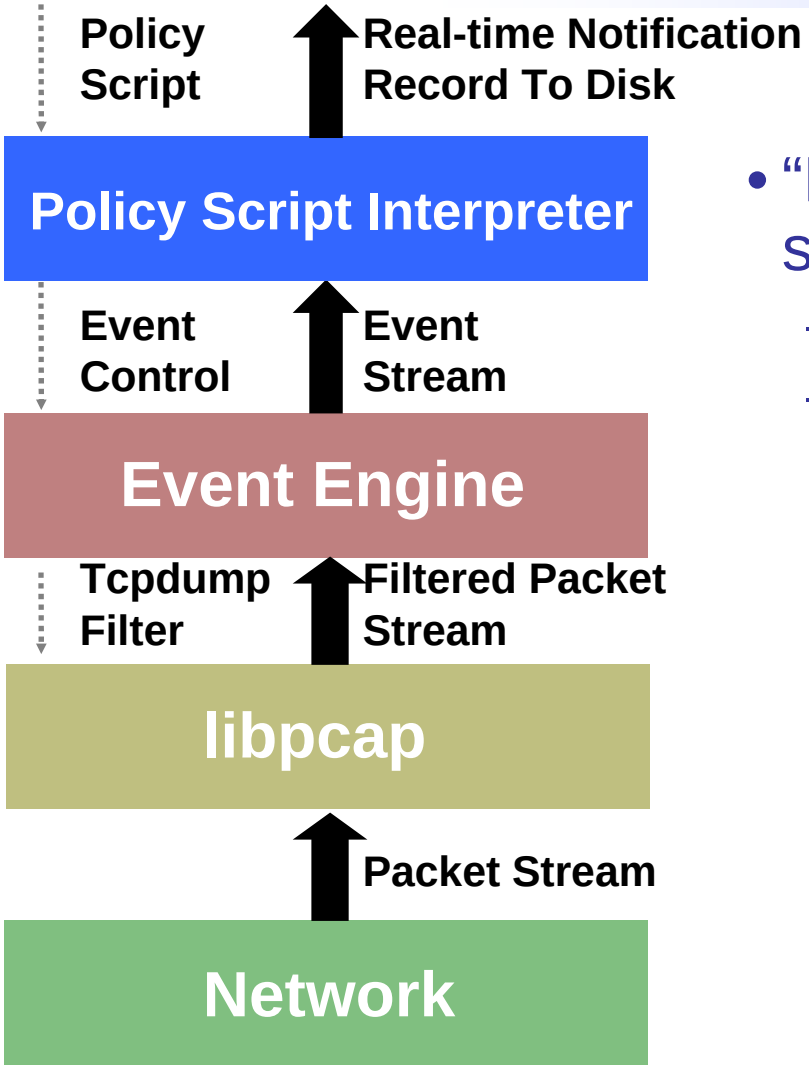
- Kernel filters down high-volume stream via standard *libpcap* packet capture library.

How Bro Works



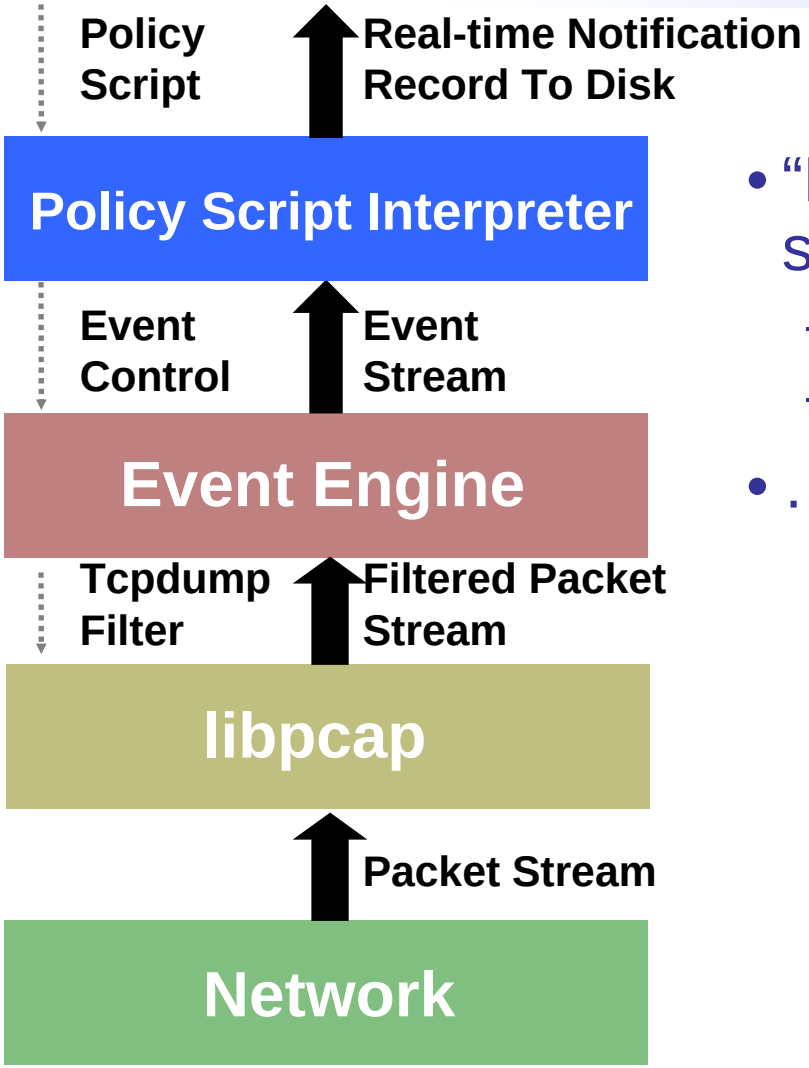
- "Event engine" distills filtered stream into high-level, *policy-neutral* events reflecting underlying network activity
  - E.g. Connection-level:
    - connection attempt
    - connection finished
  - E.g. Application-level:
    - ftp request
    - http_reply
  - E.g. Activity-level:
    - login success

How Bro Works



- "Policy script" processes event stream, incorporates:
  - Context from past events
  - Site's particular policies

How Bro Works



- "Policy script" processes event stream, incorporates:
  - Context from past events
  - Site's particular policies
- … and *takes action*:
  - Records to disk
  - Generates alerts via *syslog*, paging, etc.
  - Executes programs as a form of <u>response</u>

Bro Protocol Analyzers

- Bro includes the following protocol analyzers
  - Full analysis:
    - HTTP, FTP, telnet, rlogin, rsh, RPC, DCE/RPC, DNS, Windows Domain Service, SMTP, IRC, POP3, NTP, ARP, ICMP, Finger, Ident
  - Partial analysis:
    - NFS, SMB, NCP, SSH, SSL, TFTP, Gnutella
  - In progress:
    - AIM, BGP, DHCP, Windows RPC, SMB, NetBIOS, NCP

Sample Bro Policy

- Using the Bro language, sites can write custom policy scripts to generate alarms on any policy violation.
- For example, if a site only allows external http and mail to a small, controlled lists of hosts, they could do this:

```
const web_servers = { www.lbl.gov, www.bro-ids.org, };
const mail_servers = { smtp.lbl.gov, smtp2.lbl.gov, };

redef allow_services_to: set[addr, port] += {
        [mail_servers, smtp],
        [web_servers, http],
};
```

- Bro can then generate an *Alarm* or even terminate the connection for policy violations:

```
if ( service !in allow_services)
   NOTICE([$note=SensitiveConnection, $conn=c,]);
 if ( inbound && service in terminate_successful_inbound_service )
      terminate_connection(c);
```

Sample Bro Alarms

```
Nov 16 03:31:23 AddressDropped low port trolling a213-22-132-
   227.netcabo.pt 258/tcp
Nov 16 06:25:23 SensitivePortmapperAccess rpc: cs4/917 >
   guacamole.cchem.berkeley.edu/portmap pm_dump: (done)
Nov 16 06:30:49 AddressScan 66.243.211.244 has scanned 10000
   hosts (445/tcp)
Nov 16 06:30:50 SensitiveConnection hot: neutrino 200b >
   147.8.137.149/telnet 463b 14.2s "root"
Nov 16 06:30:50 OutboundTFTP outbound TFTP:
   sip000d28083467.dhcp -> inoc-dba.pch.net
Nov 16 06:30:52 SensitiveConnection hot: 198.128.27.21 560b >
   208.254.3.160/https 4202b 0.5s <IRC source sites>
Nov 16 06:30:53 WormPhoneHome worm phone-home signature mcr-88-
   4 -> 218.146.108.51/9900
Nov 16 06:30:56 FTP_Sensitive ftp: fun.ee/3766 >
   soling.cs.vu.nl/ftp #1537 RETR nfsshell.tar.gz (complete)
Nov 16 06:32:38  HTTP_SensitiveURI scan1/34462 >
   198.128.27.212/http %988: GET /admin/file_manager.php?action=
   download& filename=./../../../../../../etc/passwd <no reply>
```

U.S. Department of Energy

Office of Science

Recent Advances in Bro

Brian L. Tierney
Lawrence Berkeley National Laboratory

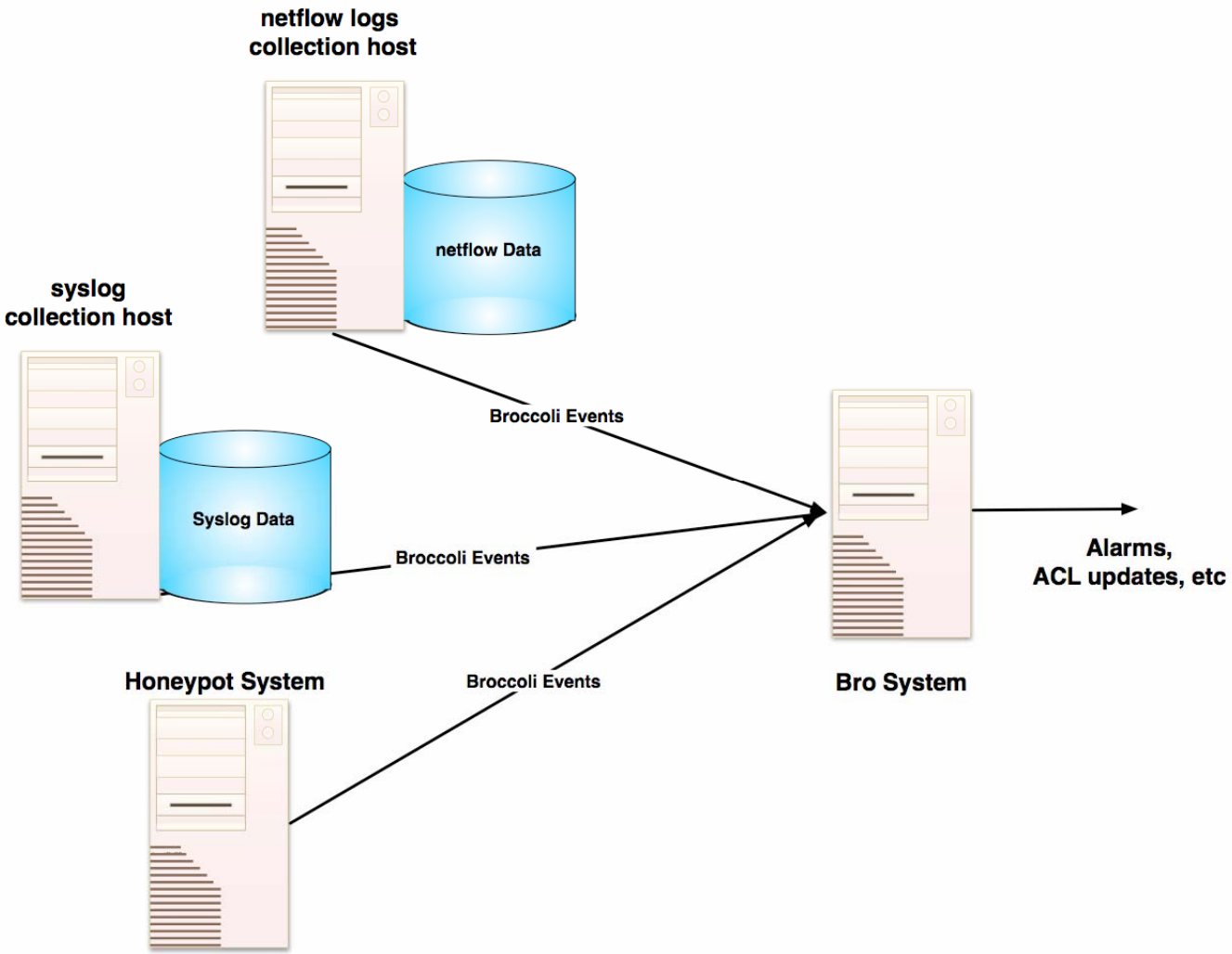
Bro Communication

- **Bro co**mmunication **li**brary (Broccoli)
  - Multiple Bro's can now communicate and exchange "events"
  - Can easily synchronize bro state between running instances of Bro
    - Just add "&synchronized" to the table declaration
      - `global scan_hosts: table[addr] of count &synchronized;`
    - This can be used to maintain a table of known hostile hosts between multiple instances of Bro

Example Uses of Broccoli

- Using Broccoli to send syslog events to Bro
  - LBNL runs a central syslog collector for entire lab
    - Subset of these logs are sent to Bro for analysis (ssh, su, sudo, etc.)
  - Bro policy is being used to analyze syslog logs
    - E.g.: multiple ssh login failures, offsite root logins, etc.
- Using Broccoli to send honeyd events to Bro
  - LBNL runs a honeypot to look for insider attacks
    - Watches unused address space for connection attempts
  - Leverages Bro's alarm/block mechanisms

Sample Broccoli Use



Sources: How the Generic Client Works

(1) Start with Syslog message

```
Mar 10 18:39:41 2.3.4.5 Message forwarded from host609:
sshd[74334]: Accepted password for user1 from 1.2.3.XXX port
45674 ssh2
```

(2) Parse into clean form

```
ssh_login double=1108696916 addr=10.11.12.13
addr=10.14.15.16 string=user1 string=publickey
```

(3) **Bro Generic Client** reads text and communicates to Bro

BRO

(4) Bro acts on communication

Correlating Hostile Network Traffic From Multiple Sites

- A comparison of hostile hosts scanning LBL and NERSC address space shows little overlap
  - 8% overlap per day, and 25% per week
  - By sharing this information between site we can block hostile hosts faster

- Using Bro we can correlate cross site activity for:
  - Network Connections
    - Scanning
    - Hostile HTTP activity
  - Hostile activity derived from syslog
  - Snort Rules
  - In general, any event identifiable by Bro

Dynamic Application Detection

- Current NIDS system require you to specify which protocol analyzer to use for a given port.
  - I.e: port 25 = SMTP; port 80 = HTTP, port 6666 = IRC, etc.
- NIDS's only look at traffic on ports they know how to analyze
- New version of Bro supports dynamic port selection
  - Uses simple protocol-specific signatures to try to guess what protocol is being seen
  - Sample use at LBL:
    - Look for http proxies
    - Look for IRC, FTP and SMTP on non-standard ports
    - Looks for IRC "botnets"
    - payload inspection of FTP data transfers
  - Note: dynamic application detection takes more CPU and IO because it looks at all traffic,
    - may need a dedicated Bro host for this at medium/large sites

Dynamic HTTP Analyzer

- HTTP analyzer can distinguish the various protocols that use HTTP as their transport protocol by looking for their characteristics
  - Includes patterns for detecting Kazaa, Gnutella, BitTorrent, Squid, and SOAP applications running over HTTP
  - The HTTP analyzer extracts the “Server” header from the HTTP responses
- Examples:
  - `ProtocolFound 66.249.65.49/62669 > 131.243.224.47/1400 FileMakerPro (via HTTP) on port 1400/tcp`
  - `ProtocolFound 66.249.66.177/47957 > 131.243.2.93/8881 Apache (via HTTP) on port 8881/tcp`
  - `ProtocolFound 198.129.90.45/1160 > 128.3.72.29/7777 Oracle (via HTTP) on port 7777/tcp`
  - `ProtocolFound 211.37.103.215/1278 > 131.243.129.75/554 RealServer (via HTTP) on port 554/tcp`

Payload inspection of FTP Data Transfers

- Attackers often install FTP-servers on non-standard ports on compromised machines
- Analysis of FTP data connections is impossible with traditional NIDSs
  - FTP uses arbitrary port combinations for data connections.
- The Bro file analyzer receives the file's full content and can utilize any file-based intrusion detection scheme.
  - includes file-type identification to Bro using libmagic
    - can identify a large number of file-types
    - E.g.: Bro is now able to categorize a data file as being of MIME type video/x-msvideo (an AVI movie)

Detecting IRC-based Botnets

- Attackers systematically install trojans bots on compromised machines
  - Large botnets provide remote command execution on vulnerable systems across the world.
- A botnet is usually controlled by a master that communicates with the bots by sending commands.
  - E.G.: flood a victim, send spam, or sniff confidential information such as passwords
- Botnets are one of the largest threats in today's Internet
- The IRC protocol is popular for bot communication
  - It is extremely difficult for a traditional NIDS to reliably detect members of IRC-based botnets.
  - Often, the bots never connect to a standard IRC server port.

Botnet Detector

- The detector sits on top of the IRC analyzer and is therefore able to perform protocol-aware analysis of all detected IRC sessions.

- To identify a bot connection, it uses three heuristics.
  - First, it checks if the client's nickname matches a (customizable) set of patterns known to be used by botnets
  - Second, it examines the channel topics if it includes a known typical botnet commands
  - Third, clients that establish an IRC connection to an already identified bot-server are also considered to be bots.
    - This is very powerful as it leverages the state that the detector accumulates over time, and does not depend on any particular payload pattern.

- Successfully located several botnets at Technical University Munich and UC Berkeley

Bro at 10Gbps

- There are a number of approaches to Bro at 10 GigE
  - Use a 10 Gig NIC
    - Pros: will just work; no special configuration needed
    - Cons: cost, even with very high end host will run out of CPU cycles for analysis
    - Performance: Current testing shows that 10 Gig Nics can only capture about 2 Gbps of traffic
  - Use Filter Based Port Mirroring or VACLs.
    - Forward only interesting traffic to a Bro host
    - Pros: works well with standard Bro configuration
    - Cons:
      - requires Juniper routers or CISCO 65xx router;
      - must know which ports to forward,
      - may not be able to filter down to only 1 Gbps of traffic

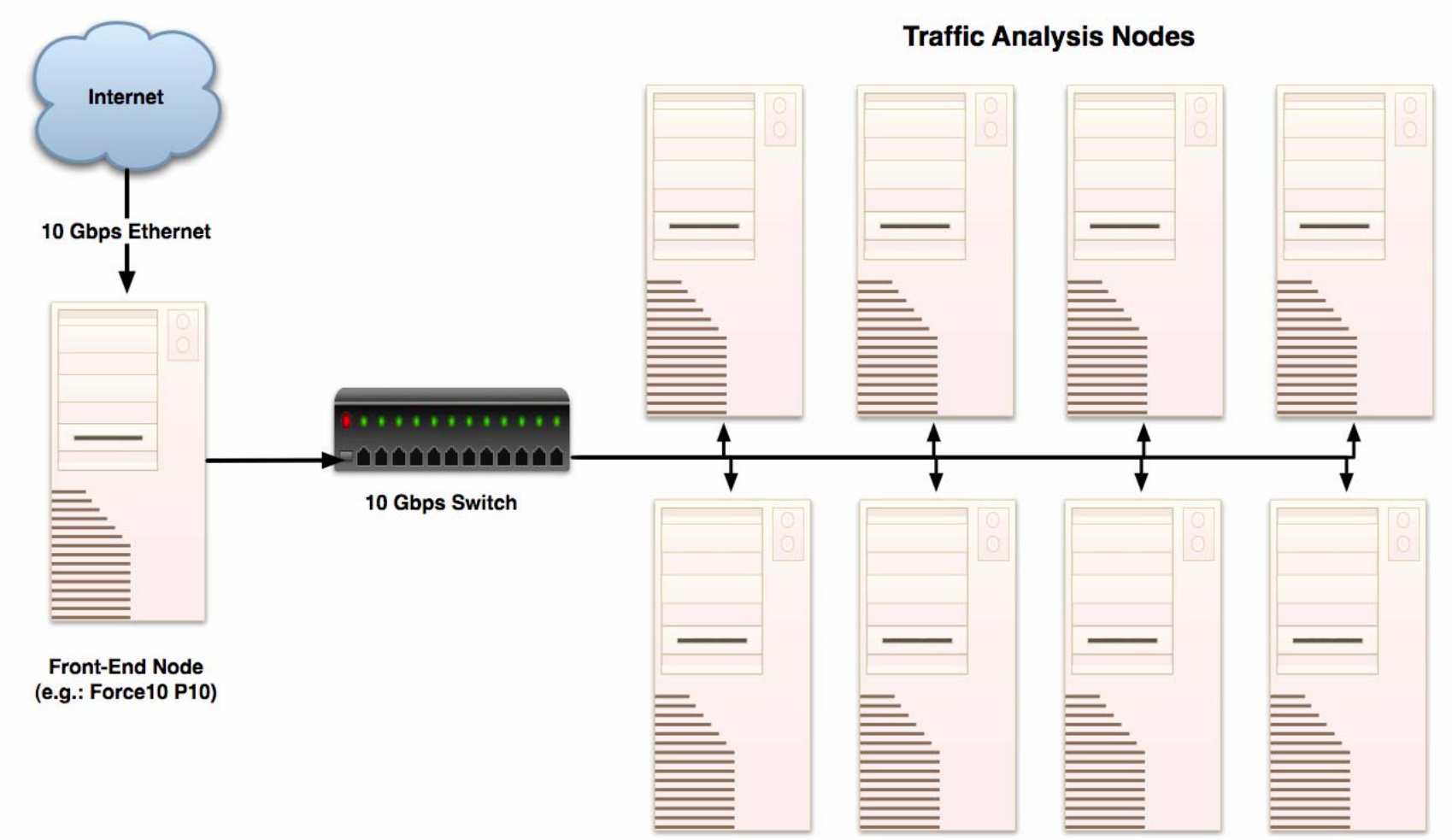
Bro at 10Gbps (cont.)

- Use Custom Hardware
  - e.g.: Force10 P10 (formerly Metanetworks) provides on line-rate capture, filtering, and blocking at 10 Gbps
    - Can configure the P10 to only pass the first N packets and the last packet of each connection to Bro
- Bro Cluster

10 Gig Bro Cluster

- Front end node uses Force10 hardware to rewrite packets, forwarding them to cluster nodes for analysis
  - Classify packets and forward to specific ethernet address
- The Approach is scalable in several ways:
  - As traffic increases, just add more nodes
  - For more detailed Bro analysis, just add more nodes
  - Should scale to 100 Gbps networks and beyond
  - Can run Bro, Snort, and other future tools all on the same cluster
- Require new Bro synchronization feature to maintain state across multiple Bro systems
- Front end node distributes traffic evenly based on a hash of source IP, dest IP, and port
- Currently deploying a cluster a LBL and UC Berkeley
- Will demonstrate this at SC06

Bro Cluster


Traffic Analysis Nodes
Internet
10 Gbps Ethernet
10 Gbps Switch
Front-End Node
(e.g.: Force10 P10)

Bro's use at SCinet/SC06

- Bro is used to detect attacks at SC06 conference.
  - Bro has been in use as SCinet's primary IDS since 2000.
- Bro monitors SCinet links.
  - SC06 SCinet links (as of 10/06)
    - 3 OC192's
    - 7 10G links
    - 1 1G link
    - OC-3 commodity
- Standard Bro on SCinet commodity link
  - SCinet's connection to commercial Internet services
- Experimental 10G Bro integrated with Force10's T10 appliance

Other Bro Uses

- Use Bro to detect insider attacks
- Deploy Bro on every subnet
  - Using $60 Linksys routers running Linux
- If Bro detects malicious activity
  - Contacts the DHCP server, which then will not renew the DHCP lease
    - Sends the host to a web site explaining why its been blocked
    - Current DHCP lease time = 2 hours
- More info at: http://www.bro-ids.org/linksys.html

For More Information

- Bro is Open Source
  - FreeBSD-style license
  - Download from: http://www.bro-ids.org/
- Questions: email bro@bro-ids.org

Keeping One Step Ahead: The Changing Computer Protection Environment

Scott Campbell
Lawrence Berkeley National Laboratory
NERSC
scampbell@lbl.gov

Introduction

General Topics:

- Section Overview
- Soothing Platitudes
- Evolution of Attacks
- Failure of current defensive strategies
- Multi-Site HPC Implications

Section Overview

What is this all about?

We will look at an approach to HPC security which addresses the question of why Johnny Can't Secure the Data Center ...

Soothing Platitudes

- General ideas to help frame the topics we are looking at.
- They may get less soothing as we go along…

Soothing Platitude #1

Do not make assumptions about what your network and systems look like.

Measure Them Inside and Out

- **Noise and Radiation**
  - **System Logs and Network Behavior**
- **Complexity**
- **Familiarity with "Normal" is Critical**

Soothing Platitudes #2

We live in a *security ecosystem:* improvements made by you will force attackers to change and their tactics.

Results can be seen in:

- **Unexpected Consequences**
- **New Attacks (Classic: Jumping Higher)**
- **New Attacks (Weird: Learning to Fly)**

Soothing Platitudes #3

Given enough time, all systems (most importantly yours) will fail.

- **See Soothing Platitude #2.**
- **Design for "graceful failure".**
- **Ideological Stance, nothing personal.**

Why Are Attacks Evolving?

Q: Why are Attacks Evolving?

A: Because Security is doing it's job.

Example: Cleartext Authentication

Problem: Adversary sitting at any point between two systems can steal credentials.

Threat: Systems *and Network from dozens of sources.*

Evolution 101.1

<u>Solution!</u>: Encrypt session, authenticate server to user.

<u>Hacker Problem</u>: Cheap and Easy food source gone. Must create new attack vector from the *remaining options*, or starve to death from credential depletion.

<u>Hacker Solution</u>: Replace SSHD.

Up From the Mud...

Continuing Escalation!

Trojan SSHD =>Tripwire

Tripwire => Kernel Modules

Kernel Modules => OTP

OTP => Channel Manipulation

Catch 22

Remember: We live in a security ecosystem: improvements made by you will force the hacking community to change their tactics.

Catch 22 – You have to fix a problem, but that solution will create more complex children.

Failure of Current Defensive Techniques

Changing character of threats over time: Threats do not go away, but are constantly being augmented by new developments.

Example: Firewall

Firewalls are used to limit network connectivity and provide simple protocol analysis.

If you can not get to the inside of a network, that network is safe. Hooray!

... Firewalls ...

- BUT -

Users LOVE to click on “things” - Some % will always go for goofy web sites or joke executables.

... Firewalls ...

It gets worse: no need to click!

- Infected images.
- Hostile web content.
- Javascript, java and activeX attacks.
- Attacks via DNS Name Resolution.

... Firewalls ...

Attacks now reach your staff and users via services essential for normal operation.

This is why existing protection Techniques are failing. Re-Redux on Soothing Platitude #2 ...

Implications to HPC

Shared environment amongst HPC Sites:

- **Large numbers of shared users.**
- **Similar environments.**
- **Similar software and open networks.**

Further Implications

- **Some of your accounts are *already* compromised.**

- **This means you will be attacked from the inside – be prepared for this.**

  Example: Linux Kernel attack(s) in Mid June.

Conclusion

Coordination and communication is essential between HPC centers, since getting out of sync with the larger HPC community will leave you open to new attacker strategies.

Do NOT be the slow antelope.

Keeping One Step Ahead

Looking for a Few Good Ideas

Scott Campbell

Lawrence Berkeley National Laboratory
National Energy Research Scientific Computing Center
scampbell@lbl.gov

Introduction

General Topics:

- Credential Theft and OTP
- Host Level Monitoring and Syslog
- Distributed IDS and Instrumentation
- Auditing SSH activity

Credential Theft

General Problem: As long as some sort of authentication mechanism has been used, someone has wanted to *steal* it.

Passwords

Generally despised by users and security people alike…

Problems:

- Theft: easy to steal, trojan/spyware etc.
- Reuse: ssh/telnet/ftp/http/imap/IM.
- Steal once, use often!

Authentications Big 3

- Something you know (password).
- Something you have (key or token).
- Something you are (biometric).

Growth in 2 (or even 3) factor authentication.

Problem: Theft Detection

- **Difficult to distinguish between authorized and unauthorized access.**

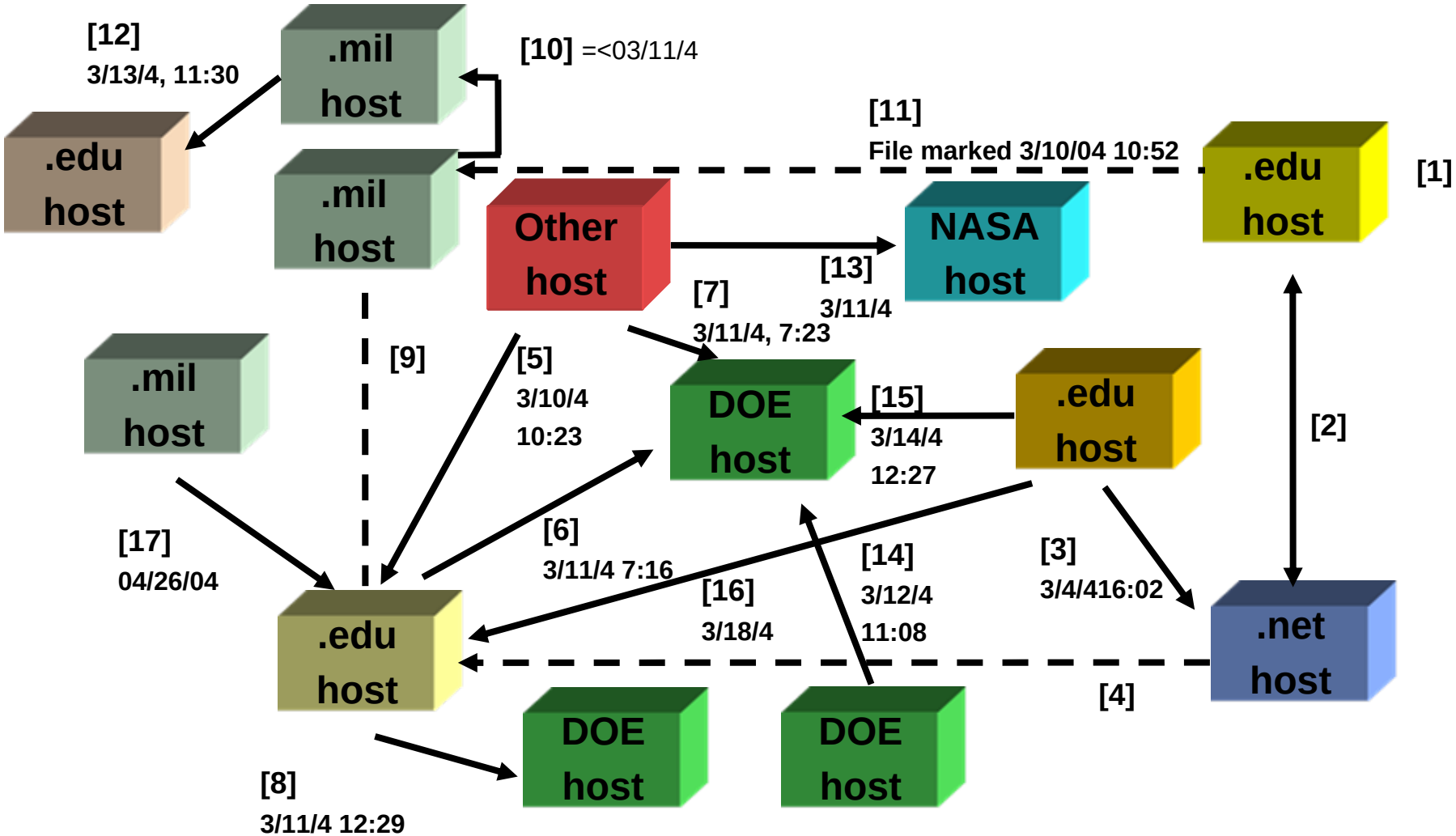
```
Mar 10 18:39:41 2.3.4.5 Message forwarded from host609:
   sshd[74334]: Accepted password for user1 from 1.2.3.XXX port
   45674 ssh2

Mar 11 11:00:50 2.3.4.6 Message forwarded from host013:
   sshd[117314]: Accepted password for user1 from 1.2.3.XXX port
   46570 ssh2
```

- **Network filtering/analysis has limited effect.**

- **Used widely in recent attacks.**

Example of Failure



Incident Details

What Worked For Attackers:

- Simple Password and Key Theft.
- Critical Mass of Systems Compromised.
- Human Error (configuration and patching).

Enter OTP

OTP: One Time Password – A mechanism used to augment user authentication by preventing credential replay. "Something you have."

OTP Details

Add authentication physically separate from the computer:

- **Info good only once – worthless to steal.**
- **Less dependant on host integrity.**
- **Authentication can be centralized.**

Problem of Credential Theft <u>completely resolved</u>. Hooray!

A Few Problems

Issues with introduction of OTP:

- Cross site collaboration: Logins at multiple sties. Token Bandelero ...
- Established authentication services: design inconsistent with OTP (X509/Grid Services).
- Initial human identification.
- Token maintenance and administration.

Can be resolved by careful design and administration.

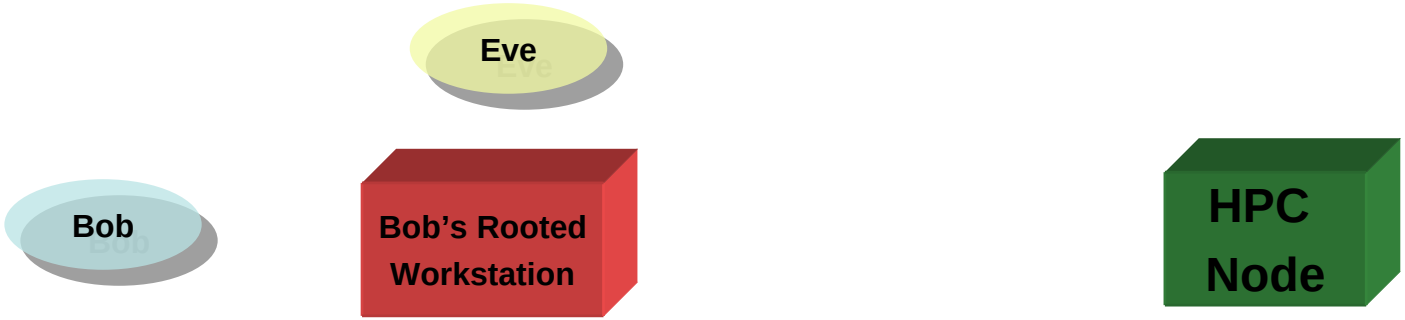
Soothing Platitudes Redux

Change in hacker perspective: view authenticated users as a *service*!

- **Ignore authentication theft.**
- **Let user log in with OTP, then take over their session.**
- **Details follow!**

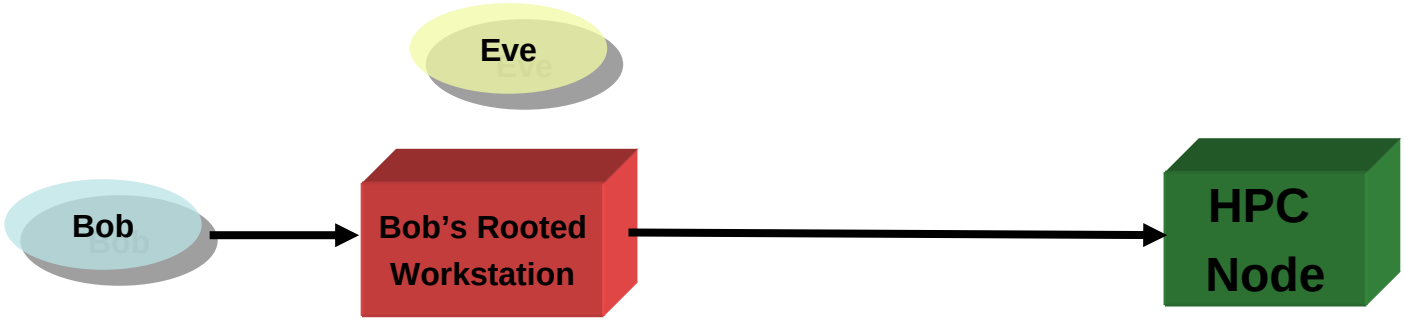
Point (again) is that hackers evolve to meet changes in your security environment.

Credential Theft and One Time Passwords



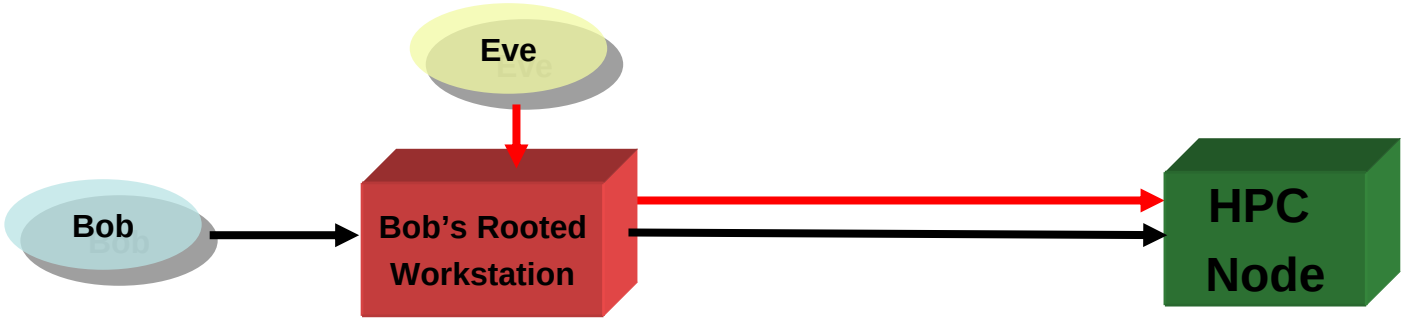
Unsuspecting user Bob is logging into a secure HPC interactive node from his university workstation. Unfortunately, an attacker, Eve, has replaced sshd on his host…

Credential Theft and One Time Passwords



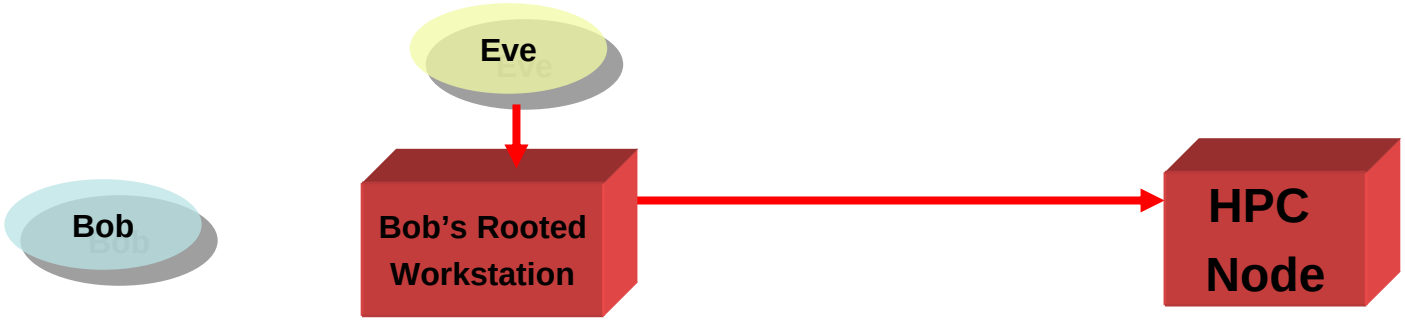
Bob authenticates to Host with OTP

Credential Theft and One Time Passwords



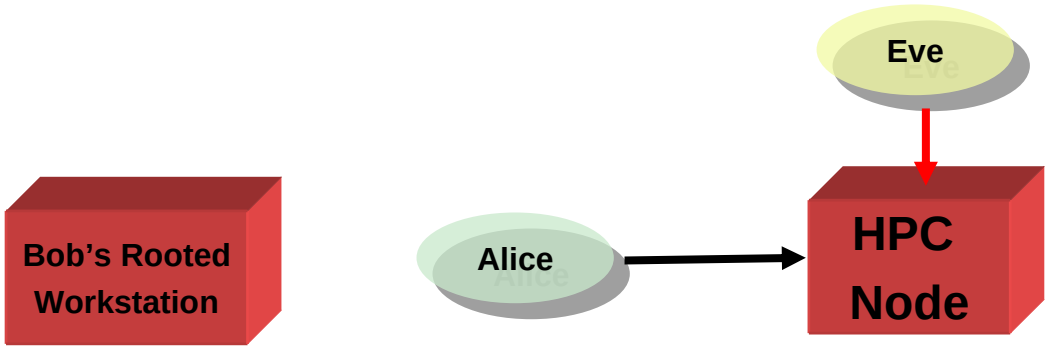
Eve hijacks or duplicates Bob's existing session.
He suspects nothing.

Credential Theft and One Time Passwords



Eve launches attack on HPC node via local root compromise as Bob. Upon success, she replaces sshd…

Credential Theft and One Time Passwords



Lather, rinse, repeat.

OTP Conclusion

OTP Implementations do an excellent job of dealing with credential theft.

- but -

You must be prepared for the response to this new mechanism by the hacker community.

Host Level Monitoring I

What do you miss when monitoring your border?

- Attacks From Inside
- Individual Host Interpretation
  - No idea what a host thinks it is seeing.
  - No idea what applications think they are seeing (ex: SSL and web servers).

Unix Host Monitoring Options

Syslog

Process Accounting

Home Grown Option

Proposed Solution: Syslog

Advantages

- Ubiquitous
- Well Understood
- Simple to Configure

Disadvantages

- UDP Based
- Free Form Data
- Insecure
- Analysis Complex in Large Data Volumes
- Performance Issues

Proposed Solution (II): SyslogNG

Advantages

- Ubiquitous
- Well Understood
- Simple to Configure?

Disadvantages

- Free Form Data
- Analysis Complex in Large Data Volumes

Data Analysis: The Real Problem

Flexibility of syslog solutions is also its weakness:

- **Unparseable data == Noise.**
- **No general solution to this problem.**

Many people working on a solution…

Analysis Continuum: Data Types

Look at a few Well Defined Quantities

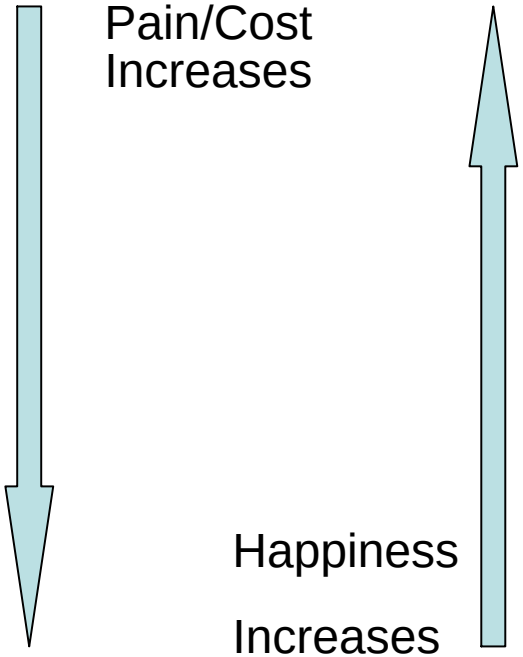
↕

View All and Extract Known Quantities

↕

Look at All and Make a General Parser

Analysis Continuum: System Distribution



Homogeneous Env:
Same Operating System,
Same Type and Version of
Network Equip etc

Semi-Homogeneous Env:
Well behaved logging behavior,
but significant uniform logging.
Limited interest in data.

Heterogeneous Env:
Many platforms, you want all
available data.

What We Have Done

Given Diverse Selection of System OS, versions and hardware, we analyze:

- **Logins**
- **SU Success and Failures**
- **Firewall Activity**
- **Grid related activity**

What to do with this data?

Integrating Different Data Sources into Bro!

Integrate the results into the site IDS!

Using the Bro Generic Client: *arbitrary structured data* can be fed into Bro.

Besides syslog data:

- **Firewall Logs**
- **Web Server Logs**
- **NetFlow Records**

Integrating Different Data Sources via Generic Client

(1) Start with Syslog message

```
Mar 10 18:39:41 2.3.4.5 Message forwarded from host609:
sshd[74334]: Accepted password for user1 from 1.2.3.XXX port
45674 ssh2
```

(2) Parse into clean form

```
ssh_login double=1108696916 addr=10.11.12.13
addr=10.14.15.16 string=user1 string=publickey
```

(3) **Bro Generic Client** reads text and communicates to Bro

BRO

(4) Bro acts on communication

Bro + Syslog Analysis

With this method, ssh login behavior across a site can be summarized, and integrated with regular connection records.

Example: If an external address attempts to log in as 'root', any non-ssh connections from or returning connections to that address are marked as hostile.

Distributed Intrusion Detection

What is meant by Distributed?

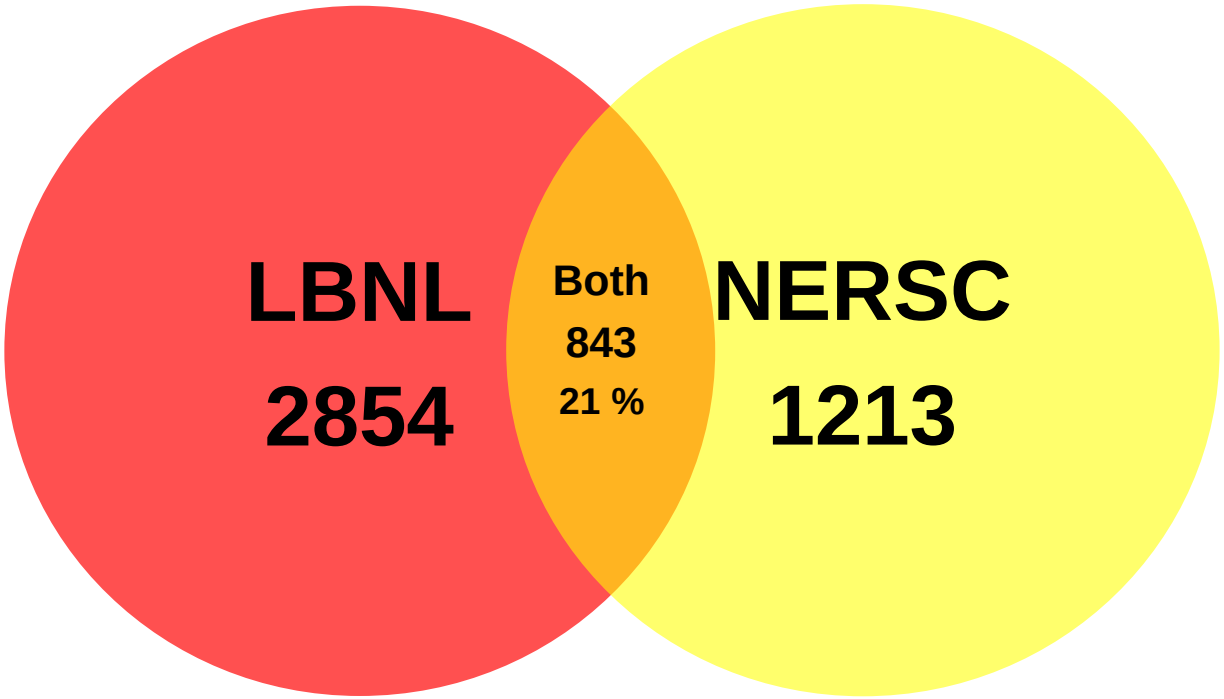
- Network Intrusion detection across multiple sites in real time.
- Integrating different data sources for security analysis.

Why do we care?

- More science is being done 'off site' and by collaborations.
- Scanning and other network attacks (worms, bots, account theft) are network agnostic.
- The hackers benefit tremendously from information sharing. We can do the same.
- Sites with better security can help others.

Scan Detection Across Multiple Sites

- Non HTTP/Windows Scans detected at LBNL/NERSC.
- Over 6 days.
- Number of unique IP addresses seen
- Around 21% of the scanning addresses seen by both address ranges during the time period.



Distributed DOE Science

Projects with 1000's of users and distributed resources are becoming more common:

- **Greater than 50% of DOE Office of Science PI's and facility users are at universities.**
- **Greater than 85% of Esnet traffic is to and from universities and other non-DOE facilities.**
- **18 of the top 20 flows in Esnet are to or from a site outside the US.**

Multiple Site Correlation

Using Bro we can correlate cross site activity for:

- **Network Connections**
  - **Scanning**
  - **Hostile HTTP activity**
  - **SSH Keystroke Analysis**
- **Syslog data – particularly SSH logins.**
- **In general, any event identifiable by Bro.**

Integrating Different Data Sources

We can now:

- See what applications see.
- See what hosts see.
- Bring together different data sources.
- Addresses issues unique to HPC environment such as Grid tools.

This comes at a cost of complexity.

Looking at SSH

Change from rsh/telnet to SSH removed snooping/man in middle attacks, but user sessions are now opaque to security staff.

...

Increase in number and sophistication of user account thefts requires us to address this issue.

SSH Options

To resolve this we looked at:

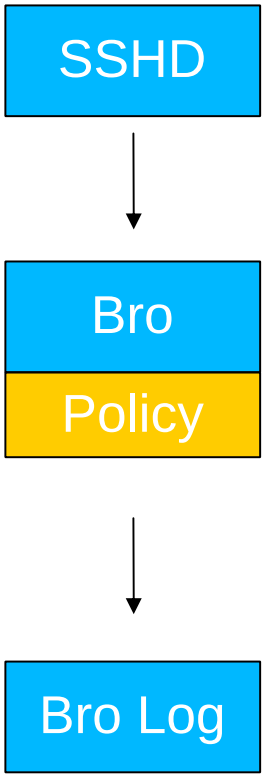
- SSH Jump point: external SSH, internal telnet.
- Key Escrow.
- Combine above options.
- Jump Point + Backdoor SSHD.
- Backdoor shells on individual systems.
- Kernel Backdoor on systems.
- Modify SSHD on individual systems.

Our Solution

Backdoor SSHD and route keystrokes to analysis system.

- **Similar in spirit to the Bro syslog analyzer.**
- **Balances need for security with the desire to limit invasive recording of information.**

SSH Login Analysis



User Types: `user@host> unset HISTFILE`

Bro Policy Filters on:

- Known Hostile Content
- Accumulation of Suspious Activity

If actionable record buffer and remainder of session, else add to temporary buffer.

If no flag to record, flush temporary buffer.

Bro Records:

```
SensitiveLogin 1.2.3.4/33445 -> 5.6.7.8/ssh "unset HISTFILE"
```

Additional Logging

Sample of a recorded session:

```
tx: ls -l
rx: -bash-2.05b$ ls -l
rx: total 11744
rx: drwxr-xr-x  2 scottc  wheel  512 Aug 19 10:51 bin
rx: drwxr-xr-x  2 scottc  wheel  512 Aug  8 17:35 etc
rx: drwxr-xr-x  2 scottc  wheel  512 Aug 18 14:56 lib
tx: exit
rx: -bash-2.05b$ exit
rx: logout
```

SSH Conclusion

Opaque traffic stream and use of ssh sessions as attack channel are forcing the analysis of keystroke data to look for attack signatures. This can be done automatically via an instrumented SSHD which feeds real time data to an IDS.

Section Conclusion

Changing attack methods require new ways of observing behavior. To address this we are implementing:

- **OTP to limit exposure of authentication data.**
- **Site based syslog servers to monitor local machine and application data.**

Section Conclusion 2

- Distributed IDS infrastructure to tie together distinct events across time and systems.

- Instrumented SSHD to access user data stream and channel behavior. Tied results into IDS system.

University of California
San Francisco

Office of Academic and
Administrative Information Systems

Risk Assessments, Mitigations and Compliance

Stephen Lau
University of California, San Francisco
Stephen.Lau@ucsf.edu

November 12, 2006

Agenda

- Risk Management
- Security Models
- Threat Analysis
- Quantifying and Determining Risk
- Performing Risk Assessments
- Conducting Mitigations
- Achieving Ongoing Compliance

Risk Management

- Risk Management is high level institutional planning for security.
  - Determine organization's priorities
  - Identify what to protect and how to protect it.

- Balancing act of risk/productivity/efficiency
  - Open vs. restricted

- It's *all* about shades of grey
  - Policy decisions
  - Cost containment
  - Risk management
  - Business needs

Restricted Open

Where is your site's set point?

Risk Management

- It's difficult in a de-centralized environment!
- New vectors are constantly appearing
  - New software and hardware systems, etc.
  - New initiatives and research partners.
- Threats are constantly changing and evolving.
  - Akin to an ongoing "arms race"

The risk landscape is constantly changing

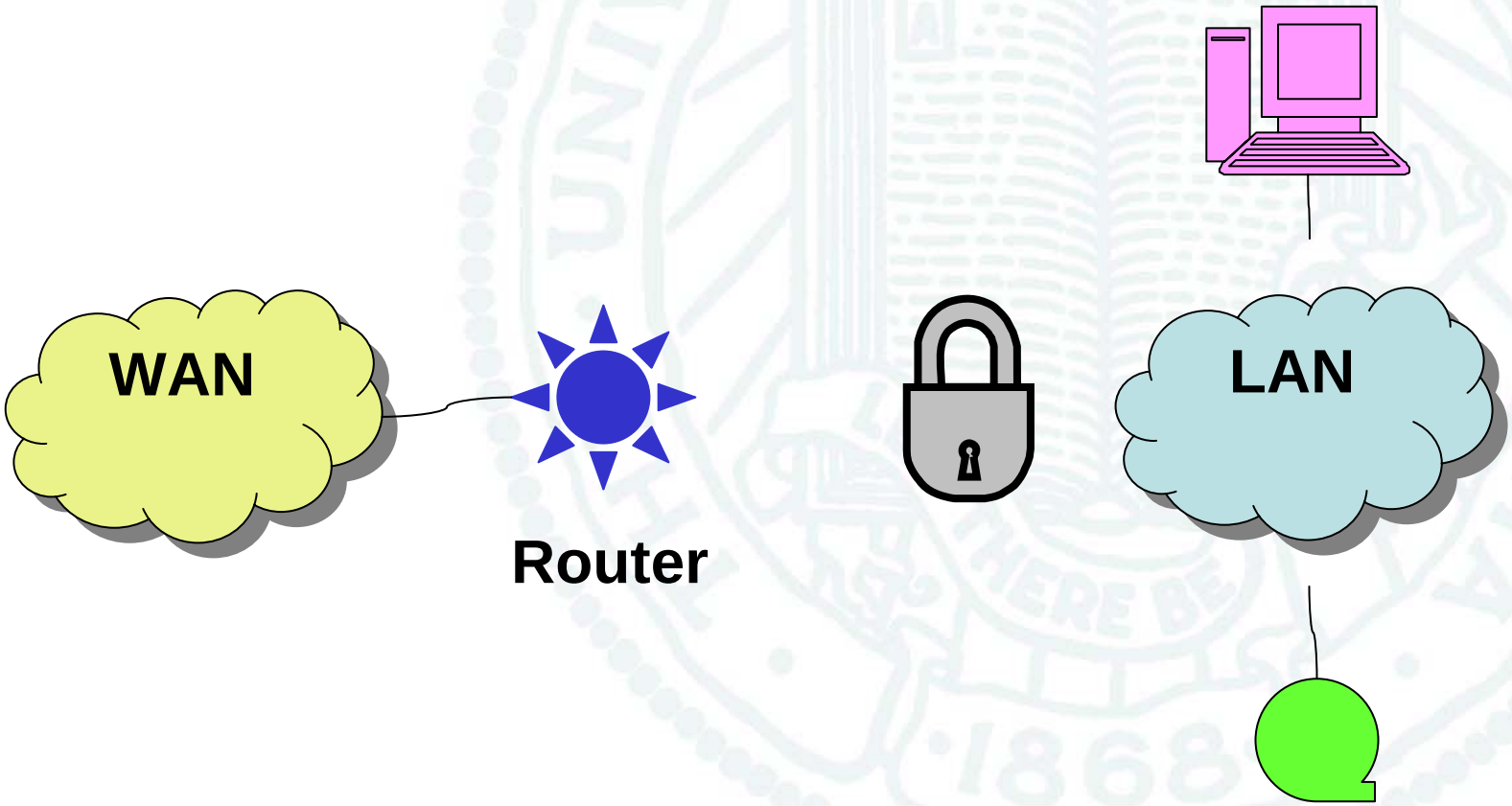
Security Models

- Why develop a model?
  - Efficient allocation of resources
  - ***Effective*** allocation of resources

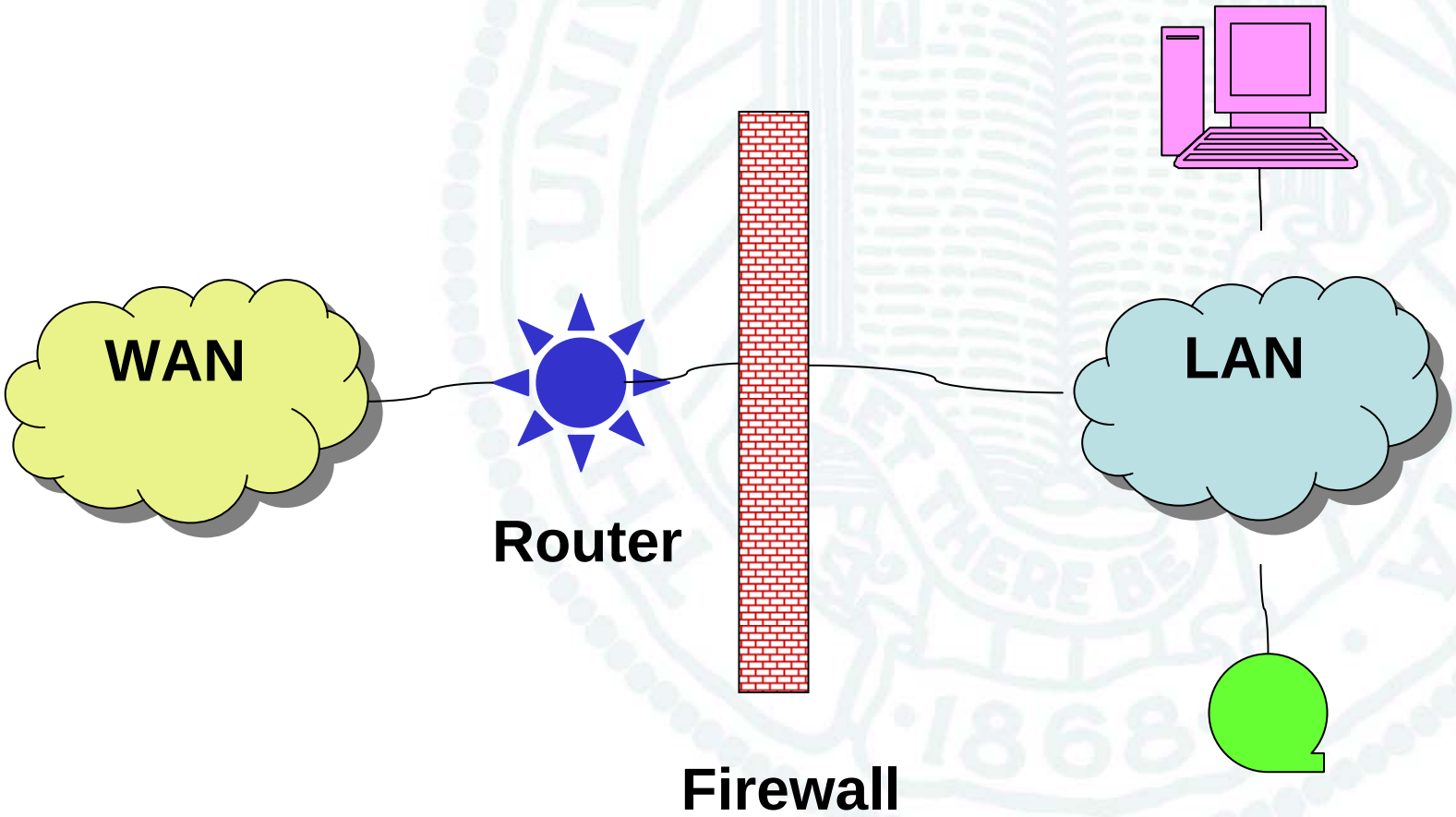
- Determine how tools and techniques will work within your model and where *your* “holes” are, not the other way around.

Vendors typically try to sell you on “their” model or approach.

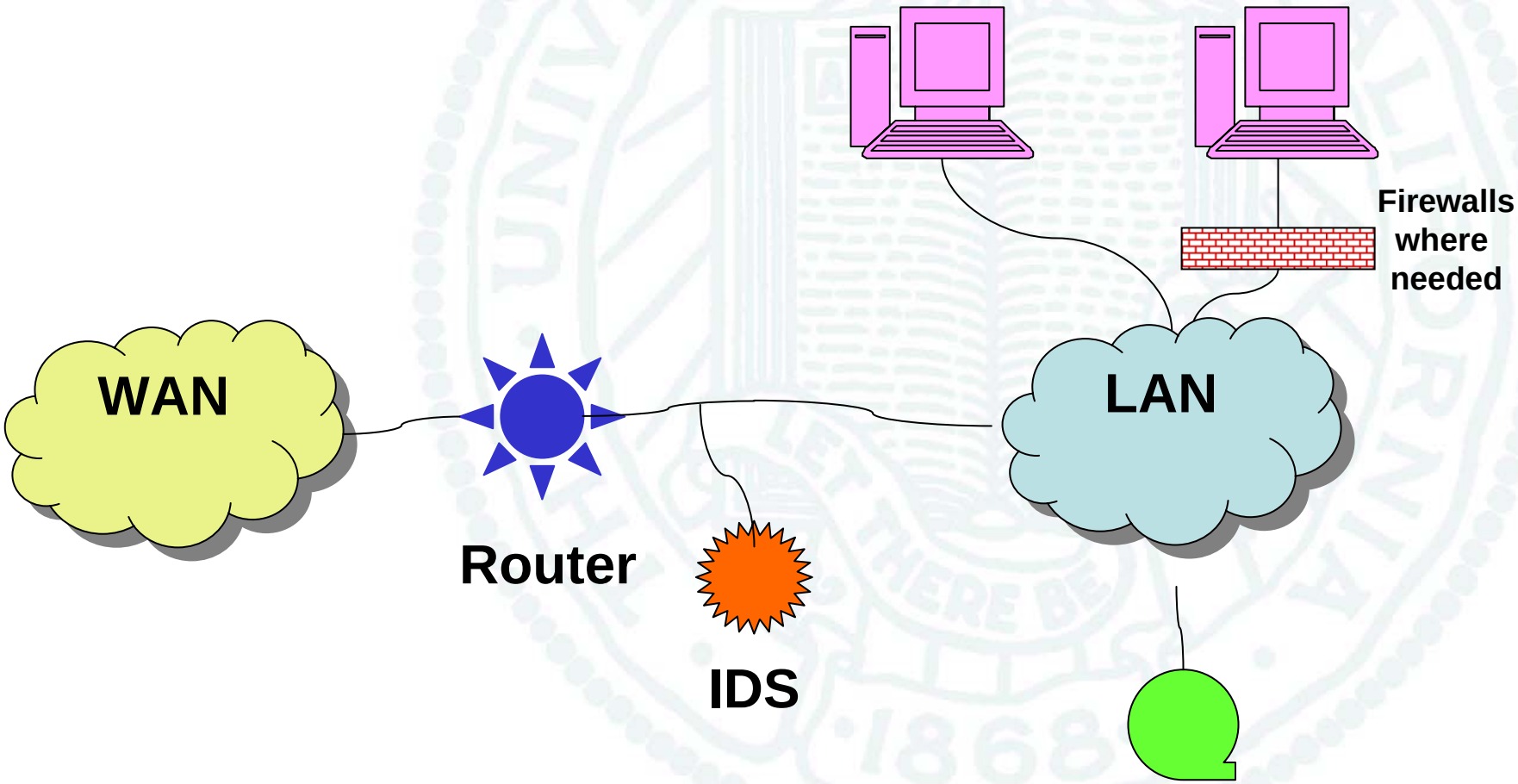
Security Model
Restrictive (Classified/Isolated)



Security Model
Industrial / Enterprise



Security Model
Open Environment



Threat Analysis

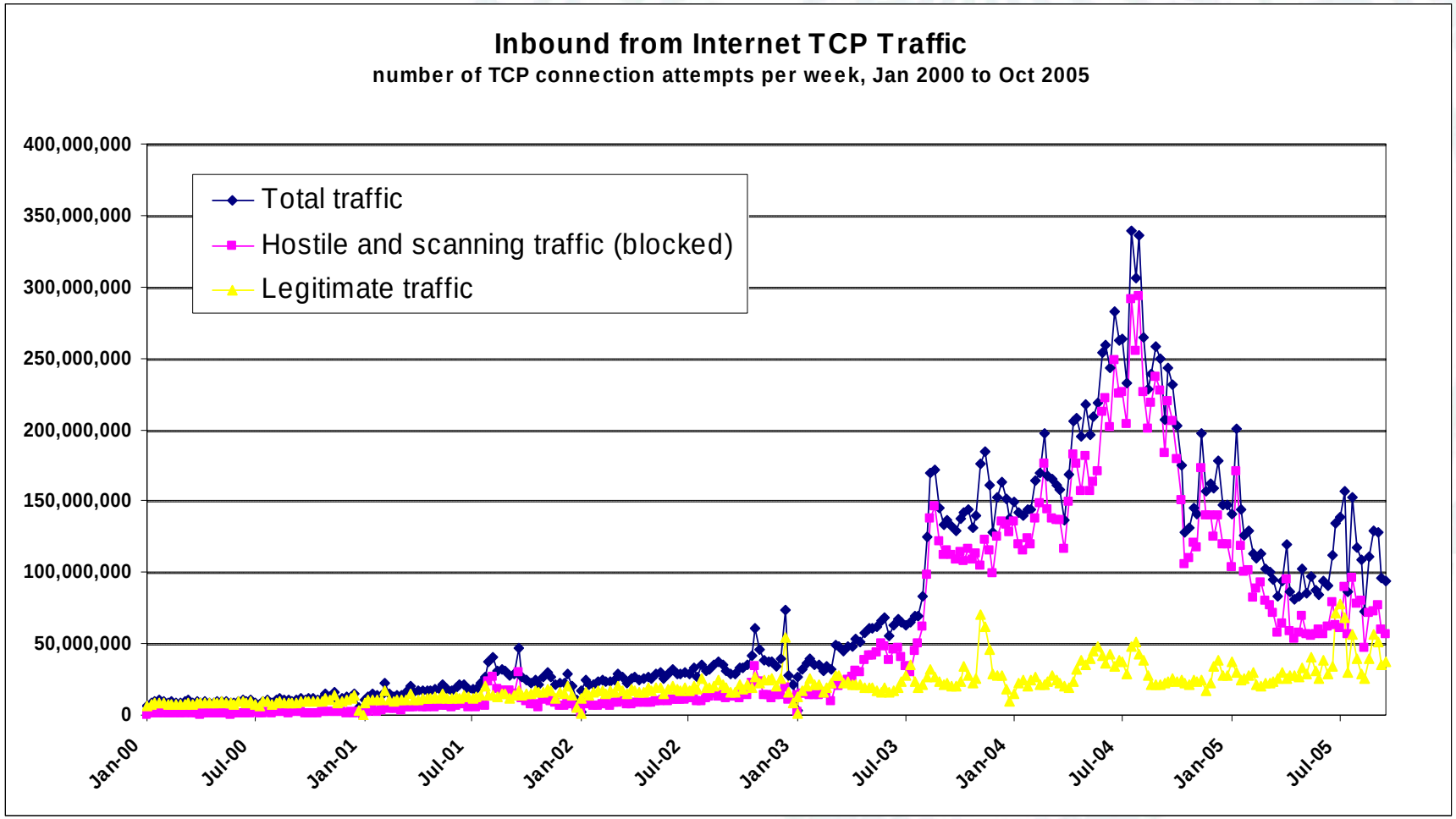
- Threat Analysis identifies all potential threats.
  - Allows a site to prioritize and determine what is feasible.
  - Won't be able to protect against everything!

- What are *your* threats*?*
  - What are you trying to protect?
  - Using what sort of resources?
  - Against what sort of adversary who has what sort of goals & capabilities?

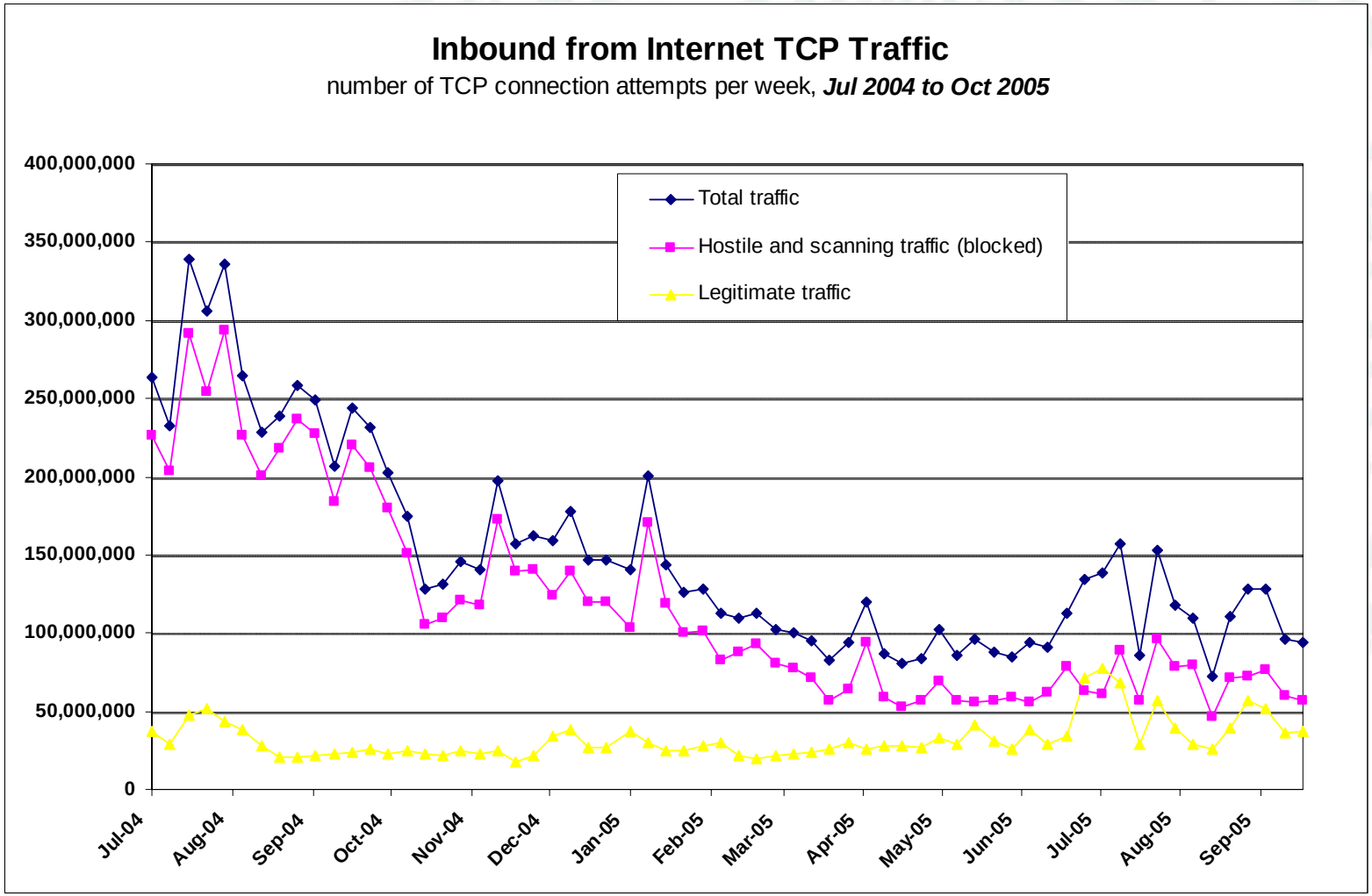
Some Threat Vectors

- Insider Threat
  - Hard to detect ⇒ hard to quantify
  - Can be *really* damaging
- Outsider Threat:
  - Attacks from “outside” (over the Internet) are *ubiquitous*.
- Account compromise
  - Via password-guessing, password cracking, passwords sniffed elsewhere
  - Via trojaned SSH servers
  - Increasingly, interactive traffic is invisible due to encryption
- Users
  - Poor user credential hygiene
    - simple passwords, sharing passwords
  - Phishing schemes
- Hosts
  - Unmanaged or improperly managed systems
  - Embedded / Legacy systems
  - Laptops infected elsewhere and brought on site.
  - Physical loss (stolen laptops)

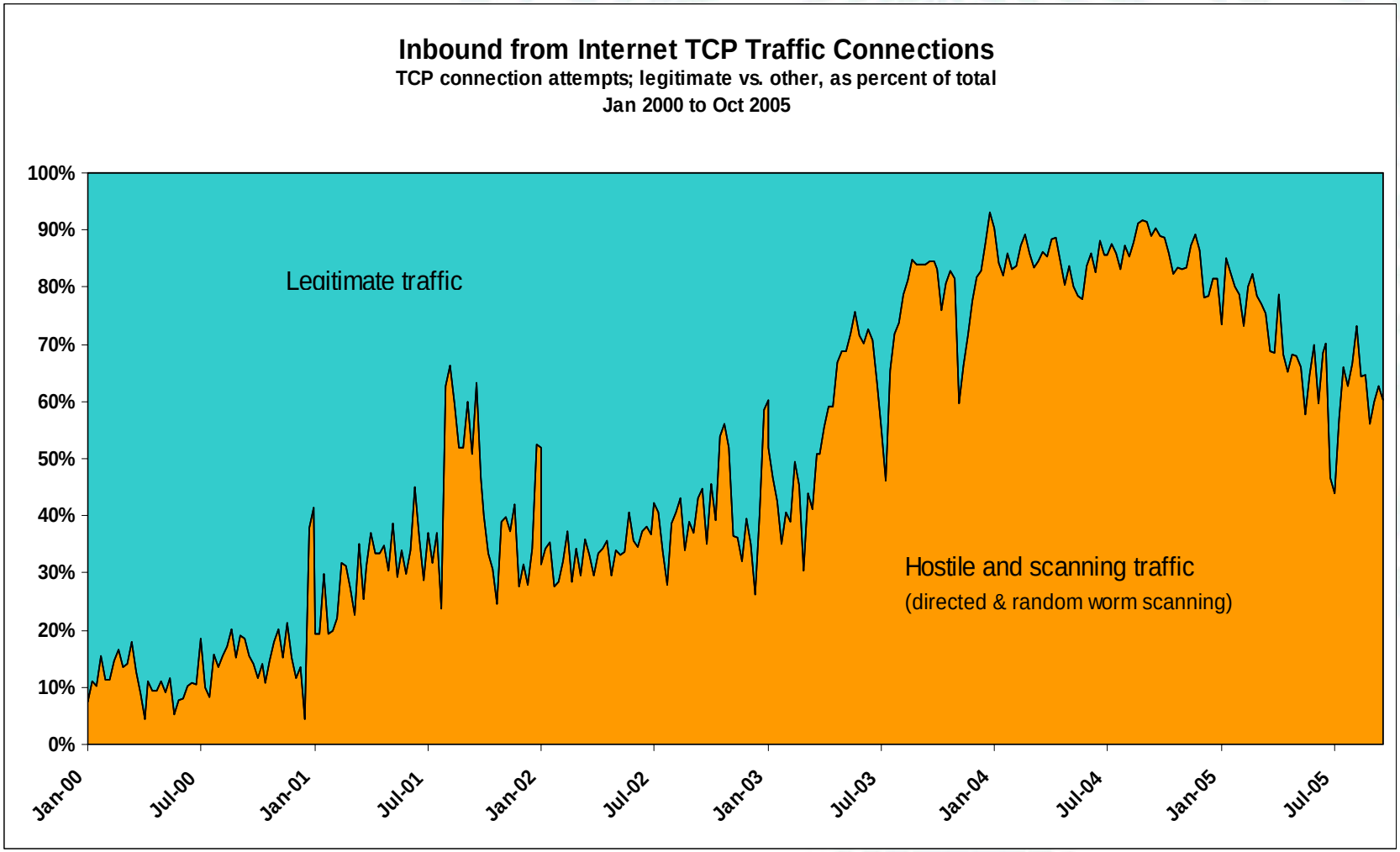
Identifying the Threat



Identifying the Threat



Identifying the Threat



Costs and Impact

- Quantifiable
  - Labor Costs
    - Loss of productivity, down time
    - Incident investigation costs
    - Costs associated with implementing new mitigation strategies
  - Equipment Costs
    - Stolen / Lost equipment
    - Unplanned and necessitated upgrades to mitigate vulnerabilities
    - Resource expenditures to manage malicious traffic
  - Legal / Regulatory costs
    - Costs associated with mandatory notifications
    - Liability issues associated with improper exposure of data such as ePHI or PII.
- Not easily quantifiable (yet still very important)
  - Site embarrassment
    - Reputation damage
  - Potential loss of future business
  - Lost research

Cost Model

- Review previous incidents to estimate damage in dollars.
  - Refine as needed.
- Completely accurate numbers not necessary
  - More interested to see change over time and relative differences.
- Requires best thinking and estimates from those responsible for protecting your site.
- Track “cost” over time to see changes.
  - Use to locate “low hanging fruit”.

Cost Savings

- Every incident *prevented* is savings for you site.
  - Unfortunately, difficult to determine this number!
- One datapoint:
  - Average “honeypot” system on the open Internet will get “compromised” in less than 4 minutes.
    - Honeypot is equivalent to a poorly managed system.
    - In other words, if you have ZERO protection, all your systems would eventually be compromised, sooner vs. later.
      - Worst case scenario.
- How many scans does your firewall/IDS block?
  - What would have been the damage if they didn’t exist?

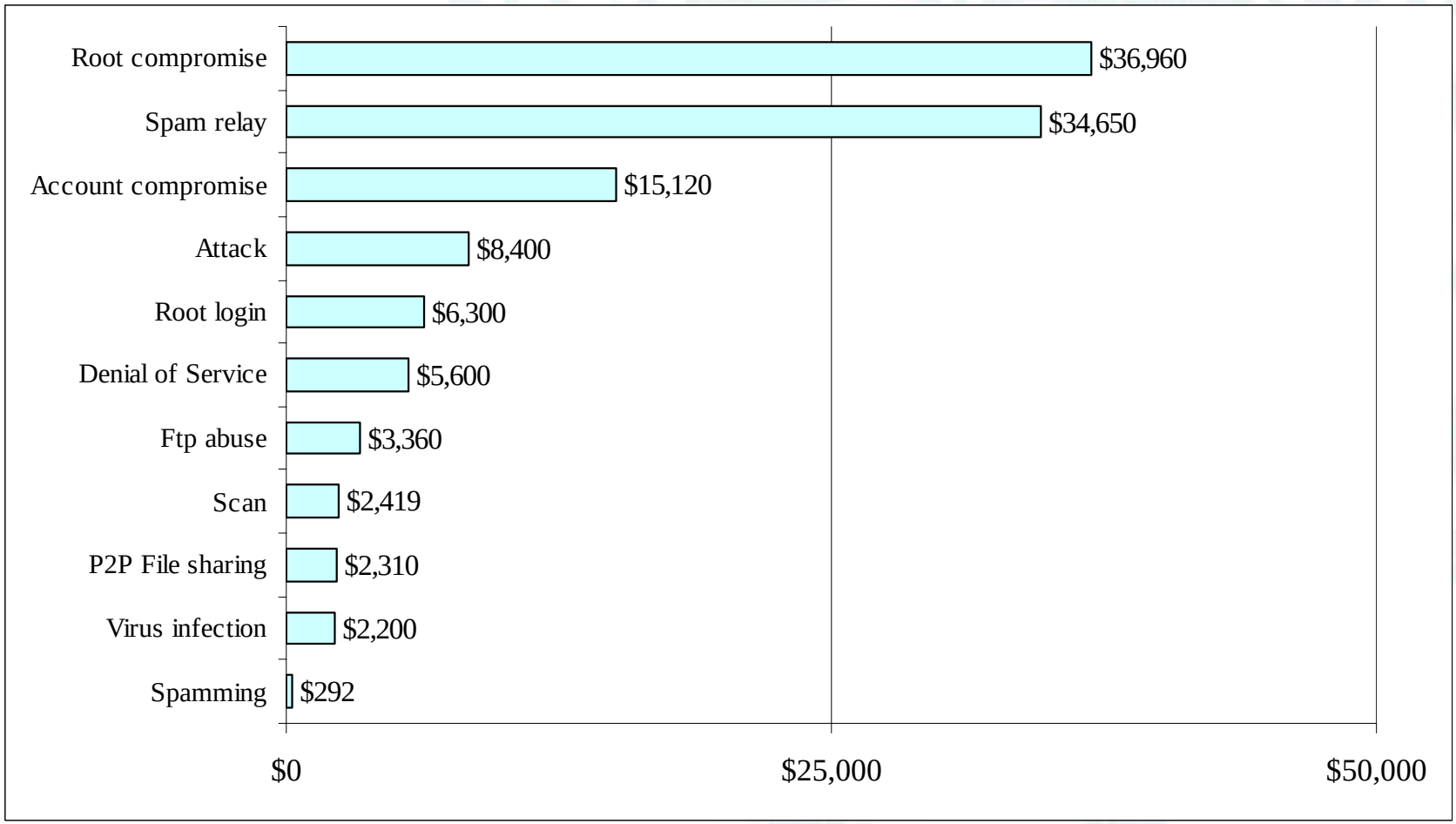
Cost Model

- The next few slides show results of using methodology to:
  - Determine the nominal, probable, and possible damage of different incidents
  - Calculate the damage avoided
  - Evaluate the cost effectiveness of individual protective measures
- Estimates will become more refined over time as meaningful data is collected.
- Your mileage will vary!

Cost Estimates

Incident type	Damage Source					Totals	
Name	*Diagnostic Effort*	*Legal Effort*	*Public Relations Effort*	*Repairs*	*Reporting Effort*	*Total Person Days*	*Nominal Damage Per Hit*
Account compromise	0.75	0.1	0.3	1	0.25	2.4	$1,680
Attack	0.75				0.25	1	$700
Denial of Service	1.75			2	0.25	4	$2,800
File damaged or destroyed		0.01	0.01	1		1.02	$714
Ftp abuse	0.75	0.1	0.3	1	0.25	2.4	$1,680
Inappropriate use	0.75	1	0.3	1	0.25	3.3	$2,310
Root compromise	1.75	0.1	0.3	2	0.25	4.4	$3,080
Root login	0.25			0.5	0.25	1	$700
Scan	0.003125				0.003125	0.00625	$4
Spam relay	0.75		0.3	2	0.25	3.3	$2,310
Spamming	0.000347			0.000347		0.000694	$0.80
Virus/Worm				0.125		0.125	$88

Cost Estimate Over 1 Fiscal Year



Calculating Probable Damage

- ***Probable Damage*** includes a factor for ***non routine*** incidents
  - Accept a cap that non-routine incidents do not exceed nominal damage by more than a factor of 1000
  - Calculated using probability of incurring costs of ten, one hundred, and one thousand times nominal damage.
  - Essentially a scale factor on Nominal Damage

Probable Damage Estimate

Name	Nominal Damage Per Hit	P10	P100	P1000	Probable damage per hit
Account compromise	$1,680	0.05	0.01	0.001	$5,778
Attack	$700	0.05	0.01	0.001	$2,407
Denial of Service	$2,800	0.05	0.01	0.001	$9,629
File damaged or destroyed	$714	0.05	0.01	0.001	$2,455
Ftp abuse	$1,680	0.05	0.01	0.001	$5,778
Inappropriate Use	$2,310	0.05	0.01	0.001	$7,944
Root compromise	$3,080	0.05	0.01	0.001	$10,592
Root login	$700	0.05	0.01	0.001	$2,407
Scan	$4	0.05	0.01	0.001	$15
Spam relay	$2,310	0.05	0.01	0.001	$7,944
Spamming	$0	0.05	0.01	0.001	$2
Virus/Worm	$88	0.3	0.05	0	$757

Possible, Probable and Avoided Damage

Incident Type	Probable damage per hit	Total Unblocked Attacks	Total Blocked Attacks	Probable damage per year	Damage avoided per year	possible damage per year
Account Compromise	$5,778	9	1490	$51,000	$8,607,000	$8,658,000
Attack	$2,407	12	633	$28,000	$1,524,000	$1,552,000
DOS	$9,629	2	283	$19,000	$2,724,000	$2,743,000
Ftp Abuse	$5,778	2	327	$11,000	$1,888,000	$1,899,000
Inappropriate Use	$7,944	1	70	$7,000	$552,000	$559,000
Root Compromise	$10,592	12	1987	$127,000	$21,041,000	$21,168,000
Root Login	$2,407	9	540	$21,000	$1,301,000	$1,322,000
Scan	$15	553	28078	$8,000	$422,000	$430,000
Spam Relay	$7,944	15	2058	$119,000	$16,351,000	$16,470,000
Total				*$391,000*	*$54,410,000*	*$54,801,000*

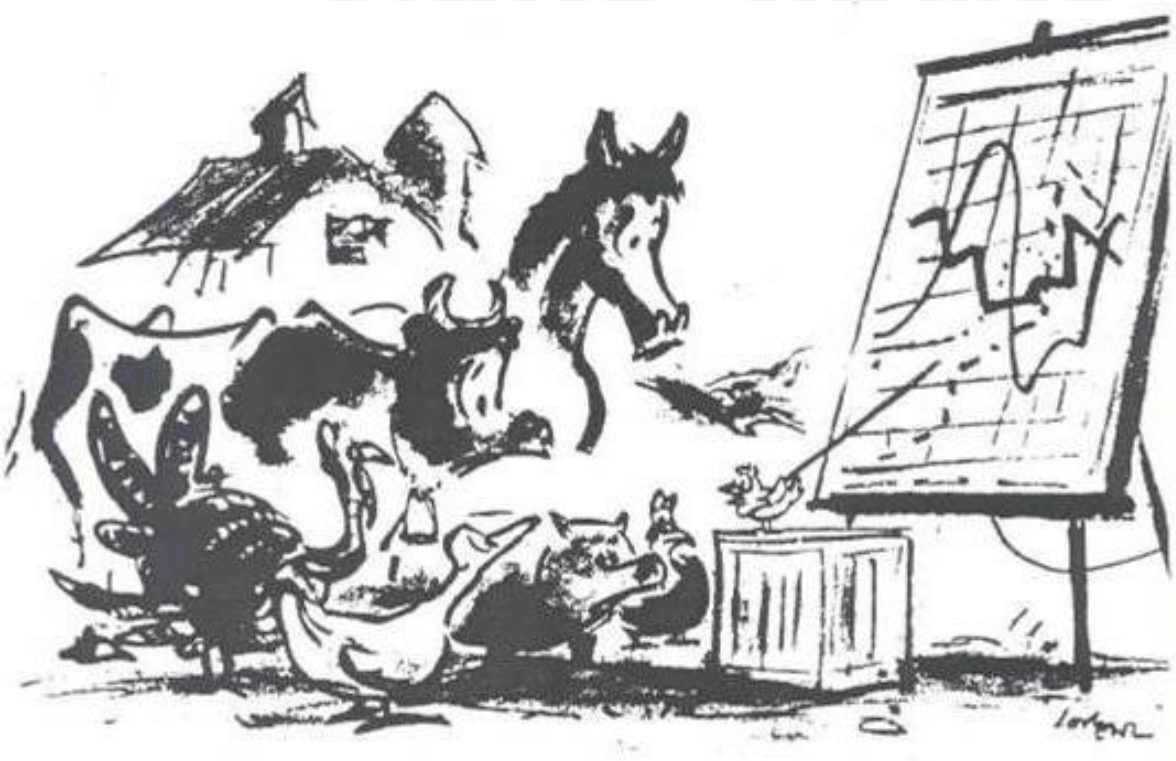
Estimating Security Measures Effectiveness

CounterMeasure	Account compromise	Attack	Denial of Service	Ftp abuse	P2P File Sharing	Root compromise	Root login	Scan	Spam relay
Warning banner program					2%				
Regular password cracking	10%			10%		10%			
Firewall program	19%	19%	19%	19%	19%	19%	19%	19%	19%
Router control lists program	19%	19%	19%	19%	19%	19%		19%	19%
Network Connection Control.	18%		18%	18%	18%	18%	18%		18%
Vulnerability scanning	50%		50%	50%		50%			50%
Bro Intrusion detection sensors and analysis infrastructure	95%	95%	95%	95%	95%	95%	95%	95%	95%
Dial in service security program	4%	4%		4%		4%	4%		
New employee orientation; System Administrator training.	9%		9%	9%	9%	9%	9%		9%
VPN infrastructure	1%					1%	1%		
Web server security requirements	3%		3%	3%		3%	3%		
System audit program for local hosts	4%		4%	4%		4%			4%

Risk Avoided and Return on Investment

Countermeasure	*Operating Cost*	*Risk Avoided*	*ROI*
Bro Intrusion detection sensors and analysis infrastructure	$140,000	$7,522,015	5273%
Vulnerability scanning program	$35,000	$332,359	850%
Firewall program	$7,000	$92,864	1227%
Router control lists program	$7,000	$87,495	1150%
Network Connection Control.	$4,200	$79,725	1798%
New employee orientation; System Administrator training.	$4,200	$35,908	755%
Regular password cracking program	$5,000	$21,274	325%
Dial in service security program	$46,200	$9,528	-79%
Web server security requirements	$15,400	$7,176	-53%

Don't Overanalyze



"And so, extrapolating from the best figures available, we see that current trends, unless dramatically reversed, will inevitably lead to a situation in which the sky will fall."

Risk Assessments

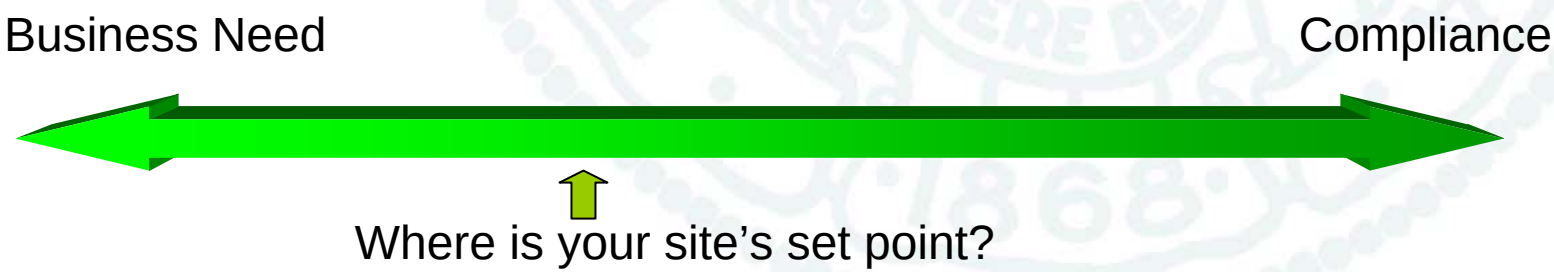
- Utilize the following
  - Identified Threat Vectors
  - Cost Model
  - Security Model
- Identify similar systems or areas on your site.
  - Enclave Model
  - i.e. administrative systems, restricted systems, etc.
- Perform assessments on key aspects of your site.
  - Identify key areas that are vulnerable.
- Different enclaves may have different requirements
  - Threat vectors
  - Cost model
  - Security model
- Attempt to collect information passively
- Assessments need to be done on a continuous basis.

Performing Risk Assessments

- Utilize scanning tools
  - Identify services enabled
  - Identify vulnerabilities
- Interviews and Checklists
  - Potentially difficult information to collect.
  - Focus on information that's not passively retrievable
    - i.e. identify any PII/ePHI data
  - Don't bury your staff in paperwork!
- Federal Guidelines from NIST
  - NIST 800-26 – Self Assessments
  - NIST 800-53 – Recommended Security Controls

Addressing Risk

- Devise methodology to address identified risks
  - Pull out your security techniques toolbag.
- Balance risk vs. cost vs. business needs
- In some cases, you may want to accept the risk.
  - Make sure decision process is well documented!



Compliance

- Ongoing adherence to site policies, laws and regulations.
- Tools that can assist
  - System logging
  - Risk assessments
  - Vulnerability detection
  - Intrusion detection
  - Cost model
- Ideally, this information is stored in a database.
  - Allows for automated compliance reporting.
- Periodic risk assessments, vulnerability scans to ensure compliance.
- Site needs to define what it means to be “in compliance”.

Summary

- Risk management is useful for institutional planning for security.
  - Balancing act between business need and acceptable risk.
- Identify the threats.
- Develop a security model.
- Determine costs associated with risks.
- Perform Risk Assessments
- Mitigate identified risks.
  - Including accepting risks.
- Achieve compliance through periodic re-assessments.
- Don't worry, be happy. ☺

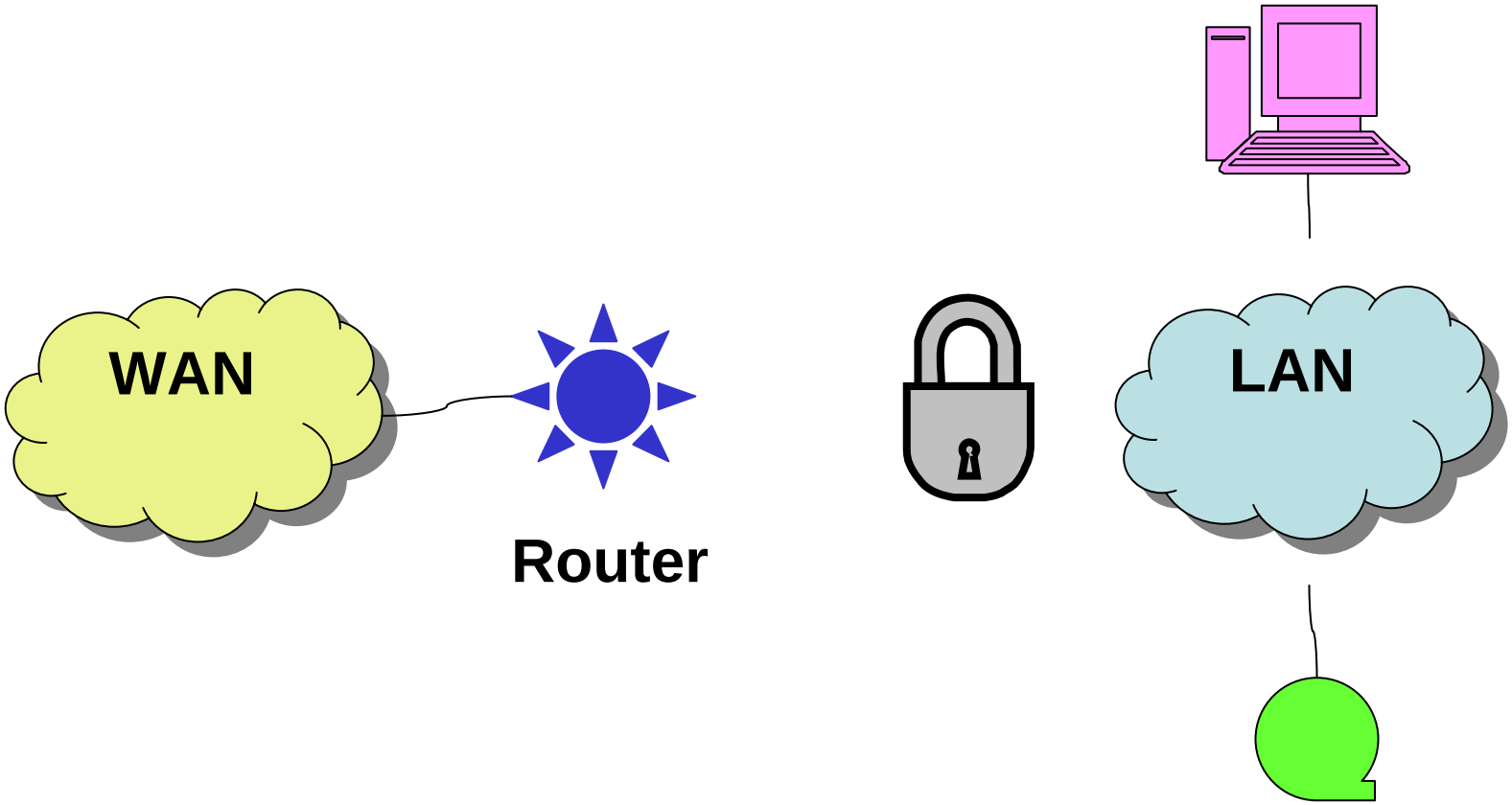
Components of Good Protection

William T.C. Kramer
Lawrence Berkeley National Laboratory
NERSC
wtkramer@lbl.gov

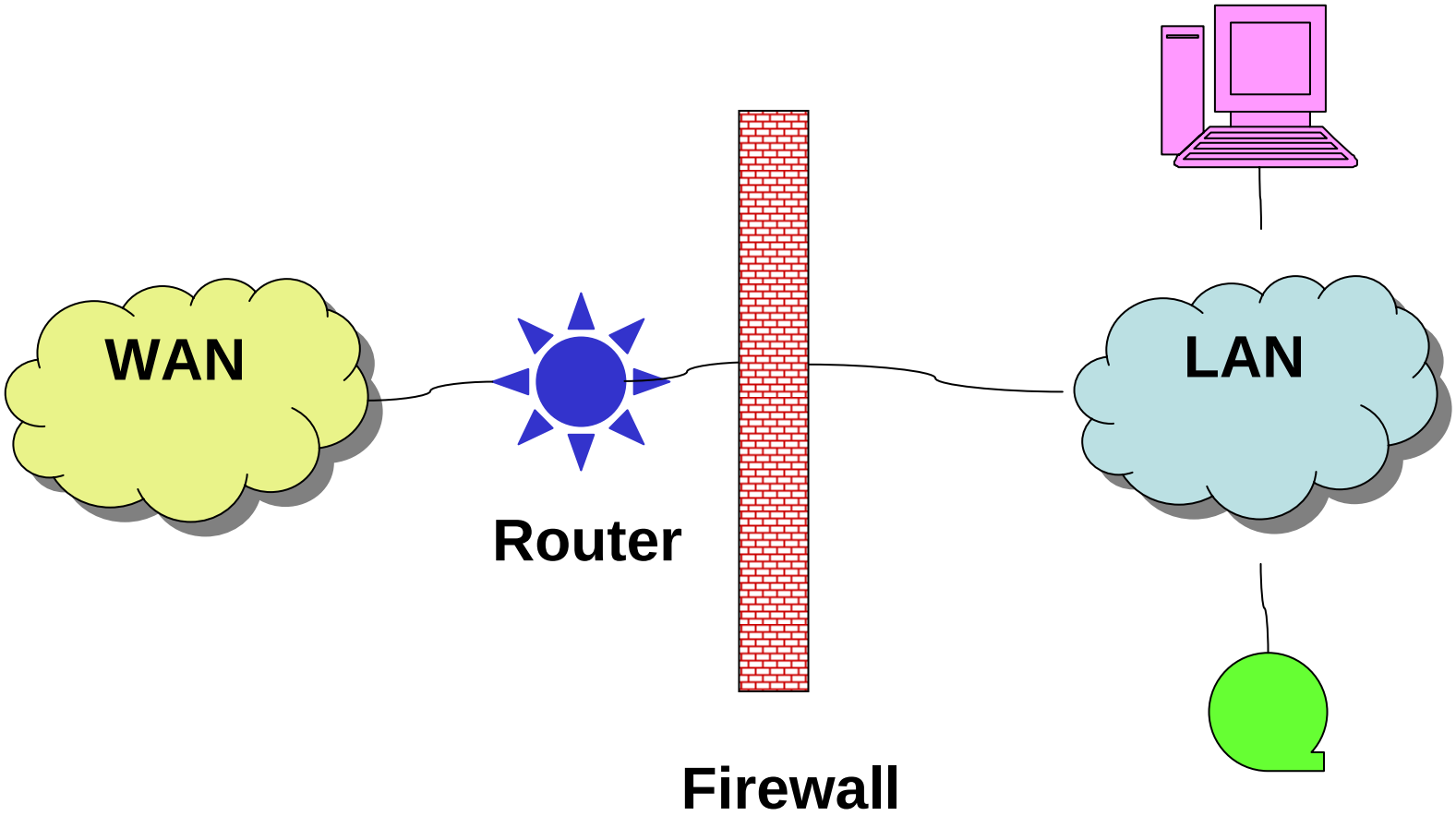
Outline

- Components of good protection
  - Defining terms
  - Goals and objectives for an open site
  - Security implications of an HPC environment
  - Policy and Procedures
  - Good systems protection
  - Response teams
  - Other “best practices”

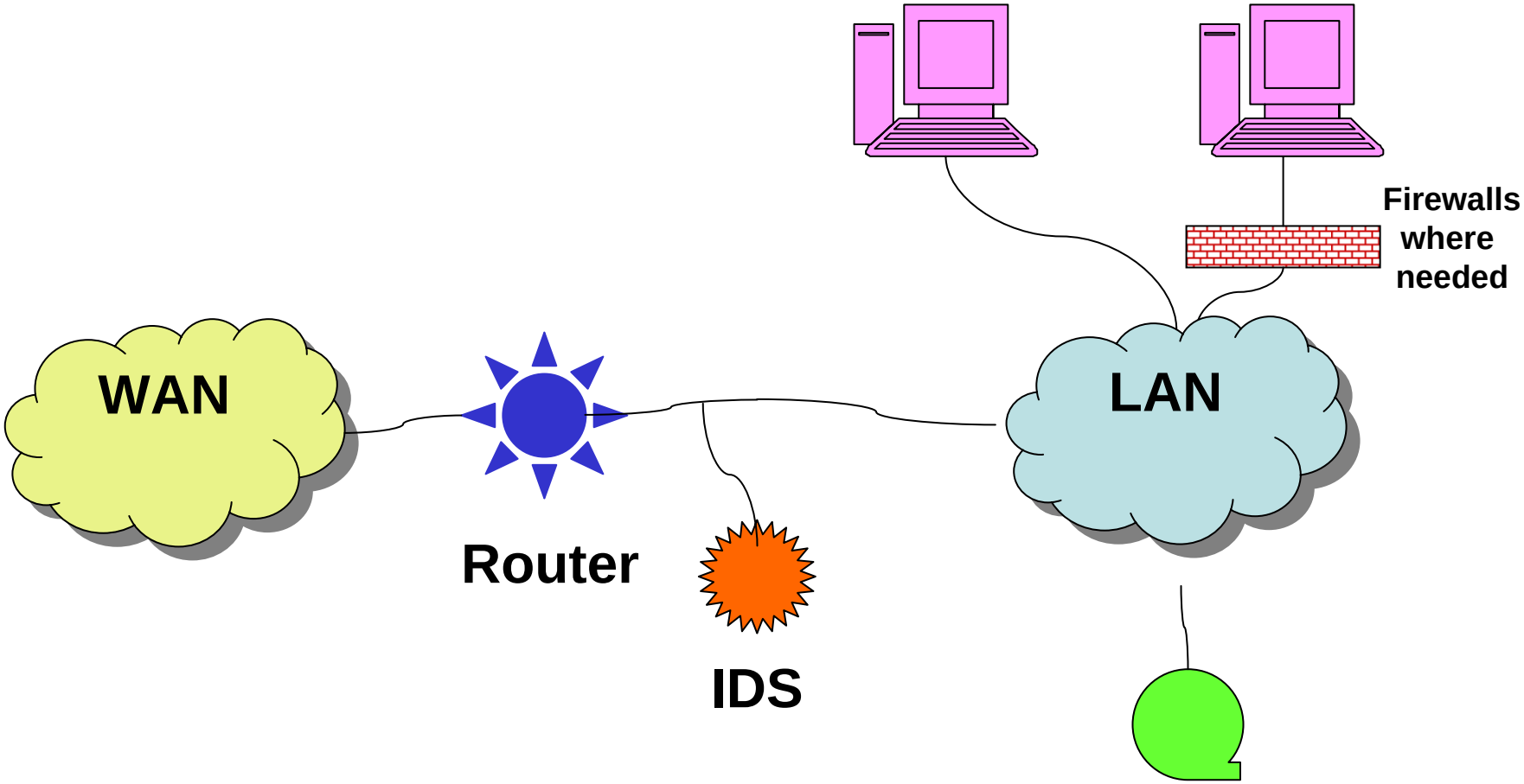
Restrictive (Classified/Isolated)



Industrial / Enterprise



Open Environment



Open Science Environment

- **Unlike enterprise institutions**
  - **Enterprise oriented computer security techniques fail**
- **Varied and atypical computational infrastructure**
- **High bandwidth / performance applications**
  - **Unique applications with unique requirements and traffic patterns**
- **Varied and distributed resources**
- **Multi-institutional collaborations across all levels**
  - **e.g. LBNL has approximately 4000 collaborations/year**

Characteristic Environment

- **Varied Systems**
  - Workstations
    - Laptops, PDAs, workstations, cell phones, etc…
    - Windows, Unix, Linux, OSX, etc...
  - Servers
    - Web servers, mail, LDAP, etc.
  - High Performance Platforms
    - High End Cluster systems
    - Mass storage systems
    - Dedicated Systems
      - i.e. Visualization, Mathematical

- **Network**
  - High speed network connections (possibly multiple)
    - e.g. 10G at NERSC
  - Wireless
  - Modem pools

Characteristic Environment

- Additional systems
  - Printers
  - Fax machines
  - Infrastructure / Embedded systems
    - Door access control, environmental controls
    - Don’t forget about these!

- Users
  - Diverse user community scattered around the globe
    - Mix of science, industry, academic
  - Multiple large scale, multi-site collaborations

- Staff
  - Spread out between multiple locations
  - Highly mobile
  - Want access from home systems

Network Traffic Patterns

- Open scientific facilities traffic patterns differ from industrial/enterprise
- Typical enterprise traffic
  - Web, email, dedicated/known services
- Typical open facility
  - Varies over time
  - Unique protocols
  - Large volumes of traffic

Example of Network Traffic

Type of Traffic	Number of Connections	Overall Percentage of Traffic
Bulk Data Transfer	666,529	83.73%
Grid Services	74,178	7.19%
Web Related	288,3754	5.30%
Database	620,1730	.27%
Mail	200,484	.04%
System Services	185,272	.04%
Interactive	116	<.1%
Total	10,212,063	96.57%

Policy and Procedures

- Cyber Protection Policy controls computer access and usage
  - How information and processing power can be accessed, manipulated and shared
  - Represents external laws and regulations, organizational mission, goals and business practices
  - Mandatory and Discretionary policies and rules
- Policy should be broad and change infrequently
  - Guided by the site mission and philosophy
  - Policy change typically involves a formalized change and review process
- Business practices should define details of policy implementation
  - Specific as needed and change as needed
  - Changes are technical, not bureaucratic
  - Should not implement its own policy
  - If business practice violates policy, then it is time to change one or the other explicitly

Policy and Procedures (cont)

- **Mandatory – Enforce access control rules that constrains access to information and/or resources based on authorization**
  - **i.e.To root or not to root, that is the question**

- **Discretionary – An individual may specify the types of access others may have**
  - **i.e. File permissions**

Defense in Depth

- Use of multiple tools and techniques leverages off strengths and weaknesses
  - Multiple sensors to detect and prevent intrusions
  - No single points of failure
- No single tool or technique guarantees a problem free environment
- Protects against the “hard outer shell, soft inside” vulnerability
  - Caveat: More resource intensive to implement and maintain, integration difficulties

Defense in Depth Layers

- **External Perimeter Defense**
  - **All points of entry into the network, the "DMZ"**
- **Internal Network Protection**
- **Host Level Protection**
- **User / Staff Protection**
  - **Education**
- **Physical Security**

External Perimeter Defense

- Determine all perimeters
  - Wireless, modems
- Intrusion Detection System
  - i.e. Bro, Snort
- Host shunning
  - Tied into perimeter defense to react to attacks
- Router filtering
  - Block archaic or unused services
    - i.e. echo, chargen
- Email Virus Filtering
  - Filters all inbound / outbound email

Internal Network Protection

- **Firewalls where appropriate**
  - Non-high performance platforms
  - Dedicated platforms
  - Developmental/Experimental systems
- **Subnet traffic filtering**
  - Further restrict traffic based on subnets
- **Network Segregation / Isolation**
  - Isolate “like” systems together
  - i.e. staff workstations shouldn’t be on same network as HPC systems

Host Level Protection

- Disable unused services upon install
- Anti-virus software
  - Available to all staff, installed by default
- Host Scanning / Vulnerability Eradication
  - Avoid “information overload”
  - Nmap, nessus
- Disable clear text passwords
- Disallow unauthenticated access
- Enable process accounting / logging
  - Provides audit trail

Defense in Depth Layers

- User / Staff Protection
  - Increase staff awareness of computer protection issues
    - Periodic in-house training for staff
    - Periodic Web/Video based training for offsite users
  - All staff / users must annually sign “Usage Agreement”
  - Periodic emails reminding staff / users about key security issues

- Physical Security
  - Restrict physical access to critical systems
  - Educate staff members
  - Provide lockdowns for staff member laptops and systems

Most Common Security Incidents

- **Sniffed passwords**
  - Someone gets a hold of a user password
  - Externally compromised system
  - Exposure via unencrypted means
- **Unpatched systems**
  - New systems (not yet patched)
  - Toolkits used to exploit known vulnerabilities
  - Visitors and staff unknowingly bring in vulnerable or pre-hacked systems
- **Viruses and Worms**
  - Home systems infected, dial in
  - Visitors bring in infected systems
  - Staff members bring systems to conferences, etc.

Good Systems Protection

- Good, clear, consistent policy
- Good business practices that are consistent with policy
- A hierarchy of protection tools and mechanisms from the border to the internals of the system
- Organized ways of discussing and addressing protection issues

Update, Update, Update

- The vast majority of compromises are know exploits for which the known corrections have been available for some time.

- Solution is keeping the systems up to date
  - Patches, New OS releases, etc.
  - For all components
  - More important with open source

- Many reasons not to
  - Staff Effort, User resistance, testing, worry about introducing bugs…

It is the single most important component of system protection

Proving Good Protection is hard

Need to have positive metrics not just negative ones

- **Examples of positive metrics**
  - **Successful accomplishment of the organization's mission**
  - **Number of proactively detected incidents**
    - **You found them first**
  - **Number of sites informed of a problem**
  - **Dollar cost of damage AVOIDED due to protection efforts**
  - **Number scans performed (without finding things)**
  - **Days since last incident**
  - **Training events**
  - **External interactions**
    - **If your peers think you are good then you probably are**

Good Systems Protection (cont)

- Examples of negative metrics
  - Number of reactively detected incidents – "breakins"
    - Someone else found them first and told you
  - Amount of lost time due to incidents
  - Number of restricted services
  - PR of such things

Most organizations typically judge negative more than positive

Configuration Control

- Providing Open Access does not mean loss of control
  - Example
    - IBM SP was delivered with 65,536 open ports
    - After a lot of investigation, it was determined 31 were needed for the system to run
  - Can use limited ranges for services.
    - A set of 1,000 ports provided for Grid Services
    - A specific set of ports for FTP
- Account Management
  - Including regular disabling and removal

Keys and Certification

- Keys
  - Passwords, PKI, One Time Passwords, SSH
- Make it possible for users to store key information at your site
  - Rather then storing it on their local system.
  - User systems are typically vulnerable and not well protected
    - More easily compromised
  - E.g Myproxy
- You can protect the information better than they can

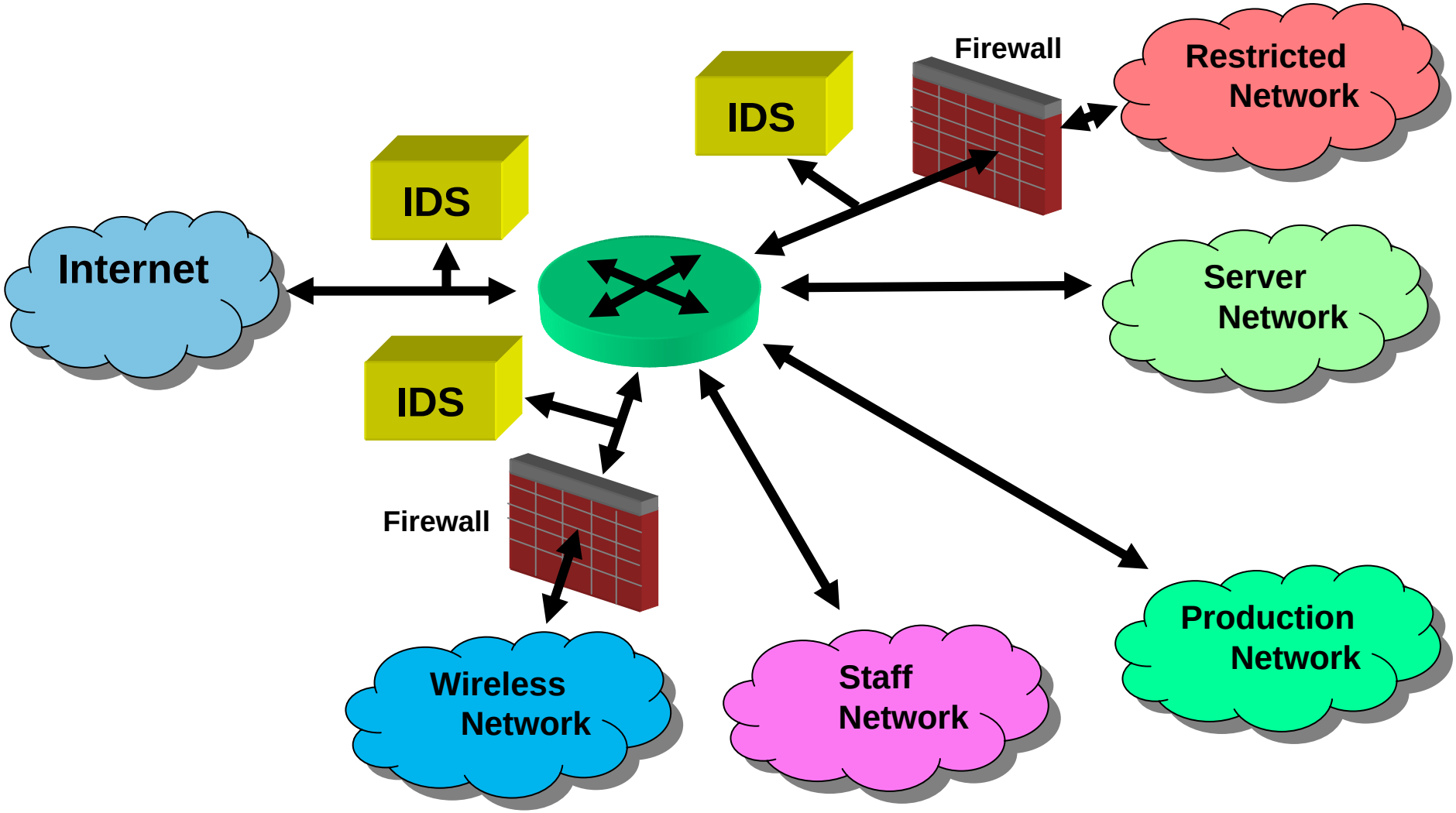
Network Segmentation

- Segment network by service and function
- Investing in different networks for different functions is worthwhile

- Public and Private Networks
  - Clusters should be built out of private networks
  - There should be a few, well defined and configured access points in the cluster for public networks
  - But never assume your private network is really private

- User and Administrative networks

Sample Network



Logging, Monitoring, Scanning

- **Logging is extremely important**
  - Good Practice
  - Forensics
  - Allows analysis for capacity and workload
- **Monitoring**
  - Does not help to log everything if it is not looked at until it is too late
  - Examples – job flow, network attempts, logins, etc.
- **Scanning**
  - Helps assure configuration management
  - Prevents mistakes that lead to vulnerabilities

Logging and Monitoring

- Need to log all activities
  - Process accounting
  - Batch system processing
  - Logins
  - Network connections
  - Violation logging
- Transfer logs to another system on a regular and timely basis
  - Protects against modifications
  - Backup
  - Post Processing

Enlist the users

- Protection has to facilitate the user doing their work – not inhibit it
- Make users aware and responsible
  - Proactively acknowledge a clear appropriate use policy
  - Delegate responsibility to users for certain things they actually can control
    - Some things they have to do such as deciding what data is sensitive
  - Include users into the evolutionary process of protection changes
- Does not work if protection is always getting in the way of the users
  - They will go around to get their work done

Enlist the users

- **Have them report "suspicious activity"**
  - **Strange files or directories**
  - **Unusual login times**
  - **Unverified phone call from "NERSC" asking for passwords or account information**
- **Have them report external incidents**
  - **Please report any incidents at sites that you use to access NERSC**
- **Report incidents where they suspect credentials are sniffed or stolen**

Our users have been critical to identifying and resolving many incidents!

Staff and Organization

- Excellent people with enough time to spend on protection
- Integrate security staff with system and network management groups
  - At NERSC, network and security groups are integrated into one group.
- Integrate security practices into all staff functions
- Integrated security team in addition to dedicated security staff
  - Channels communications to various groups
  - Acts are a response team
  - Ensures representation and buy in for policy development
  - Serves as a broad technical resource

Avoid "us" vs. "them" mentality

Summary

- A site needs good, and consistent policy and business practices
- A hierarchy of protection tools and mechanisms from the border to the internals of the system
- Update always
- Keys, passwords and certs should not be stored on user systems
- Well defined network architectures
- Logging and monitoring of systems is key

Question and Answer

Any Questions?

Please complete feedback forms.

Contact Information

- Stephen Lau
  - University of California, San Francisco
  - Email: Stephen.Lau@ucsf.edu
  - Phone: +1 (415) 476-3106
  - PGP Key Fingerprint:
    - 44C8 C9CB C15E 2AE1 7B0A 544E 9A04 AB2B F63F 748B

- William T. C. Kramer
  - NERSC/LBNL
  - Email: wtkramer@lbl.gov

- Scott Campbell
  - NERSC/LBNL
  - Email: scampbell@lbl.gov
  - Phone: +1 (510) 486-6986
  - After Hours: +1 (510) 486-8600
  - NERSC Security Contact: security@nersc.gov
  - PGP Key Fingerprint:
    - 89F9 593E 2381 88B2 B270 78B2 2B63 E5AB C07B 641C

- Brian Tierney
  - LBNL
  - Email: bltierney@lbl.gov
  - Phone: +1 (510) 486-7381
  - PGP Key Fingerprint:
    - BC53 77DB 2437 0B83 0C8A 8796 DA36 68F4 3CF5 0958

Bibliography

- S. Dharmapurikar and V. Paxson, *Robust TCP Stream Reassembly in the Presence of Adversaries*, Proc. USENIX Security Symposium 2005.

- H. Dreger, A. Feldmann, V. Paxson, and R. Sommer, Operational Experiences with High-Volume Network Intrusion Detection, Proc. ACM CCS, October 2004.
  - **`http://www.icir.org/vern/papers/high-volume.ccs04.pdf`**

- H. Dreger, C. Kreibich, V. Paxson and R. Sommer, *Enhancing the Accuracy of Network-based Intrusion Detection with Host-based Context*, Proc. Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA) 2005.

- H. Dreger, A. Feldmann, M. Mai, V. Paxson and R. Sommer, *Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection*, Proc. USENIX Security Symposium, August 2006.

- J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, Fast Portscan Detection Using Sequential Hypothesis Testing, Proc. IEEE Symposium on Security and Privacy, May 2004.
  - **`http://www.icir.org/vern/papers/portscan-oak04.pdf`**

Bibliography

- R. Kemmerer and G. Vigna, “Intrusion Detection: A Brief History and Overview,” *IEEE Security & Privacy*, April 2002.
  - `http://www.computer.org/security/supplement1/kem/index.htm`
- S. Kornexl, V. Paxson, H. Dreger, A. Feldmann and R. Sommer, *Building a Time Machine for Efficient Recording and Retrieval of High-Volume Network Traffic*, Proc. ACM IMC, October 2005.
- C. Kreibich and R. Sommer, *Policy-controlled Event Management for Distributed Intrusion Detection*, 4th International Workshop on Distributed Event-Based Systems (DEBS'05), 2005, Columbus/Ohio, USA.
- S. McCanne and V. Jacobson, “The BSD Packet Filter: A New Architecture for User-level Packet Capture,” *Proc. 1993 Winter USENIX Conference*, San Diego, CA.
- B. Mukherjee, L. Heberlein, and K. Levitt, “Network Intrusion Detection,” *IEEE Network*, 8(3), pp. 26-41, May/Jun. 1994.

Bibliography (cont.)

- R. Pang, V. Yegneswaran, P. Barford, V. Paxson and L. Peterson, Characteristics of Internet Background Radiation, Proc. ACM IMC, October 2004.
  - **`http://www.icir.org/vern/papers/radiation-imc04.pdf`**

- V. Paxson, “Bro: A System for Detecting Network Intruders in Real-Time,” *Computer Networks*, 31(23-24), pp. 2435–2463, 14 Dec. 1999. (Also *Proc. 7th USENIX Security Symposium*, Jan. 1998.)
  - **`ftp://ftp.ee.lbl.gov/papers/bro-CN99.ps.gz`**

- T. Ptacek and T. Newsham, “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection,” 1998. (unpublished but widely influential)
  - **`http://www.icir.org/vern/Ptacek-Newsham-Evasion-98.ps`**

- M. Ranum et al, “Implementing a generalized tool for network monitoring,” *Proc. LISA ’97*, USENIX 11th Systems Administration Conference, San Diego, Oct. 1997.

Bibliography (cont.)

- R. Sommer and V. Paxson, *Exploiting Independent State For Network Intrusion Detection*, Technical Report TUM-I0420, Technische University Muchen, November 2004.

- R. Sommer and V. Paxson, *Exploiting Independent State For Network Intrusion Detection*, Proc. ACSAC 2005.

- S. Staniford, V. Paxson and N. Weaver, "How to 0wn the Internet in Your Spare time," *Proc. 11th USENIX Security Symposium*, August 2002.
  - **`http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html`**

- Y. Zhang and V. Paxson, "Detecting Backdoors," *Proc. 9th USENIX Security Symposium*, August 2000
  - **`http://www.icir.org/vern/papers/backdoor-sec00.ps.gz`**

- Y. Zhang and V. Paxson, "Detecting Stepping Stones," *Proc. 9th USENIX Security Symposium*, August 2000.
  - **`http://www.icir.org/vern/papers/stepping-sec00.ps.gz`**