U.S. Department of Energy

Office of Science

BERKELEY LAB

NERSC

ICSI

Cybersecurity at Open Scientific Facilities

(tutorial 140)

SC2004

PITTSBURGH PA NOVEMBER 6 - 12

James Rothfuss

Lawrence Berkeley National Laboratory

Information Technologies and Services Division

jsrothfuss@lbl.gov

Dr. Vern Paxson

ICSI Center for Internet Research (ICIR)

International Computer Science Institute

and

Lawrence Berkeley National Laboratory

vern@{icir.org, ee.lbl.gov}

William T.C. Kramer

Lawrence Berkeley National Laboratory

National Energy Research Scientific Computing Center

kramer@lbl.gov

Stephen Lau

Lawrence Berkeley National Laboratory

National Energy Research Scientific Computing Center

slau@lbl.gov

Today’s Topics

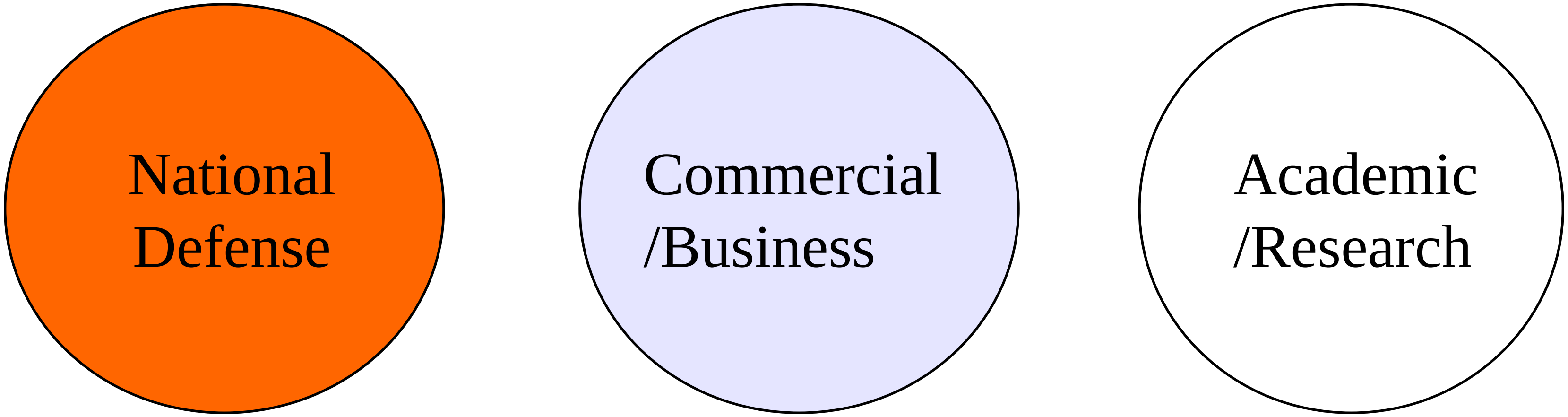
- Security, Protection, and the Open Environment
- The Threat
- System Level Protection
- Network Protection
- Protecting the Grid
- Incident Response
- Real World Example: Protecting SCinet
- Risk

Security, Protection, and the Open Environment

James Rothfuss

What is an
Open Environment?

Consider three forms of environment:



National Defense

Commercial /Business

Academic /Research

Classical Notion of Security

Secure

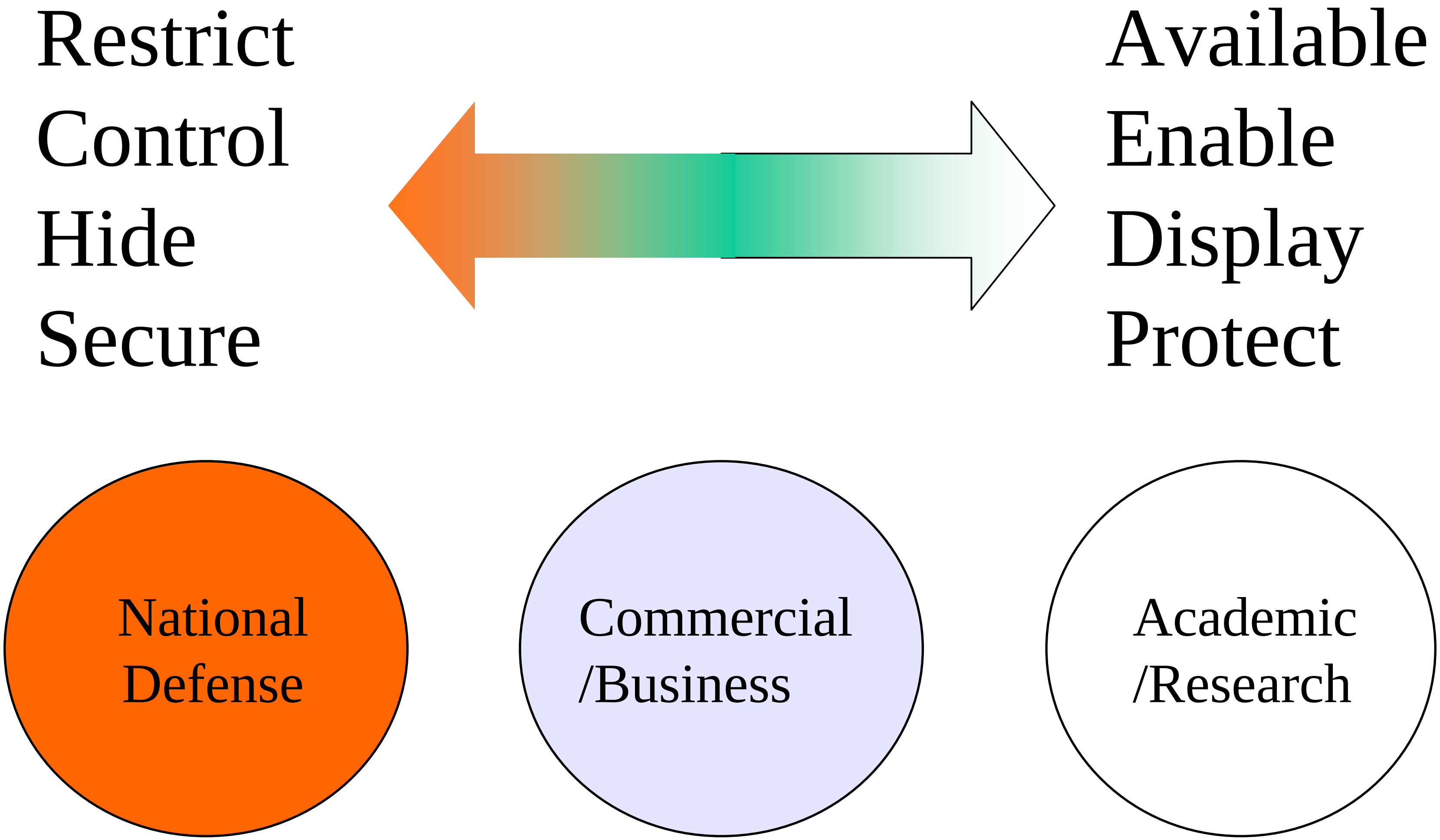
Restrict

Control

Hide

Restrict by Default, Open only as Necessary

OPEN by default
restrict only as necessary



Restrict
Control
Hide
Secure

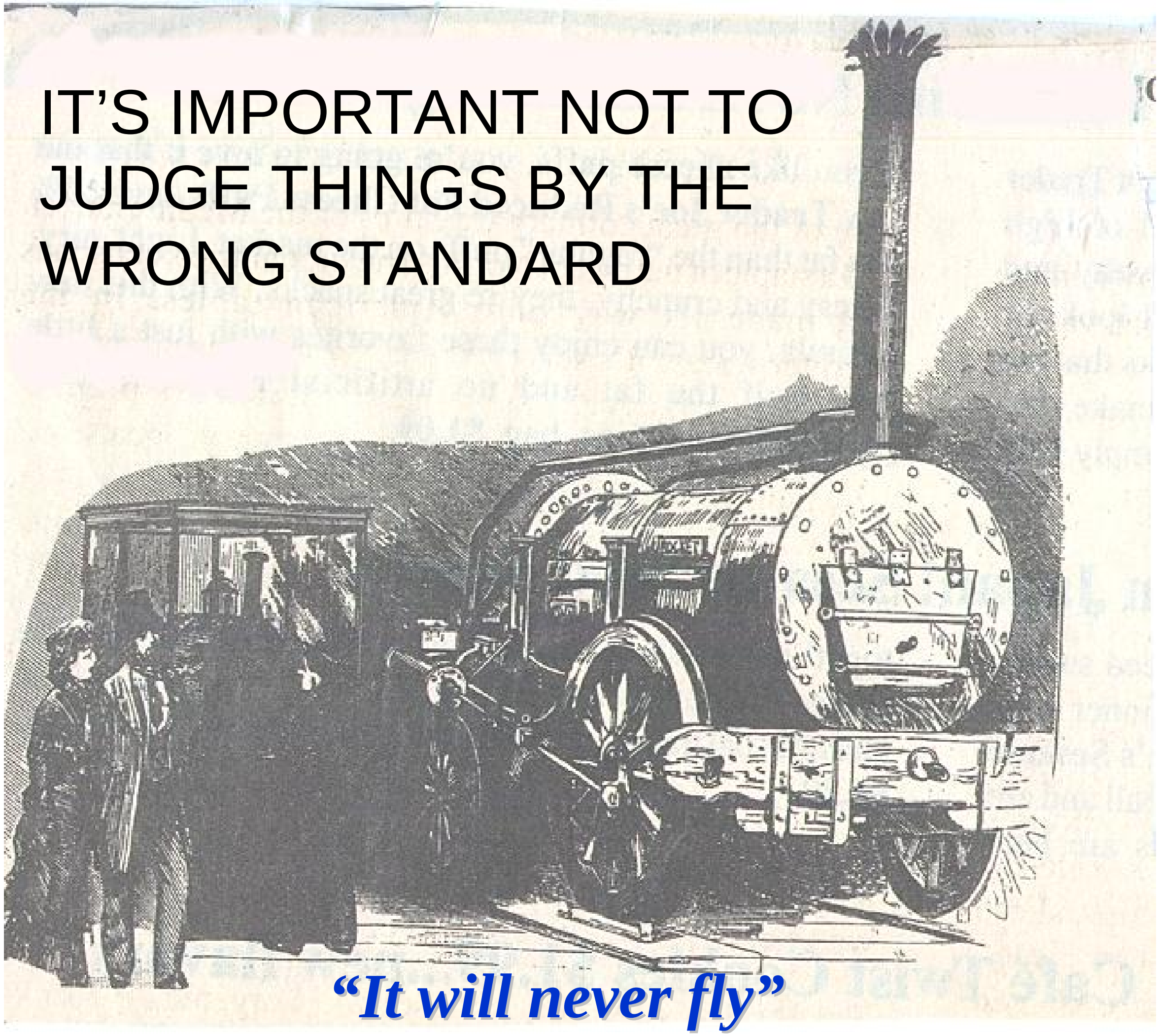
Available
Enable
Display
Protect

National Defense

Commercial /Business

Academic /Research

What is Security?



IT’S IMPORTANT NOT TO JUDGE THINGS BY THE WRONG STANDARD

“It will never fly”

Security is an Attribute, not an Object

"Nothing useful can be said about the security of a mechanism except in the context of a specific application and environment" (Bob Courtney - IBM Information Systems)

Attributes of a system:

Performance (Speed)

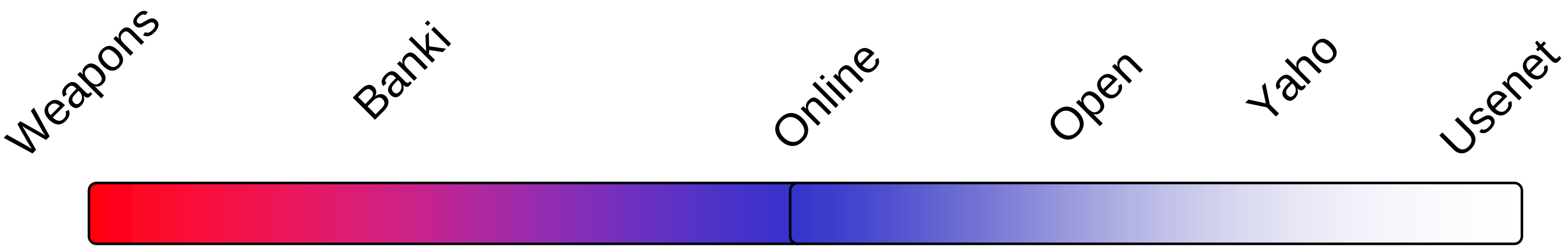
Functionality

Usability

Reliability

SECURITY

Primary Protection Concern

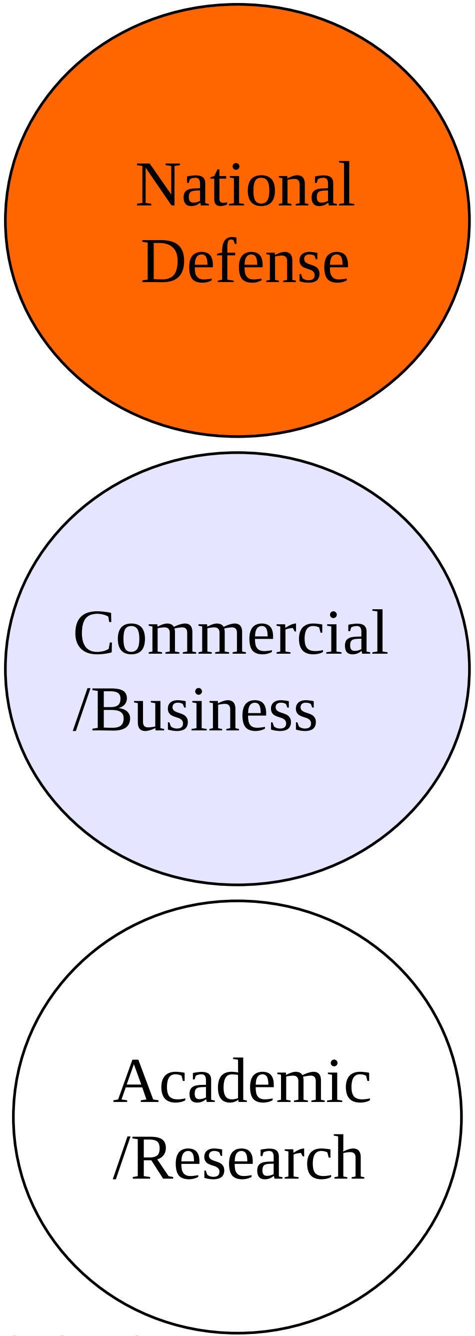


Information Protection

Resource Protection

In the Open Environment computers are TOOLS

Protect the tools (computers, network) from threats that could render those tools ineffective



National Defense

National Secrets (information)
Computers with defined purpose
Ability to retain a strong defense

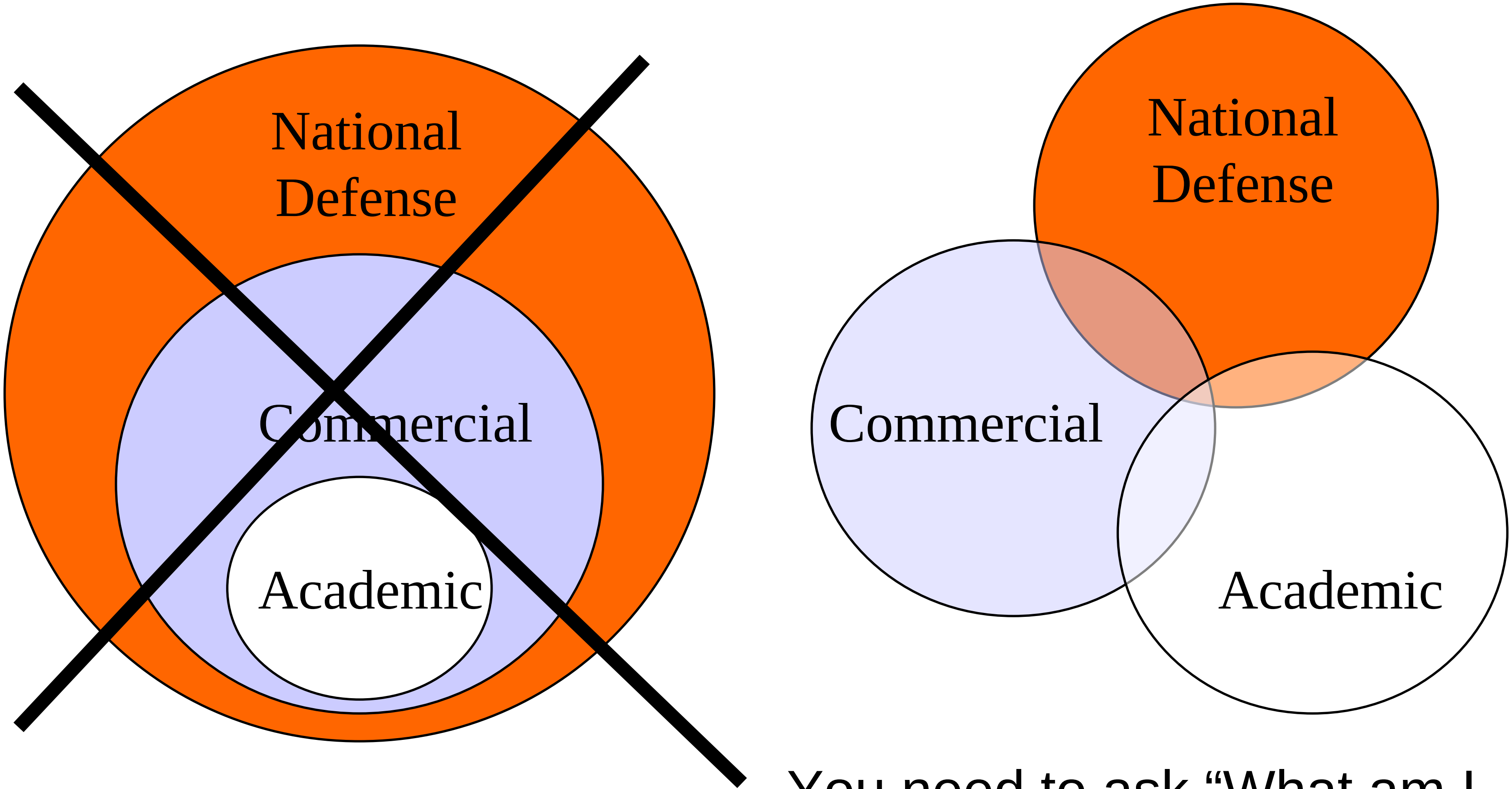
Commercial /Business

Monetary Assets
Endless repeat of transactions
Ability to run the business

Academic /Research

The process of research
Computers as general purpose tools
Ability to create Ideas

Different Environments require Different Protective Measures



You need to ask “What am I trying to protect and what am I trying to protect it from”

Security Changes

You are not solving a static puzzle

You are in a Chess Game against thousands of intelligent opponents

... where your opponent cheats

Security vs. Protection?

"The question 'Is the system secure?' is essentially meaningless.

The meaningful question is 'Is the system protected against events believed to be harmful?'"

(Alan Krull, IBM Informations Systems)

“The troops were sent to **secure** the village”

“The troops were sent to **protect** the village”

Which village would you like to live in?

This tutorial is about PROTECTION, not security

The Threat

Dr. Vern Paxson

What is the Threat?

- **A crucial basic question is** *What is your threat model?*
  - **What are you trying to protect?**
  - **Using what sort of resources?**
  - **Against what sort of adversary who has what sort of goals & capabilities?**

- **It's *all* about shades of grey, policy decisions, limited expenditure, risk management**

Threat models (cont)

- E.g.: a federally funded, national research laboratory
- No classified research
- Few “crown jewels”
  - Maybe: very expensive machines, financial data, medical patients
- #1 threat: <u>newspapers</u>

Threat models (cont)

- **Why? A single high-profile news report can** *percolate up to D.C.***and** *cost millions of dollars in funding***.**

- **Implication: <u>avoid embarrassment</u>**
  - **E.g.: compromised hosts used to launch outbound attacks.**
  - **E.g.: porn or MP3 servers/clients.**
  - **E.g.: public bragging about compromising a.gov site.**
- **Implication: <u>don’t make funding agency look bad</u>**

Threat models (cont)

- **This is *not* to say that a break-in without these is negligible.**
- **Loss of service and/or impaired productivity *does* matter.**
- **But: understanding threat model helps focus priorities.**

Today's Threats for an Open Environment

- **In general, two types of threats:** *insider* **and** *outsider*.
- **Insider threat:**
  - **Hard to detect ⇒ hard to quantify**
  - **Can be** *really* **damaging**
  - **In our experience:** *rare*

Outsider Threat

- **Attacks from "outside" (over the Internet) are** *ubiquitous***.**
- **Internet sites are** *incessantly* **probed**
  - **Per "Internet background radiation" study**
- **Using simplistic definition of "scan" (connection attempts to ≥ 20 hosts), LBNL's address blocks are scanned 100s of times/day** *after* **removing firewalled ports, per Cube Of Doom.**
- **More refined/sensitive definition ("TRW")** *detects 10s of thousands* **of remote scanners.**
- **What do they scan for? A wide and** *changing* **set of services/vulnerabilities, attacked via "***auto-rooters***".**

Other Outsider Threats

- **Account compromise:**
  - **Via password-guessing, password cracking, passwords sniffed elsewhere**
  - **Via trojaned SSH servers : - (**
  - **Increasingly, interactive traffic is invisible due to encryption**
  - **We don't generally see session hijacking**
- **Another concern: "phishing" for personal information used for monetary gain**
- **Big BIG concern: laptops / home machines infected elsewhere.** *The notion of "perimeter" has become diffuse and porous.*

What Are They After?

- Short answer: Not Us. (i.e., attacks are not targeted)

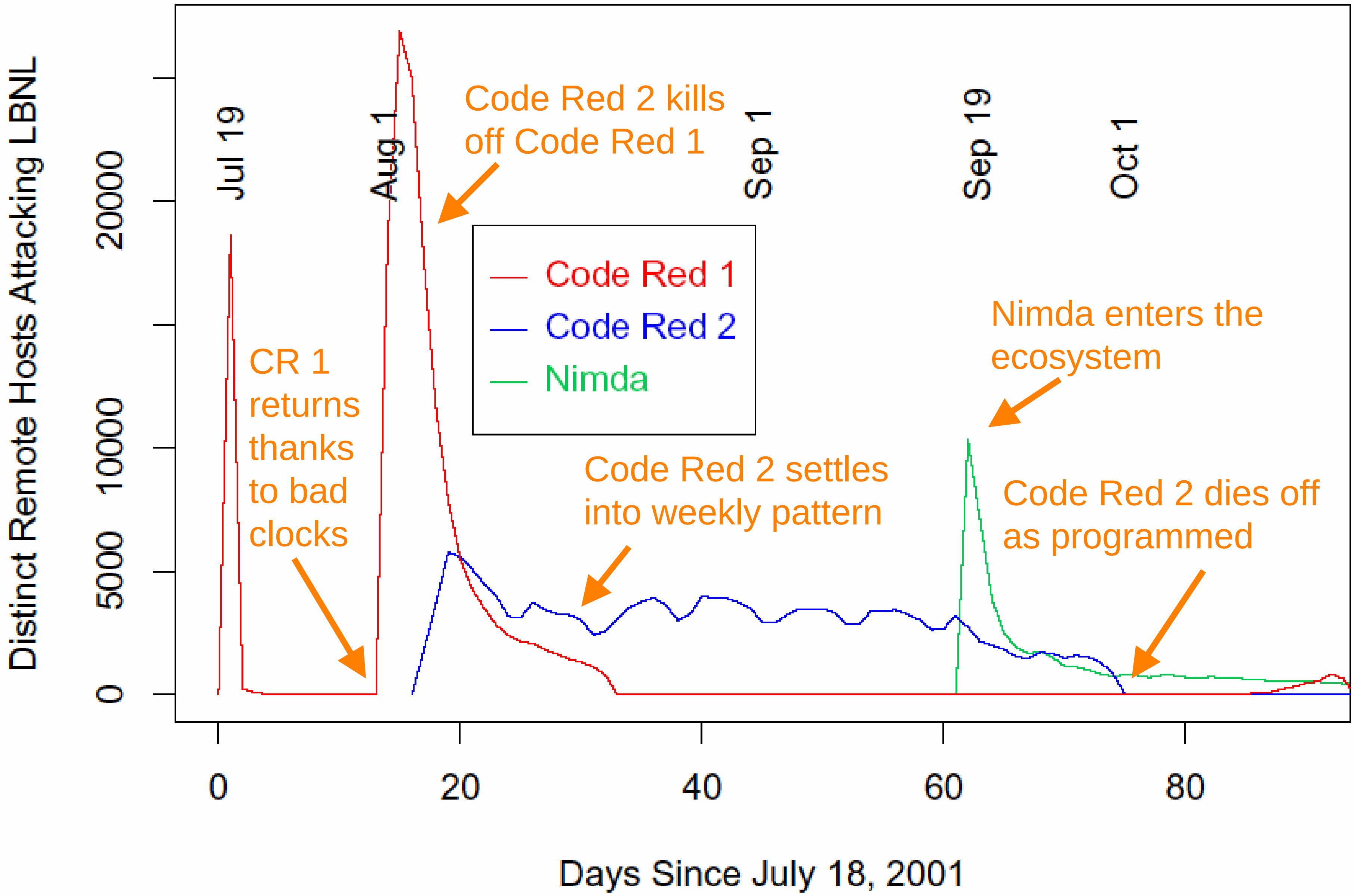
- They seek "Zombies" for:
  - DDOS slaves
  - Spamming
  - Finding more targets

- They seek bragging rights:
  - E.g., via IRC or Web page defacement

- They rarely cause damage beyond cleanup costs.

And Increasingly, They're on Autopilot

- **Self-propagating malware:** *worms* **and** *viruses.*
- **Constitutes a major portion of "background radiation":**
  - **Worms are now endemic**
  - **Still propagating years after their release**
  - **Some have strange periodic cycles (esp. Code Red 1, which spreads 1st-19th of each month, dies on 20th due to bug)**
  - **Others are** *parasitic*, **exploiting backdoors left by other worms**
  - **Together worms form an** *ecosystem*
- **Likely to get worse for a good time come**



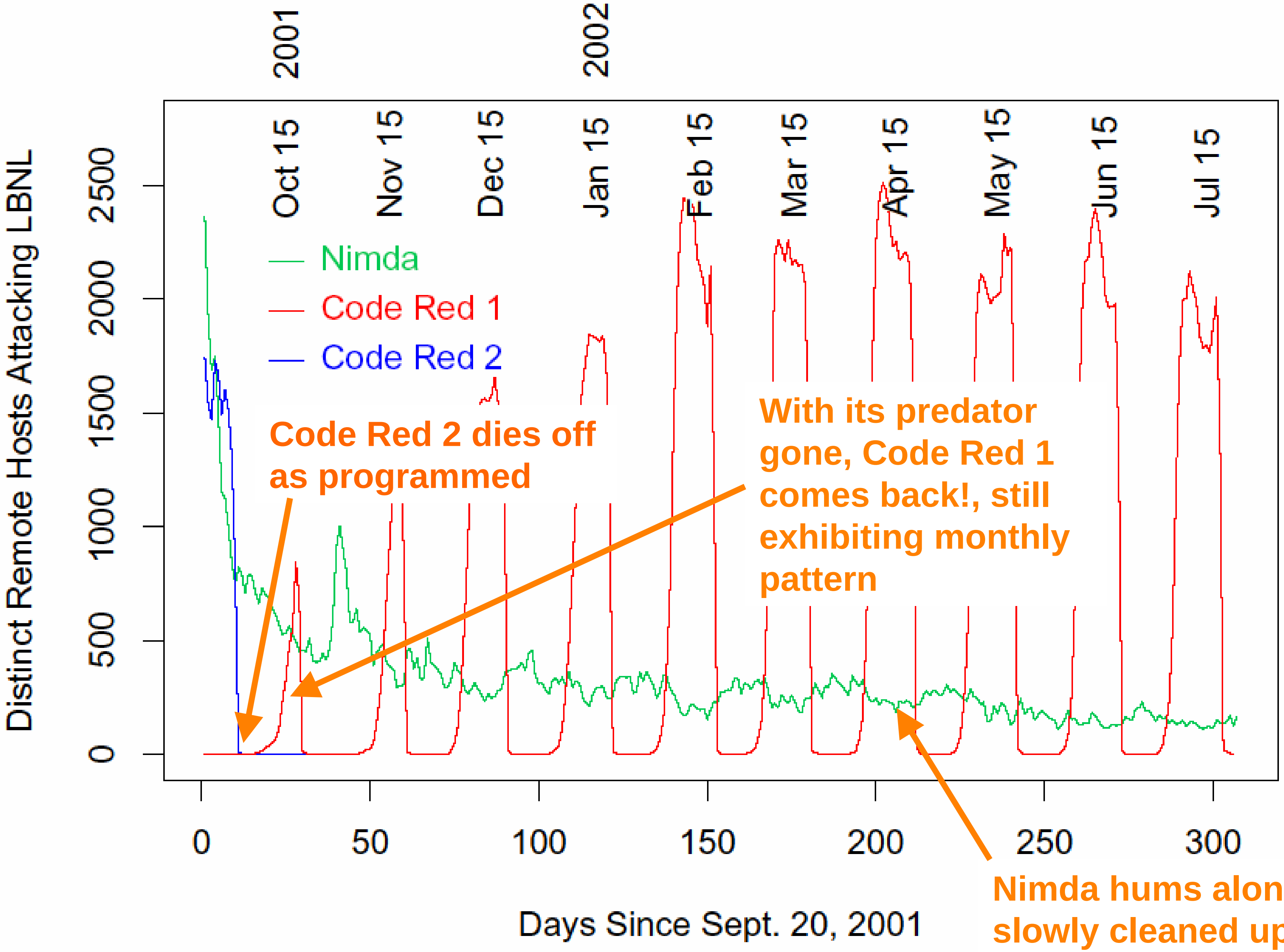
Code Red 2 kills off Code Red 1

CR 1 returns thanks to bad clocks

Code Red 2 settles into weekly pattern

Nimda enters the ecosystem

Code Red 2 dies off as programmed



Code Red 2 dies off as programmed

With its predator gone, Code Red 1 comes back!, still exhibiting monthly pattern

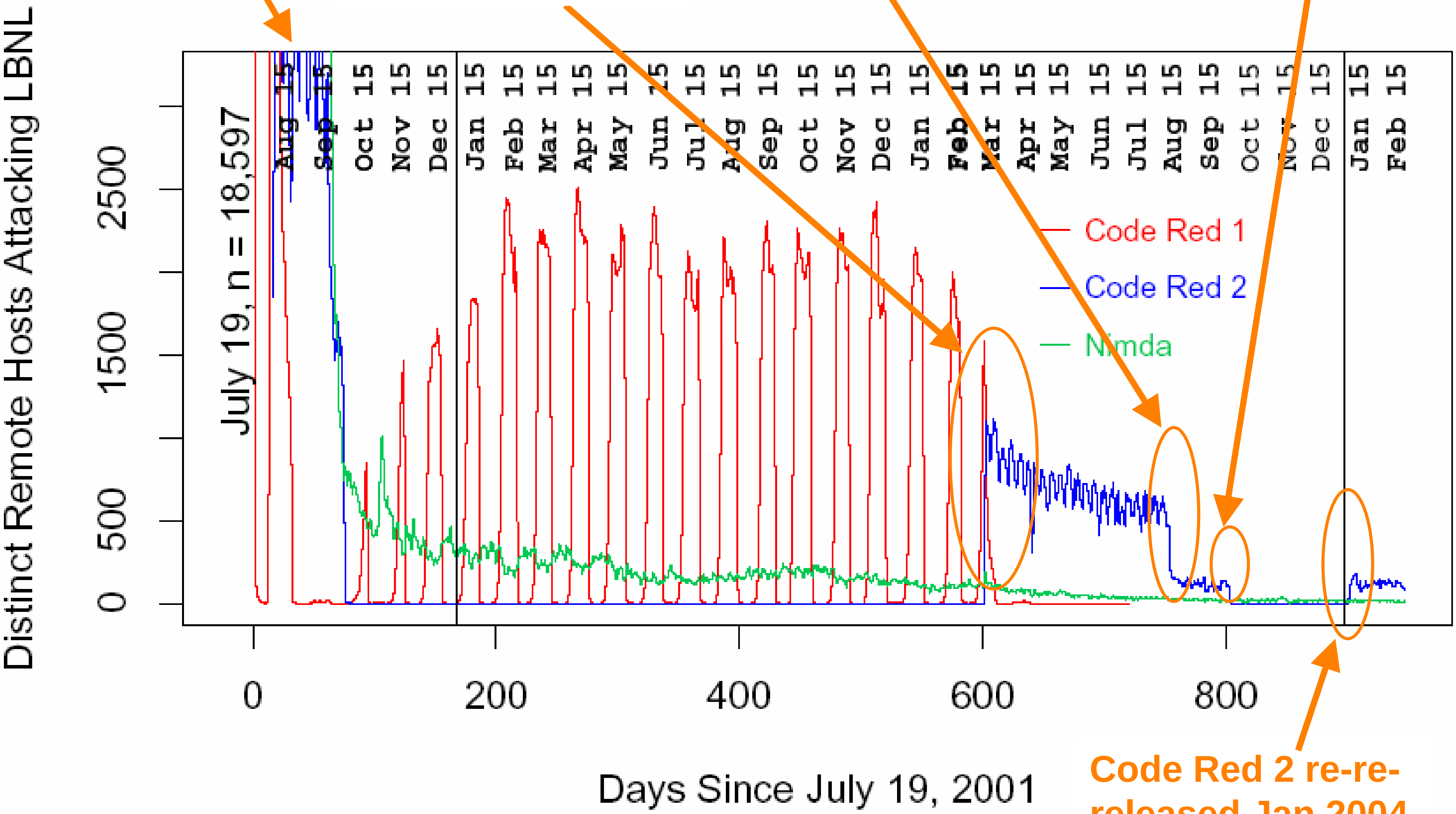
Nimda hums along, slowly cleaned up

Code Red 1 and Nimda endemic

Code Red 2 re-released with Oct. 2003 die-off

80% of Code Red 2 cleaned up due to onset of *Blaster*

Code Red 2 dies off again



Code Red 2 re-re-released Jan 2004

System Level Protection

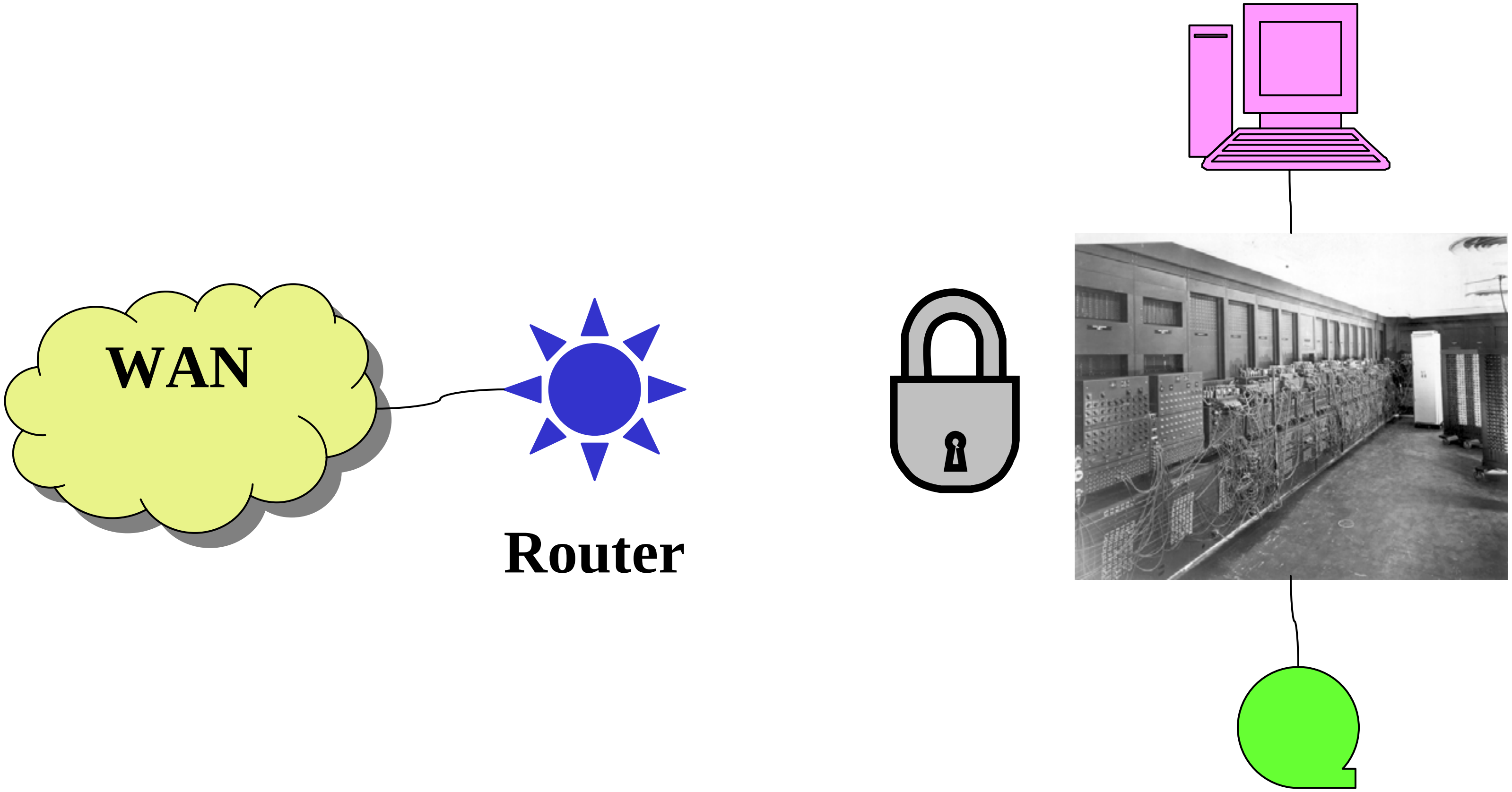
William T.C. Kramer

Agenda

- Security Approaches
- System Protection

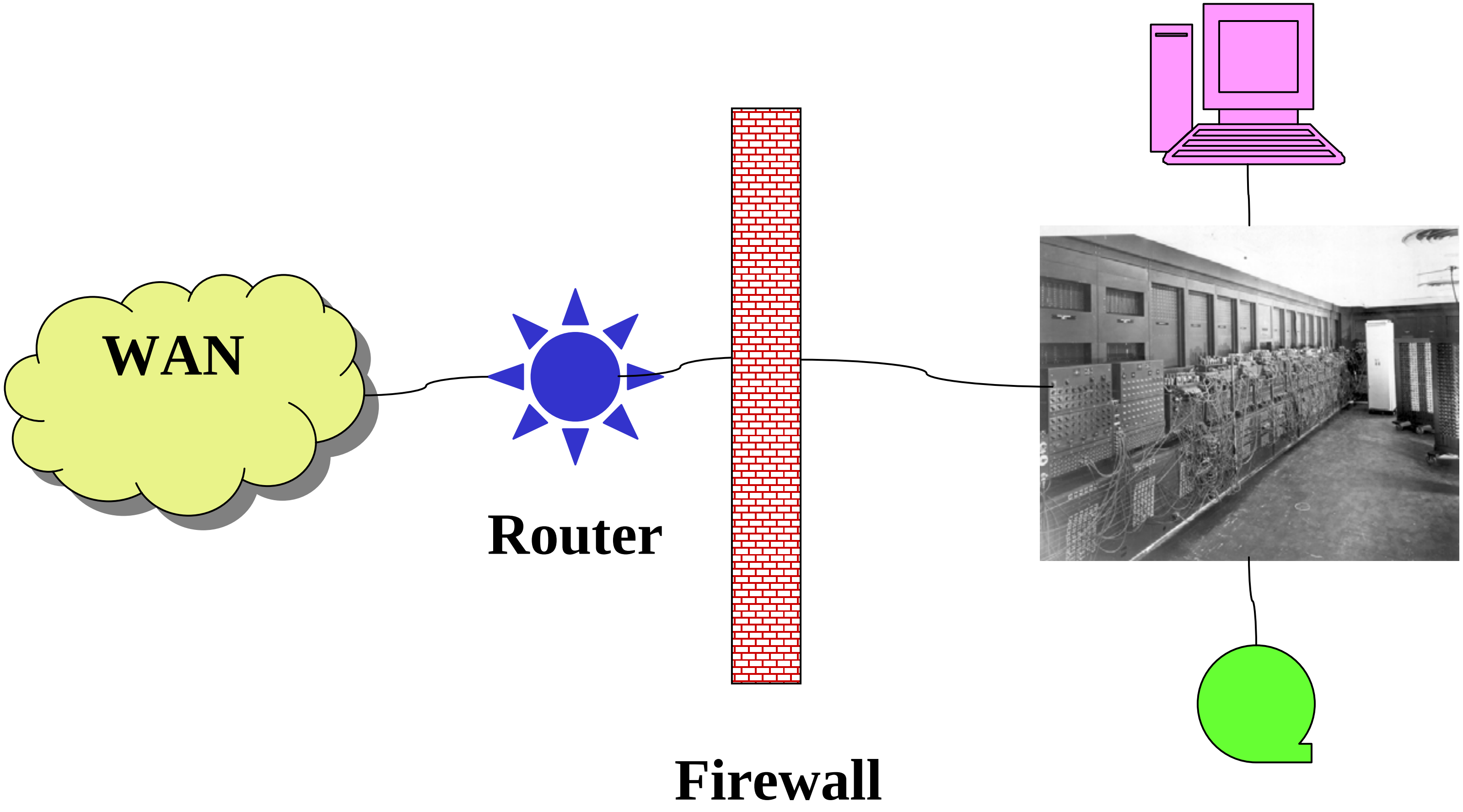
Security Approaches

Classified/Isolated



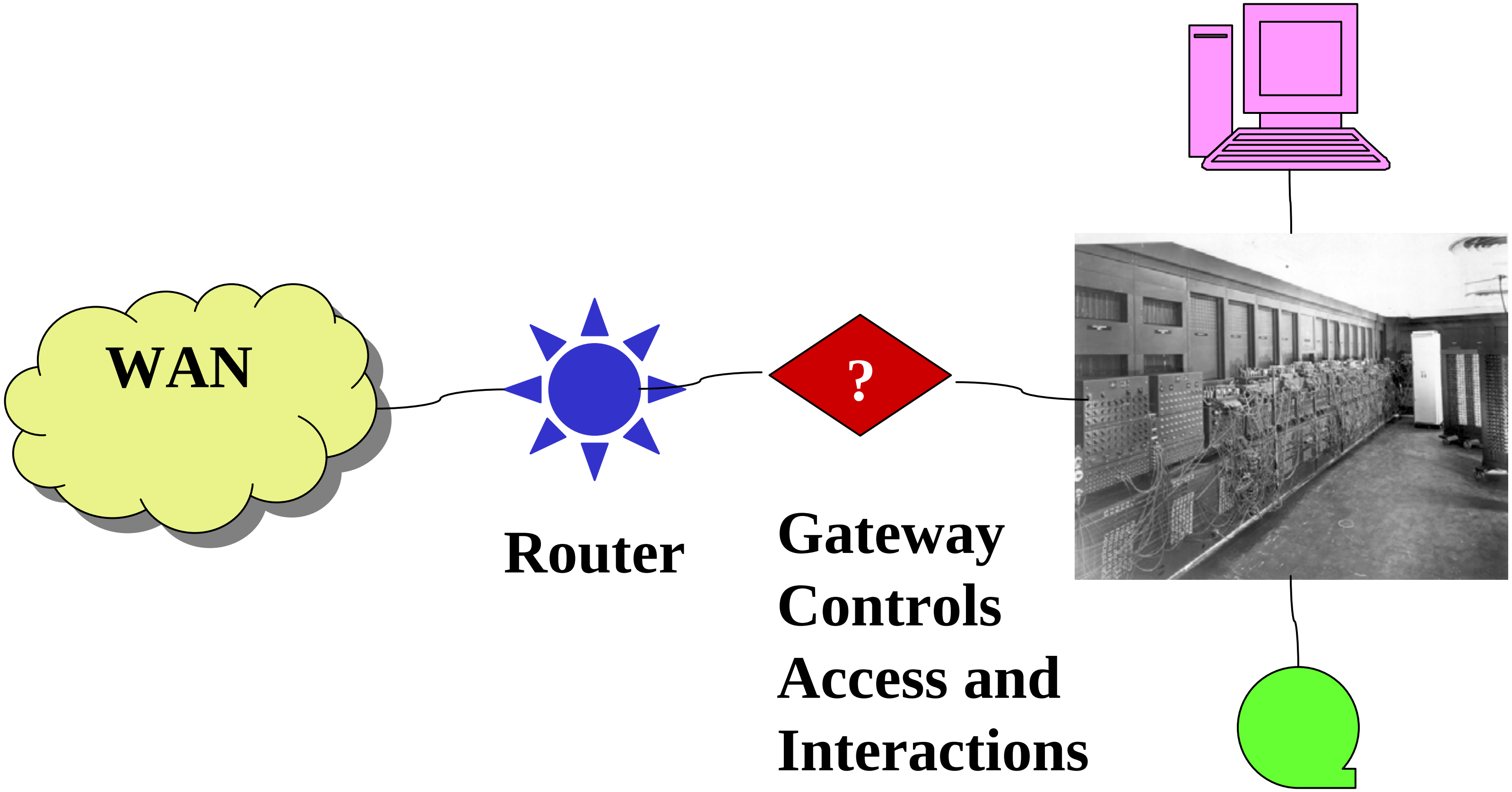
Security Approaches

Industrial



Security Approaches

Gateway



WAN

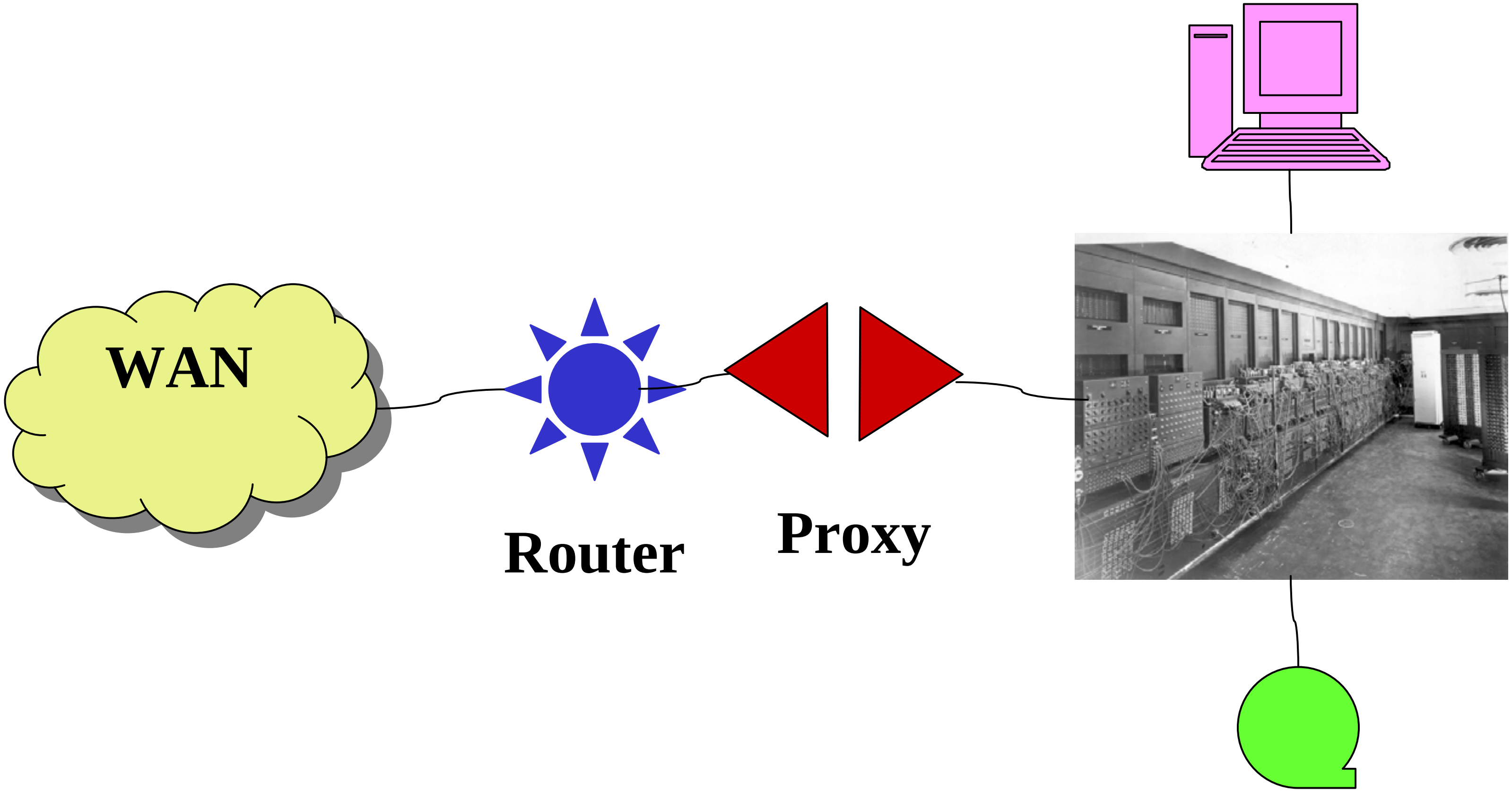
Router

?

Gateway Controls Access and Interactions

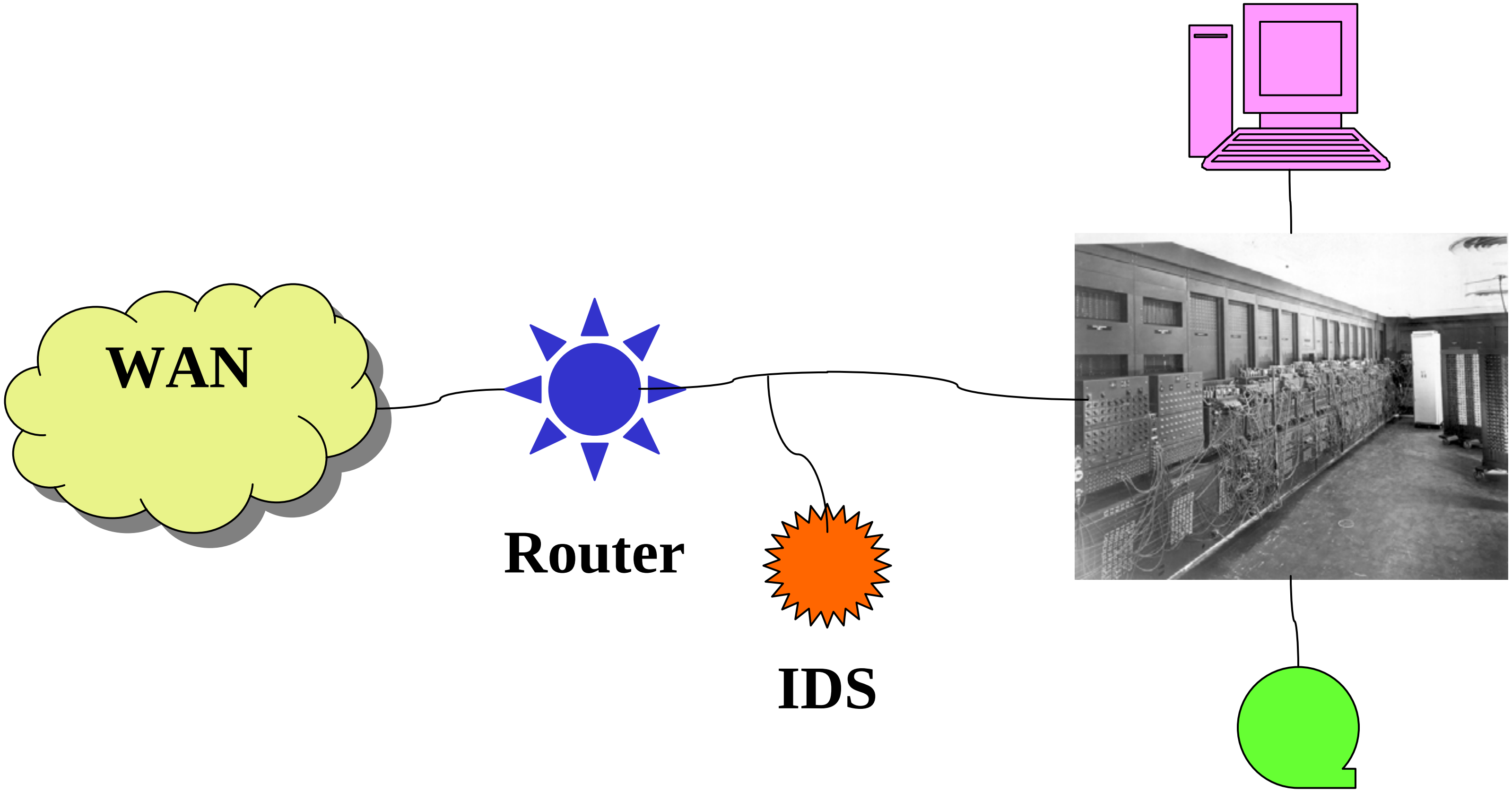
Security Approaches

Proxy



Security Approaches

Open



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

Characteristic Environment

- **Network**
  - **High speed network connections (possibly multiple)**
    - e.g. OC-48 at NERSC
  - **Wireless**
  - **Modem pools**

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

Characteristic Environment

- **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, in a general sense, regards controlling the way computers are used**
    - ⇒ How information and processing power can be accessed, manipulated and shared.
  - **Has to represent external laws and regulations, organizational mission, goals and business practices**
  - **Mandatory and Discretionary policies and rules**
- **Policy should be broad and change infrequently**
  - **Should be guided by the site mission and philosophy**
  - **Typically, policy change involved a long process and lots of review**
- **Business practices should have the details of policy implementation, be as specific as needed and change as needed**
  - **Changes are not bureaucratic but technical**
  - **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 a set of access control rules that constraints an entities' (person or program) access to information and/or resources on the basis that entities' authorization
  - i.e.To root or not to root, that is the question
- Discretionary – An individual entity 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**
  - **Multiple Bro systems for monitoring**
- **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
- Excellent people with enough time to spend on protection

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 is

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 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
- Public Key Infrastructure (PKI) is a system that uses digital certificates to increase the reliability of authentication. Before you can use the certificate authentication, certificates have to be created with a Certificate Authority (CA) software.
- The Lightweight Directory Access Protocol (LDAP) is a de facto standard to distribute certificates. Using the LDAP enables interoperability with third party directory servers, which are based on the LDAP standard.

Keys and Certification

- **Make is possible for users to store key information with you – 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**

Cluster Networking

- 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
- Definition of node functions

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

Logging and Monitoring

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

Scanning

- **Scanning helps prevent mistakes that lead to vulnerability**
- **Examples**
  - **Home grown**
  - **Cfengine, binaudit, St Michael**
    - St. Michael is a set of kernel modules that provides integrity checks of the Linux kernel. It does this by save md5 hashes of various critical memory regions in the kernel and then routinely checking these hashes. Some of the items in checks are...
      - ⇒In addition, it provides the following...
        - →Make /dev/kmem read only
        - →Make files really immutable (you can't chattr -i even as root)
        - →Attempts to recover kernel text from backup
        - →Reboots system if recovery isn't possible

Scanning

- Scanning helps prevent mistakes that lead to vulnerability
- Examples
  - Cfengine, binaudit, St Micheal
    - St. Michael is a set of kernel modules that provides integrity checks of the Linux kernel. It does this by save md5 hashes of various critical memory regions in the kernel and then routinely checking these hashes. Some of the items in checks are...
      - ⇒In addition, it provides the following...
        - →Make /dev/kmem read only
        - →Make files really immutable (you can't chattr -i even as root)
        - →Attempts to recover kernel text from backup
        - →Reboots system if recovery isn't possible
  - Scan ports and services

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

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

Network Protection

Dr. Vern Paxson
James Rothfuss

A look at Network Intrusion Detection

- **Why network intrusion detection? Why not?**
- **Styles of approaches.**
- **An example of a NIDS: BRO.**
- **The fundamental problem of evasion, possible solutions.**
- **Detecting activity: sniffers, stepping stones, backdoors.**

What can you learn watching a network link?

- **Far and away, most traffic travels across the Internet unencrypted.**

- **Communication is layered with higher layers corresponding to greater semantic content.**

- **The entire communication between two hosts can be reassembled: individual** *packets* **(e.g., TCP/IP headers), application** *connections* **(TCP byte streams), user** *sessions* **(Web surfing).**

- **You can do this in real-time.**

Tapping links (cont)

- **Appealing because it’s *cheap* and gives broad coverage.**
- **You can have multiple boxes watching the same traffic.**
- **Generally (not always) undetectable.**
- **Can also provide insight into a site’s general network use.**

Problems with passive monitoring

- **Reactive, not proactive**
  - **However, this is changing w/ intrusion *prevention* systems**
- **Assumes network-oriented (often "external") threat model.**
- **For high-speed links, monitor may not keep up.**
  - **Accordingly, monitors often rely on filtering (kernel/BPF).**
  - **Very high speed: beyond state-of-the-art.**
- **Depending on "vantage point", sometimes you see only one side of a conversation (especially inside backbone).**
- **Against a skilled opponent, there is a fundamental problem of evasion: confusing / manipulating the monitor.**

Styles of intrusion detection — *Signature-based:*

- **Core idea: look for specific, known attacks.**
- **Example:**

```
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
```

Signature-based (cont)

- **Can be at different semantic layers, e.g.: IP/TCP header fields; packet payload; URLs.**

- **Pro: good attack libraries, easy to understand results.**

- **Con: unable to detect new attacks, or even just variants.**

Styles of intrusion detection — *Anomaly-detection*

- Core idea: attacks are *peculiar*.

- Approach: build/infer a profile of "normal" use, flag deviations.
- Example: "user `joe` only logs in from host A, usually at night."

- Note: works best for *narrowly-defined* entities.

- Pro: potentially detects wide range of attacks, including novel.
- Con: potentially misses wide range of attacks, including known.
- Con: can potentially be "trained" to accept attacks as normal.

Styles of detection — *Activity*-(or *Specification*-) based

- Core idea: piece traffic into events, look for patterns of activity that deviate from a site's *policy*.
- Example: "user `joe` is *only* allowed to log in from host A."

- Note: this is the primary approach used by Bro.

- Pro: potentially detects wide range of attacks, including novel.
- Pro: framework can accommodate signatures, anomalies.

- Con: policies/specifications require significant development & maintenance. Harder to construct attack libraries.

Some general considerations about the problem space

- **Security is about *policy*.**
- **The goal is risk management, not bulletproof protection.**

- **All intrusion detection systems suffer from the twin problems of *false positives* and *false negatives*.**
- **These are not minor, but an Achilles heel.**

- **Scaling works against us: as the volume of monitored traffic grows, so does its diversity.**

- **NIDS research "in the lab" is *far removed* from operational reality.**

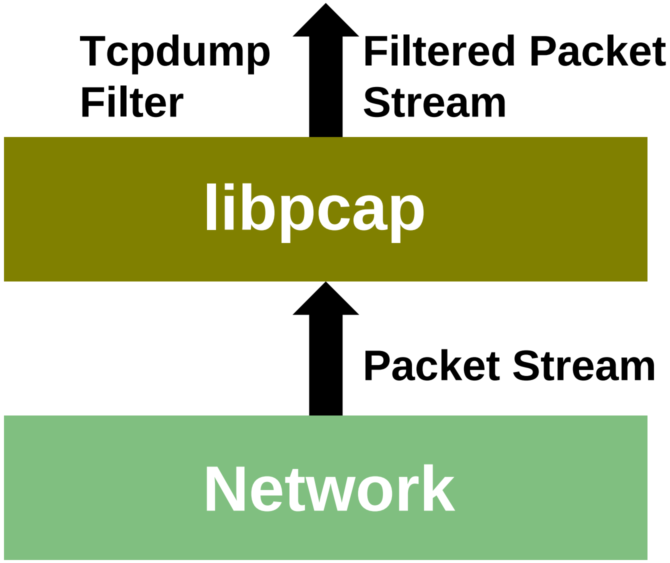
A look at Bro — design goals & constraints

- **High-speed, large volume monitoring (FDDI/GigEther).**
- **Real-time notification.**
- **Mechanism separate from policy.**
- **Extensible.**
- **Avoid simple mistakes ⇒ specialized policy language.**
- **The monitor will be attacked.**

How Bro Works

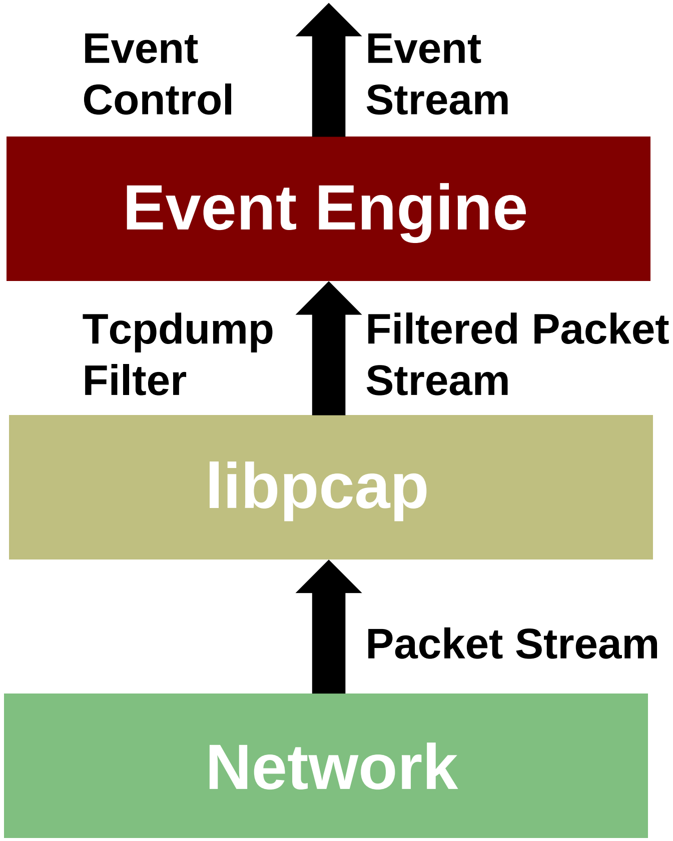
- Taps GigEther fiber link passively, sends up a copy of all network traffic.

How Bro Works



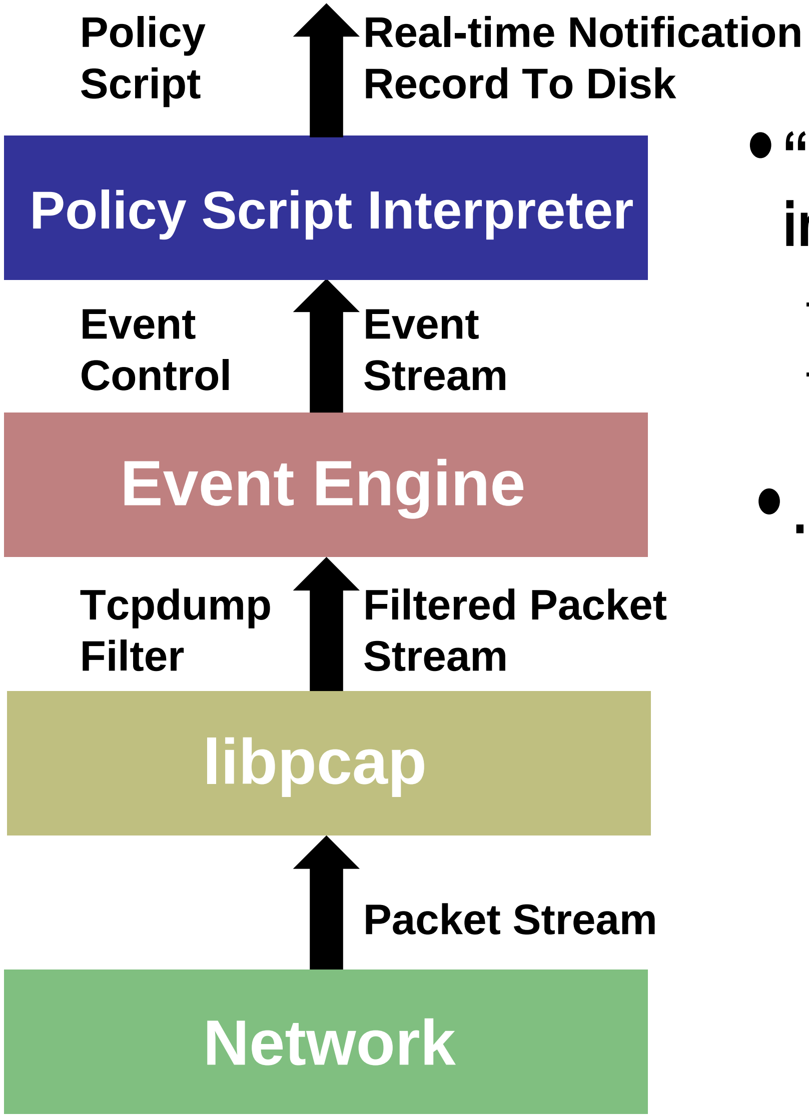
- Kernel filters down high-volume stream via standard *libpcap* packet capture library.

How Bro Works



- "Event engine" distills filtered stream into high-level, *policy-neutral* events reflecting underlying network activity
  - E.g., connection_attempt, http_reply, user_logged_in

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* or paging
  - Executes programs as a form of response

Event engine

- Event engine does generic (non-policy) analysis.
- E.g. Connection-level:
  - `connection_attempt`
  - `connection_finished`
- E.g. Application-level:
  - `ftp_request, pm_request getport, login_input_line`
- E.g. Activity-level:
  - `login_success, stepping_stone, ssh_signature_found`

- If you define a handler for a given event, it will be invoked any time the event occurs. Otherwise, event engine skips the work for detecting the event.

Extending the Engine

- **Engine is implemented using a C++ class hierarchy.**

- **For example**, `TelnetConn` **derives from** `LoginConn`, **which derives from** `TCP_Connection`, **derives from** `Connection`.

- `Telnet_Conn` **uses two** `TCP_NVT` **(network virtual terminal) objects, one per connection direction.**

- `TCP_NVT` **derives from** `TCP_EndpointLine`, **which derives from** `TCP_EndpointContents`, **which derives from** `TCP_Endpoint`.

The Bro Policy Language

- **Strongly typed -> catch errors at compile time.**
- **Arithmetic types,** `pattern, time, interval, port, addr`.
- **Records, associative tables & sets:**
  - `global ftp sessions: table[conn id] of ftp session info`
- **Strings are counted rather than NUL-terminated:**
  - `USER nice\0USER root`

Analyzers

- **For all TCP connections (via SYN/FIN/RST packets):**
  - **start time, duration, service, addresses, sizes**
  - **port, address scanning, including stealth scans**

- **App's: DNS, HTTP, SMTP, FTP, NTP, Finger, Portmapper, Ident.**
- **Telnet and RLogin:**

```
Login_successful, login_failure,
  activating_encryption, login_confused
```

=> `login input line, login output line`

Prevention in Addition to Detection

- **“`rst`” terminates the local end of a TCP connection via RST packet(s). (Tricky for picky TCP stacks that insist on exact sequence numbers.)**

- **“`drop-connectivity`” talks to border router, throws away given remote traffic: a *reactive firewall*.**

- **Both invoked via `system`( ), per arbitrary policy.**

- **At LBNL, on typical day a few hundred scans dropped.**
- **Routers run with 1,000–4,000 ACL entries.**

Status

- **Operational since 1996.**

- **Two dozen monitor boxes deployed at LBNL, UCB, ICSI, Munich, DOE HQ.**
- **LBNL boxes see up to 2 billion packets/day (~23Kpps).**
- **Avg: 900 <u>filtered</u> pps; peaks: 37,000+ pps.**

- **Connection logs: 2.4 GB/day.**

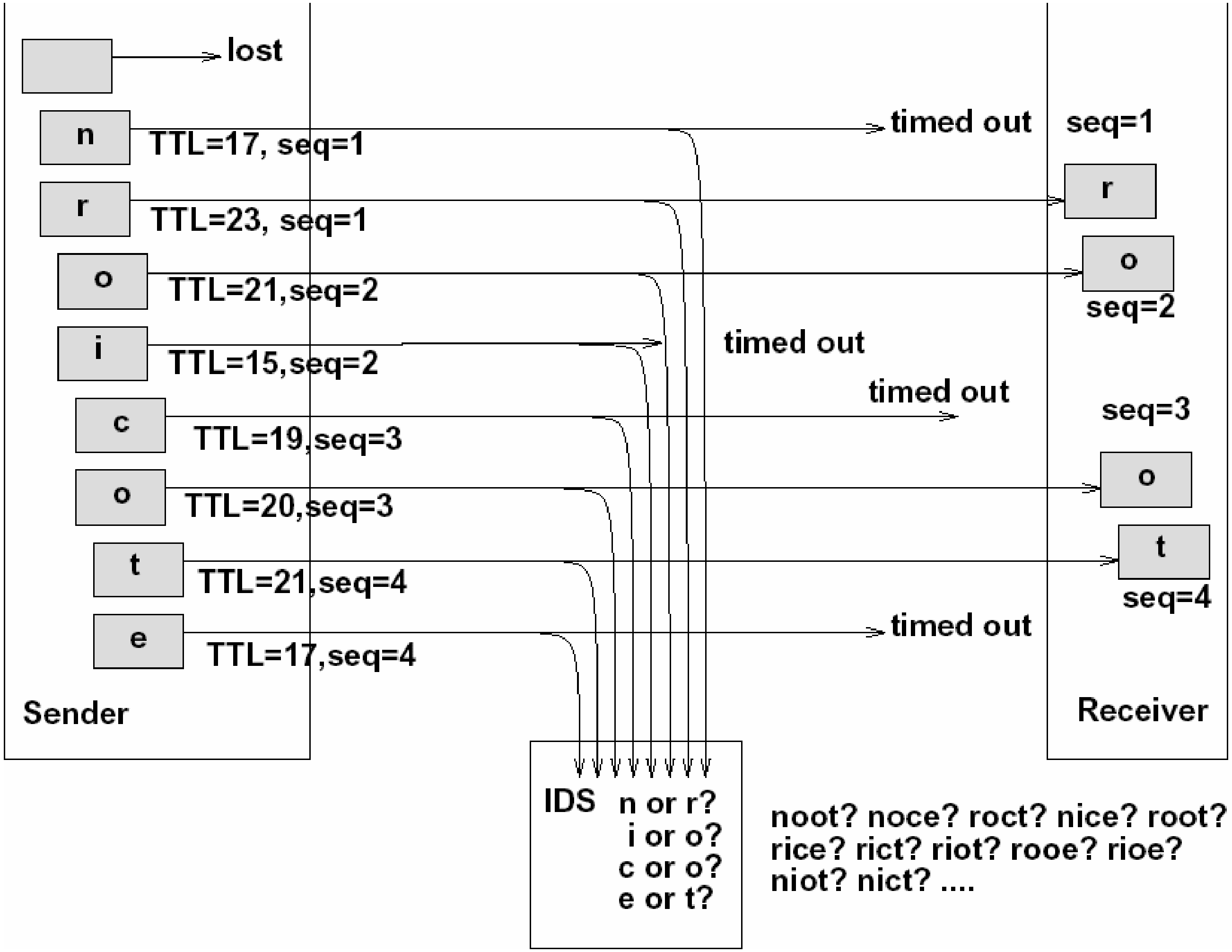
Status (cont)

- At LBNL/ICSI, “bulk trace” machines record some/all traffic for off-line analysis.
- Also monitor: NERSC, JGI, ESNET, internal nets.
- 92,000 lines of C++. Unix/libpcap-based.
- 13,000 lines of Bro scripts, most of it site-independent.

The Problem of Evasion

- **Consider *passive measurement*: scanning traffic for a particular string ("`USER root`")**
- **Easiest: scan for the text in each packet**
  - **No good: text might be split across multiple packets**
- **Okay, remember text from previous packet**
  - **No good: out-of-order delivery**
- **Okay, fully reassemble byte stream**
  - **Costs** <u>state</u> **….**
  - **…. and still evadable**

Evading Detection Via Ambiguous TCP Retransmission



The Problem of Evasion

- **Fundamental problem passively measuring traffic on a link: Network traffic is** *inherently* **<u>ambiguous</u>**

- **Attackers can craft traffic to confuse/fool monitor**

- **Okay, can’t you then generate an alarm when you see ambiguous traffic?**

Crud seen on a DMZ

- **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 (!).**

Countering Evasion-by-Ambiguity

- **Involve end-host: have it** *tell you* **what it saw**
- **Probe end-host in advance to resolve vantage-point ambiguities ("active mapping")**
  - **E.g., how many hops to it?**
  - **E.g., how does it resolve ambiguous retransmissions?**
- *Change the rules - Perturb*
  - **Introduce a network element that "normalizes" the traffic passing through it to eliminate ambiguities**
    - E.g., regenerate low TTLs (dicey!)
    - E.g., reassemble streams & remove inconsistent retransmissions

Detecting activity — sniffer detection

- **Depending on your threat model, you can often get a <u>lot</u> of mileage out of detecting** *evidence of a compromise* **rather than the attack itself.**

- **E.g., at LBNL, inbound IRC = break-in.**

- **Another form: <u>sniffer detection</u>.**
  - **e.g., via** *increased ping times*
  - **e.g., via** *observing reverse DNS queries*
  - **e.g., via** *transmitting bogus username/password pairs*
  - **note: works for bad guys detecting IDS, too.**

Detecting “stepping stones”

- Internet attacks invariably do not come from the attacker’s own personal machine, but from a *stepping-stone*: an intermediary previously compromised.

- Furthermore, usually it is a *chain* of stepping stones.
- Manually tracing attacker back across the chain is virtually impossible.

- So: want to detect that a connection going into a site is closely related to one going out of the site.

Detecting stepping stones

- Approach:
  - Leverage unique on/off pattern of user login sessions.
  - Look for connections that end idle periods at the same time.
  - Two idle periods correlated if ending time differ by <= sec.
- If enough idle periods coincide => stepping stone pair.

- For A -> B -> C stepping stone, just 2 correlations suffices.
- (For A -> B -> . . . -> C -> D, 4 suffices.)

Detecting stepping stones

- **Works very well,** *even for encrypted traffic*.

- **<u>But:</u> easy to evade, if attacker is cognizant of algorithm.**

- **<u>And:</u> also turns out there are frequent** *legit* **stepping stones.**

Detecting backdoors

- “Backdoor”: a service installed on a compromised machine to allow the attacker to surreptitiously return.

- How to find access to these against sea of background traffic?

- General algorithm for interactive traffic (Zhang/Paxson 2000):
  - look for frequent small packets
  - look for small packets with large interarrivals

Detecting backdoors (cont)

- **Protocol-specific: SSH, Rlogin, Telnet, FTP.**

- **Algorithms also amenable to filtering for large perf. gain:**
- **e.g.,**

  -
    ```
    tcp[(tcp[12]>>2):4]=0x5353482D &&
    (tcp[((tcp[12]>>2)+4):2]=0x312E
    or tcp[((tcp[12]>>2)+4):2]=0x322E)
    ```

Detecting backdoors (cont)

- **Plus: a hack for detecting some *root* backdoors ("# ").**
  =>Found 437 root backdoors in single 24-hour period at UCB.

- **Also recognizers for non-interactive protocols:**
  - **HTTP, SMTP, Napster, Gnutella, KaZaA.**

- **In general, algorithms perform quite well.**
- **And: can employ <u>filtering</u> with little loss of accuracy.**

- **But: find many *legit* backdoors.**

Summary of Network Intrusion Detection

- **Security is not about bullet-proof; it's about** *policies* **and** *tradeoffs***.**
- **You can detect a whole lot by piecing together judiciously filtered network traffic into events reflecting activity …**
- **… but there are significant problems with evasion leading to an arms race.**
- **Traffic contains much more diversity/junk than you'd think, including** *incessant scanning* **for vulnerabilities.**
- **The endpoint host is a great location to look for attacks.**
- **Increasingly, NIDS need to be supplemented by an active forwarding element, for both high performance and** *intrusion prevention***.**

Network Equipment Tracking System

Fully automated vulnerability discovery and elimination

- Network information continuously collected
- Systems continuously scanned
- Network vulnerabilities detected as they appear
- Vulnerabilities immediately resolved
  - Automatically Blocked
  - Automatically alert owners/sys admins
  - Automatically remove blocks when vulnerabilities are fixed

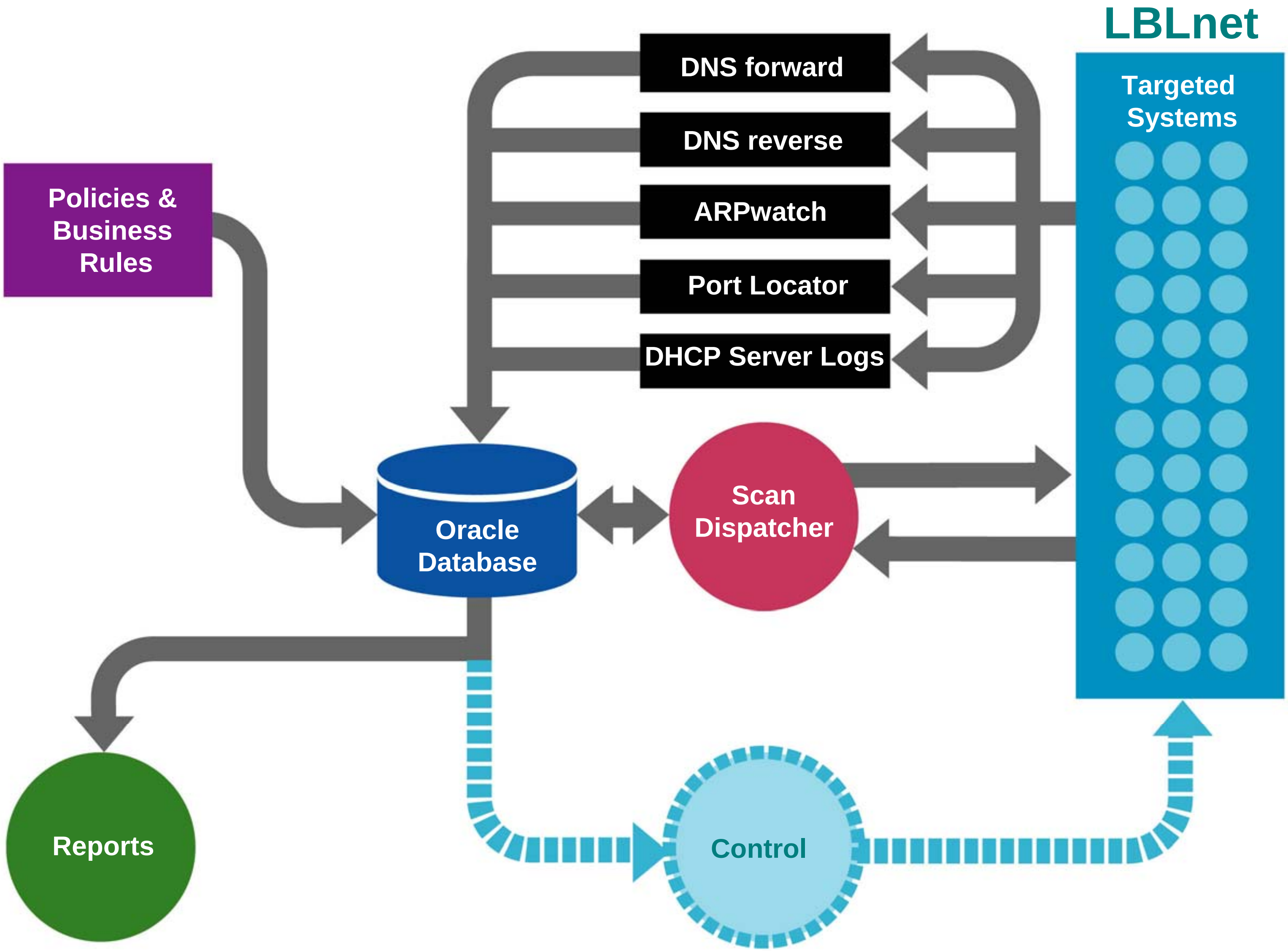
Safe systems given full access

-Internet access is maximized

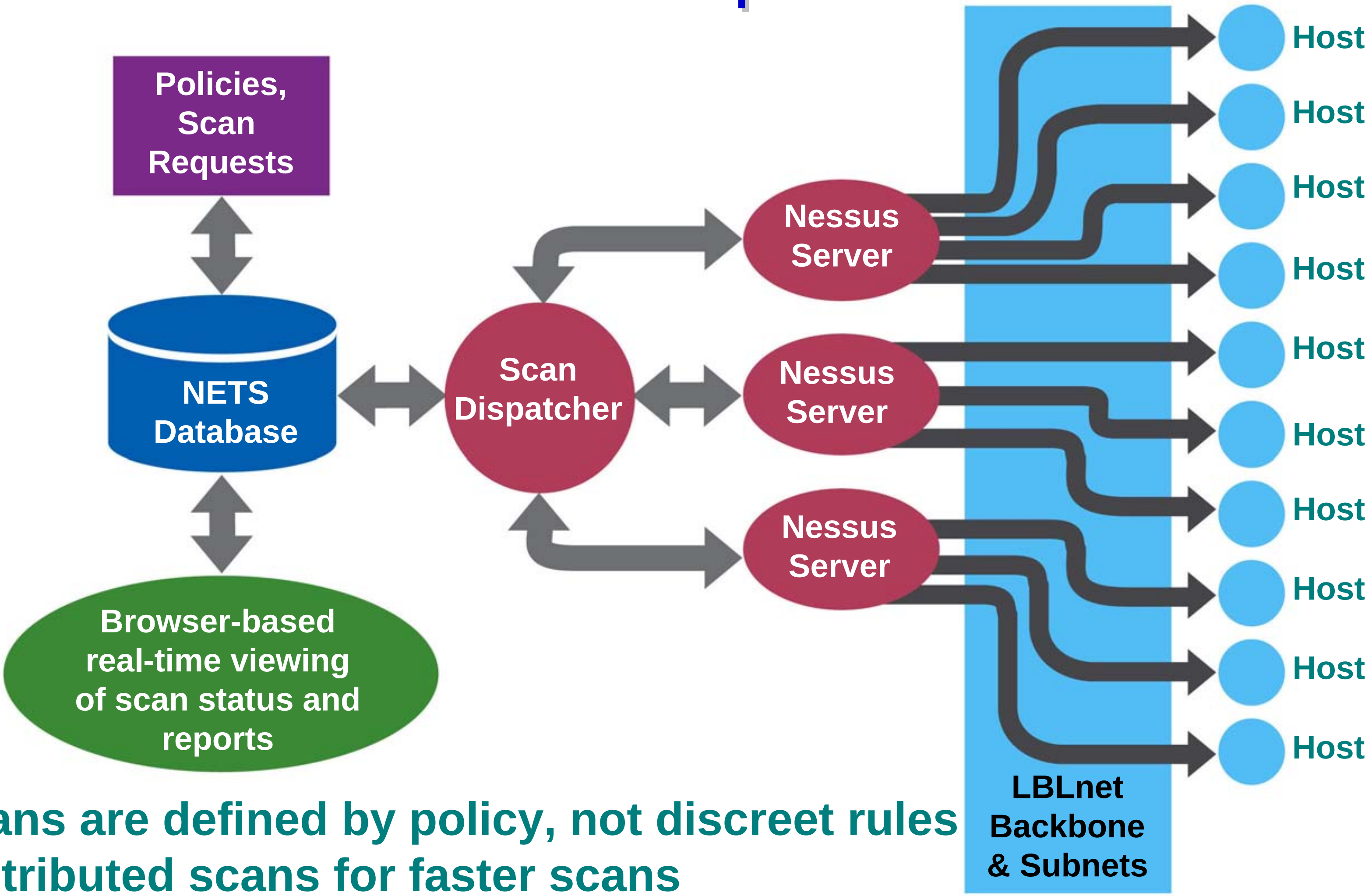
LBNL Network Equipment Tracking System (NETS)

- ✓ Network information collected
- ✓ System connections are detected
- ✓ Systems are probed
- ✓ Vulnerabilities blocked
- ✓ Automatic block removal as vulnerabilities are fixed

NETS Prototype



The Scan Dispatcher



- Scans are defined by policy, not discreet rules
- Distributed scans for faster scans
- Priority setting
- Scans initiated by NETS (automatic)

DHCP Jail

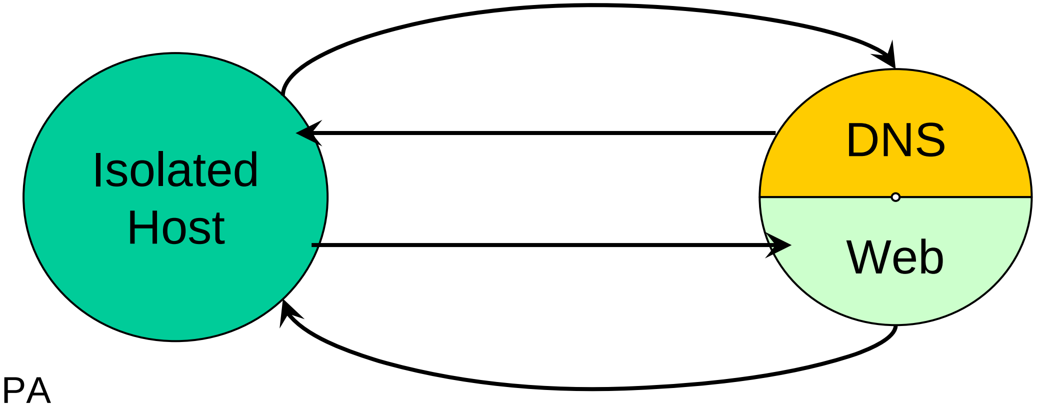
Deny Boot – DHCP server refuses service to a given host

RESULT: NO NETWORK ACCESS

(note, this does not necessarily deny the ability for the host to boot)

Host Isolation – Do not provide a default route (gateway address) and one source route. Direct the single source route to a special DNS server that resolves *everything* back to itself.

RESULT: NETWORK ACCESS TO ONE OTHER HOST



Future Advances

- Gathering Information
  - Continuous load balanced scanning
  - Incorporate more sensors information
- Access Control
  - Active blocking at DMZ router
  - Firewall for better access control
  - Active blocking on internal routers
- Host Inventory
  - Network history
  - Mandatory registration
  - Host and owner certification
  - Deploy host agent software

Protecting The Grid

Stephen Lau

The Grid

- Unfettered access to computing resources across organizational and geographical boundaries
- Ad Hoc collections of collaborators, code, computers, datasets and instrumentation formed into a single virtual computing environment.
- Moving out of the laboratory into production

Grid Security Risks

- Identity Theft
  - Theft of user credentials, passwords
  - Very hard to detect and counter
  - They look like your users!

- Remote Exploits
  - “Traditional” form of network attacks

- Local Exploits
  - Potentially most damaging
  - Attacker already on system

Prioritizing Risks

- **Good network security environments**
  - **Identity Theft is most common attack**
  - **Remote Exploits less common**
  - **Local Exploits are rarely seen without the other two**

- **Inadequate network security environments**
  - **Remote Exploits are most common**
    - Example: windows based worms attacking unprotected PC's on the internet
  - **Identity Theft less common than Remote Exploits**
  - **Local Exploits initiated as "inside jobs" least common**

Not Mutually Exclusive

- **Multiple attacks used**
  - A successful attack results in the host being controlled by the attack
    - **ID Theft, Remote and Local exploits all used**
  - Typically the final step is some form of local exploit to gain admin access
    - **Example: Identity theft -> local exploit**
  - After machine has been taken over
    - **Identity theft tools used for wholesale account harvesting**
    - **Vulnerable services are backdoored to allow remote access**
- **Hackers adapt to the environment by looking for path of least resistance**
  - When Remote Exploits are difficult, ID theft is used
  - When Remote Exploits are easy, automated tools can be used for widespread attacks
  - Hackers adapt existing tools to new purposes
- **“Arms Race” against hackers**

Grid Risks: Identity Theft

- Grid has potential for a worldwide “single sign-on”
  - Uses x509 certificates
  - Any place that trusts your certificate will allow you to login
  - What if a bad guy gets control of your certificate?
    - They have stolen your identity and can access anything that trusts your certificate

ID Theft: Long Term Certificates

- **Properties**
  - Good for roughly 1 year
  - Private key is encrypted using passphrase
  - Typically stored in user's home directory
  - Used to generate proxy (short term) certificates
- **Issues?**
  - Vulnerable to trojaned certificate management binaries and keystroke loggers
    - ⇒Variations of these attacks have already been used against SSH
  - Infrastructure for handling revoked certificates typically half-baked
    - ⇒Revocation is manual process, and relatively few clients check revocation lists
  - Users cannot be depended upon to properly manage them
    - ⇒File system permissions may be inadequate
      - → FermiLab discovered during audit that 5% of ssh keys had incorrect permissions
    - ⇒Users may use trivial (or null) passphrases for convenience

ID Theft: Short Term Certificates

- **Properties**
  - Generated from Long Term Certificates
  - This certificate is what is actually used in authentication
    - ⇒ Possession of this certificate/key is sufficient to access Grid services
  - Good for typically a few hours to a few days
  - Protected only by filesystem permissions (no crypto)
    - ▪ Must be this way to ensure usability
  - By default, is stored in shared access /tmp directory
- **Issues**
  - Can be easily harvested with stolen administrative privileges
    - ▪ No need for passphrase to decrypt
    - ▪ Same vulnerability as Kerberos Tickets
      - ⇒ Hackers have stolen kerberos tickets and misused them already: method is known
    - ▪ Cannot enforce proxy lifetime with default tools
    - ▪ Poorly protected certificate may be good for an entire year!
  - No way to revoke a short term certificate without revoking long term certificate
    - ▪ Even if long term cert revoked, it is unclear if relying sites will notice due to spotty certificate revocation procedures

ID Theft: Recommendations

- **Activate Certificate Revocation support on client machines!**
  - **If a certificate is compromised, you want to know IMMEDIATELY**

- **Set standards for timeliness of certificate revocation**
  - **Parties responsible for revoking certificates should be operating at the standard of operational security staff**

- **Set standards for reporting to Certificate Authorities that a certificate may have been compromised**
  - **Compromises at a single site can have very far reaching effect**
  - **Environment must be fostered that promotes cooperation between sites for collective security**
    - Culture of fault-finding creates disincentive for sites to report and sabotages collective security

ID Theft: Recommendations

- Avoid long term certificates
  - If you have to use long term certificates – they have to be centrally managed by a professional staff
  - Use MyProxy or similar service if long term certs necessary
  - Eliminate long term certificates entirely
  - Educate users on proper certificate hygiene

- Manage short term certificates better
  - Filesystem permissions not really inadequate protection
    - Perhaps a kernel credential cache
  - Create an OCSP framework for real time revocation of proxy certificates
  - Add policy language to proxy certs so that they can only be used only for specific purposes
    - Example: a cert may only be used for file copying and not shell access

Grid Risks:Remote Exploits

- **Properties of Grid Services**
  - **Grid services must be on the network**
  - **Any network service is a potential target of remote exploits**
  - **Grid software can be distributed over the network and then run at remote sites**

- **Issues**
  - Firewalls must be opened up
    - **For specific services on fixed ports – GridFTP, GateKeeper, MDS**
    - **For temporary daemons on ephemeral ports**
  - How can you be sure that traffic coming in on opened port is legitimate?
  - How can you trust the code that is being sent over?

Remote Exploits: Recommendations

- Restrict ephemeral Grid connections to specific port range
- Use network intrusion detection tools to monitor network
  - If another protocol is using Grid ports, it should trigger response
- Patch systems!!
- Perform good system administration.
- When possible, use "safer" environments like a Java Virtual Machine or some sandboxing method (like chroot)
- Be involved with Grid development community.

Grid Risks:Local Exploits

- Local exploit risks for Grid are not significantly different from the "background radiation" of normal local exploits
  - Protecting against local exploits is generally harder to accomplish
- Harden kernels to control
  - Stack overflow attacks
  - Limit ability to perform privilege escalation (cannot setuid)
  - Block access to devices that allow reading/modifying memory directly
  - Block loading of kernel modules
- Run dubious processes in a sandbox
  - Chroot, CHOS, virtualized servers, etc…
- Keep machines up to date on patches
- Set more restrictive file permissions
- Deploy centralized syslogging

Incident Response

Stephen Lau

Definitions

- Computer Security Incident
  - Any event that may have resulted in a *potential* violation of existing policy.

- Incident Response
  - Actions related to handling of a computer security incident.

When Things Go Very Wrong (and they will)

- Prepare, prepare, prepare
  - Have procedures in place beforehand
  - Educate staff and team members beforehand
  - Have recovery mechanisms in place
  - Know what is “normal”

- Goal: Evaluate incident, contain it and return to operating state as soon as possible

- **NOT** the time to “fix longstanding issues”

When Things Go Very Wrong

- **Communication**
  - **Provide means to report incidents 24/7**
  - **Out of band communications essential**
    - Encrypted email / Phone call backs
    - Be aware of social engineering
  - **Limit communication to only those who “need to know”**
    - Essential in initial stages
  - **Keep a log of** ALL **communications and actions**
    - Necessary if legal action taken
  - **Be aware of information released**
    - Privacy issues

When Things Go Very Wrong

- **Initial Response**
  - **Determine if there is an incident or not**
    - Not all “incidents” are incidents
    - Helps if you “know your network”
  - **Collect information before taking action**
    - “Running down the halls” is counterproductive
    - Collect data via multiple methods if necessary
  - **Attempt to preserve as much data as possible**
  - **Limit amount of people involved to as few as possible**

When Things Go Very Wrong

- Containment
  - Prevent further damage
  - Ensure other systems are not vulnerable to attack
    - Possibly scan other systems for same vulnerability
  - You may want to preserve as much information as possible
    - See “evidence issues”
  - Limit amount of people involved if possible
- Recovery
  - Ensure that system will not be affected again
  - Restore from backups or reinstall
    - Did you prepare beforehand?

When Things Go Very Wrong

- **Post mortem**
  - **Evaluate incident response**
  - **Determine the vulnerability**
    - Fix process if necessary
  - **Ensure other systems are not vulnerable**
  - **Document entire incident, the response, and resolution**
  - **Ensure preservation of evidence, if necessary**

Real World Example: Protecting SCinet

Stephen Lau

SC2004

PITTSBURGH PA NOVEMBER 6 - 12

SCinet

SCinet

- SC Conference high speed network
  - Created by a dedicated team of volunteers
  - http://scinet.supercomp.org/

- We have no control over hosts and don't even know what is going to be shown!

- Many systems and applications are prototypes.

SC2003 Staff

SCinet

- An “open” network
  - No firewall.

- Diverse user base
  - Attendees, exhibitors, researchers
  - Industry, academia, government

- Diverse network
  - Exhibit floor
  - Extensive wireless coverage
  - Conference infrastructure (registration, show offices)
  - Educational rooms

Problem

- Ensure SCinet remains functional through the show.
  - Conference only runs one week!

- Threats
  - Outsider attacks
  - Clueless exhibitors and attendees
  - Crazed demos

Security within SCinet

- Policy
  - SCinet takes security *very* seriously.
  - Exhibitors handed policy document that is revised on an annual basis due to “lessons learned”.

- Security is built into the process.
  - Planning considers security “upfront”.
  - Not tacked on as an afterthought.

Defense in Depth

- Perimeter Defense
  - Bro and mon IDS
- Network subdivided based on function
  - Allows for filtering based on function
  - i.e. Exhibit show floor, Conference infrastructure
- Traffic filtered in some instances
  - Wireless filtered
    - Mainly to protect the wireless from itself

Defense in Depth

- User Education
  - SCinet Help Desk
  - Security demonstrations, i.e. Cube of Doom, password display
- Host Level
  - Ability to locate hosts
    - Wireless and wired
  - Ability to “jail” obnoxious hosts.
- Physical Security
  - Primarily to protect SCinet assets - (N > 10) million dollars in assets
  - SCinet access restricted and protected 24x7, before, during and after conference.

Incident Response

- Core of several security professionals.
- Not able to conduct complete “incident response”.
- Attendees responsible for own systems.

- Goal to locate and identify hosts threatening SCinet.

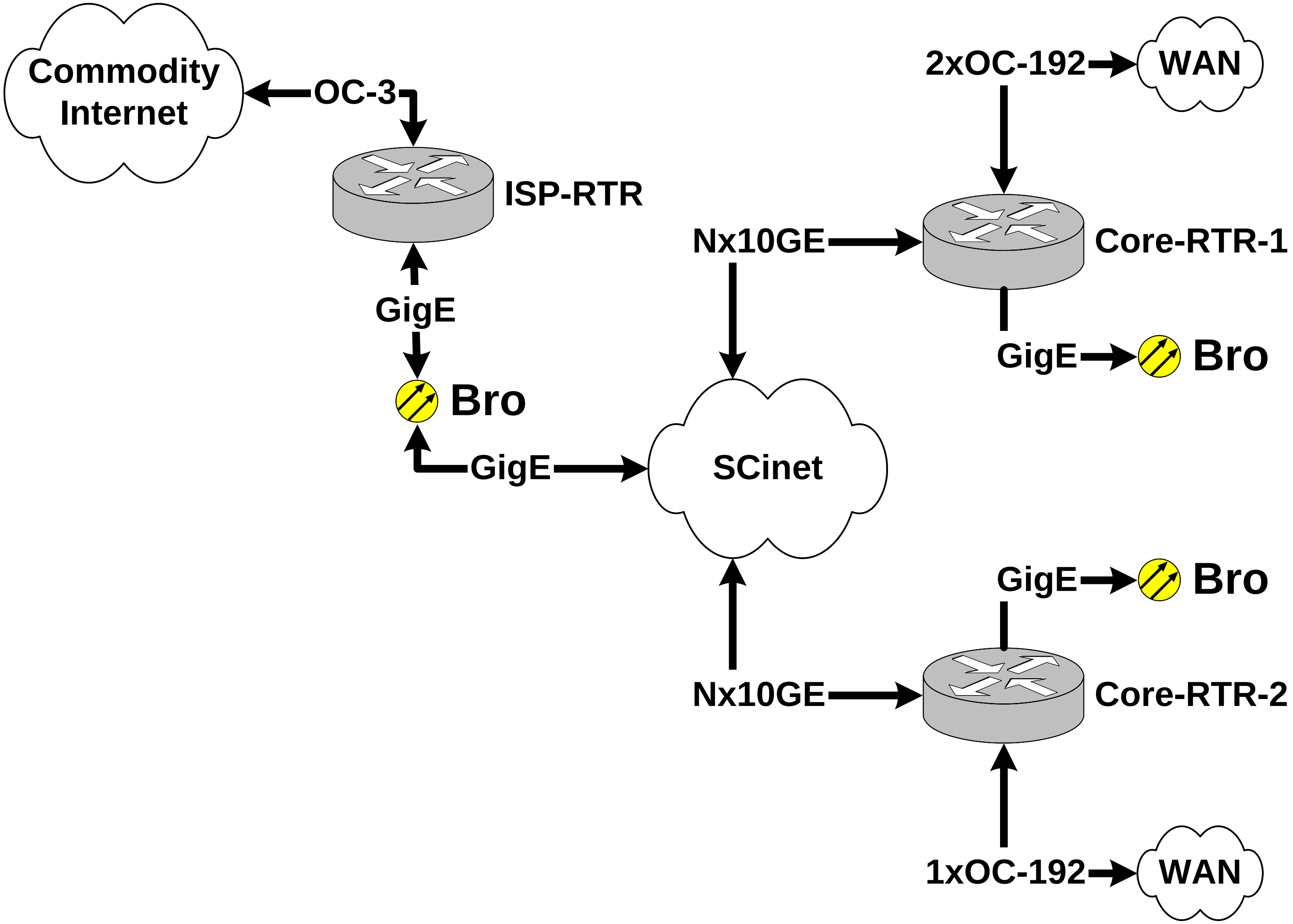
SC2003 Security Incidents

Linux Root Compromises	1	Solaris Root Compromises	1
Accounts with NO passwords	235	Clear Text Root Logins	32
Welchia Infected Systems	63	Repeat Welchia Infected Systems	2
Slammer Worm Infected Systems	6	System Infected with Other Windows Worms	10
Rogue Access Points	1	Rogue Ad-Hoc Wireless	10
Inbound Scans	1118	Inbound Directed Port Scans	60
External Nimda Probes	912	Complaints about Password Display	5
Repeat Root Compromises	2	Complaints about Spinning Cube	1

Bro at SCinet

- Bro primary IDS for SC conference since SC00
  - Used to monitor SCinet traffic
  - Detect 0wned systems
  - Ensure conference network does not get taken down by attacks
- Maximum observed bandwidth
  - 23 Gbps at SC2003 (Bandwidth Challenge)
  - Used router hardware BPF
- Passive monitoring only
  - Automatic countermeasures disabled
- Educational tool for attendees
  - Password capture and display
  - Alert exhibitors to “risky behavior”
    - i.e. .rhosts with root enabled

SC2003 Bro Infrastructure



SC2004

- Bro is primary IDS
- Added wireless capability to “jail” offending users and direct them to “de-worm” website
- Increase filtering of infrastructure network
- SCinet wide syslog capability
- Increase number of security demonstrations
  - Visit SCinet booth to view demonstrations

Summary

- Security doesn’t necessarily require large infrastructure investments.
  - Caveat: Need to design with security in mind.

- Open security models do work.

- Dedicated staff is essential.

RISK

James Rothfuss

No Such Thing as 100% Security

“A ship in harbor is safe, but that’s not what ships are built for.” [11]

"Maximum security is always a prison." (Mike Moxcey)

LBNL Model

$$$

- Starts with a review of cyber incidents to determine actual damage in dollars
- Depends on the best thinking and estimates of those responsible for protecting LBL cyber resources
- Calculates risk avoided and return on investment for protective measures

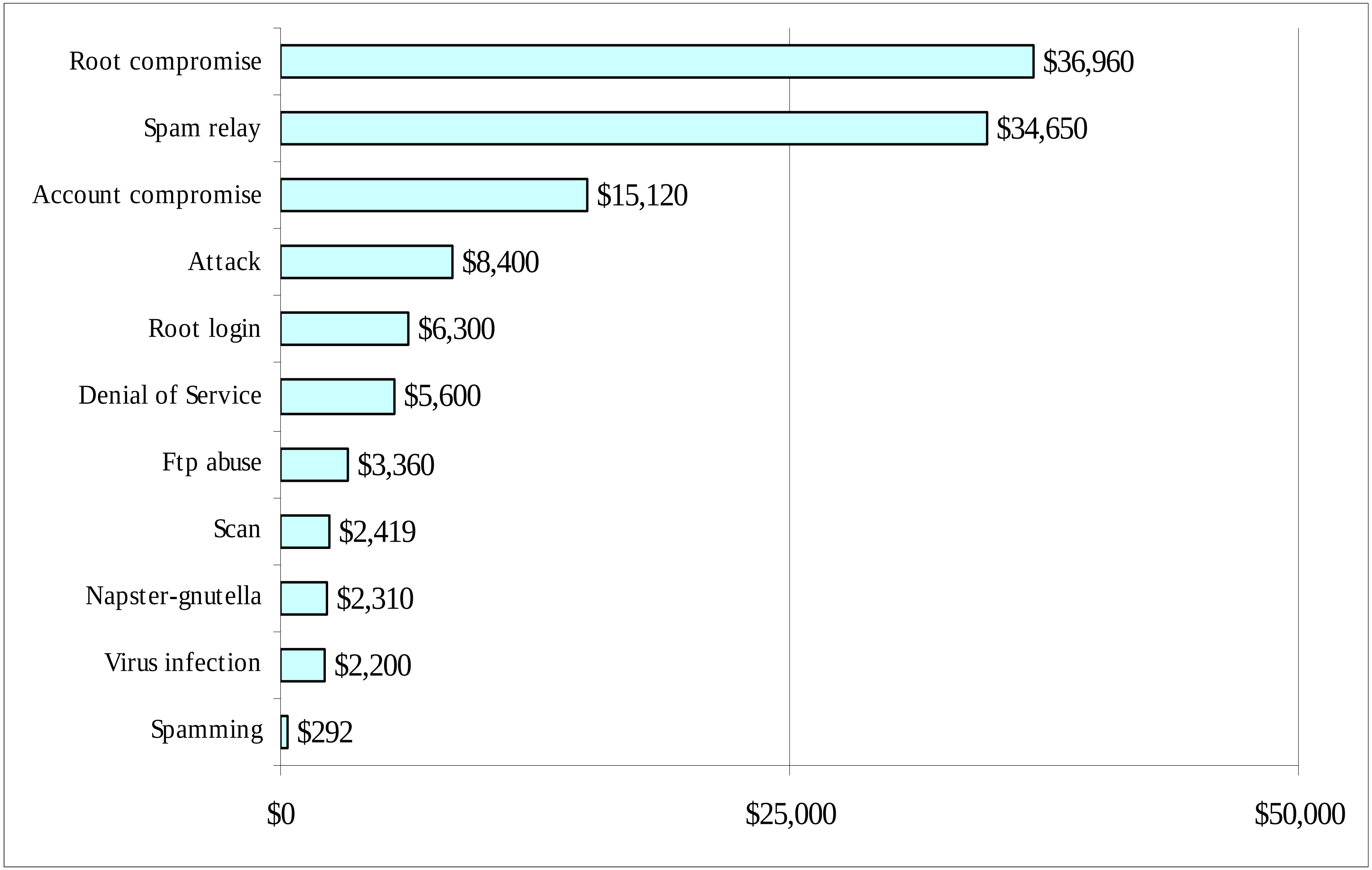
Examples of Cost Based Analysis

- **The next few slides show results of using the methodology to**
  - **determine the nominal, probable, and possible damage of different cyber incidents**
  - **calculate the cyber damage avoided**
  - **evaluate the cost effectiveness of individual protective measures**

Nominal 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**

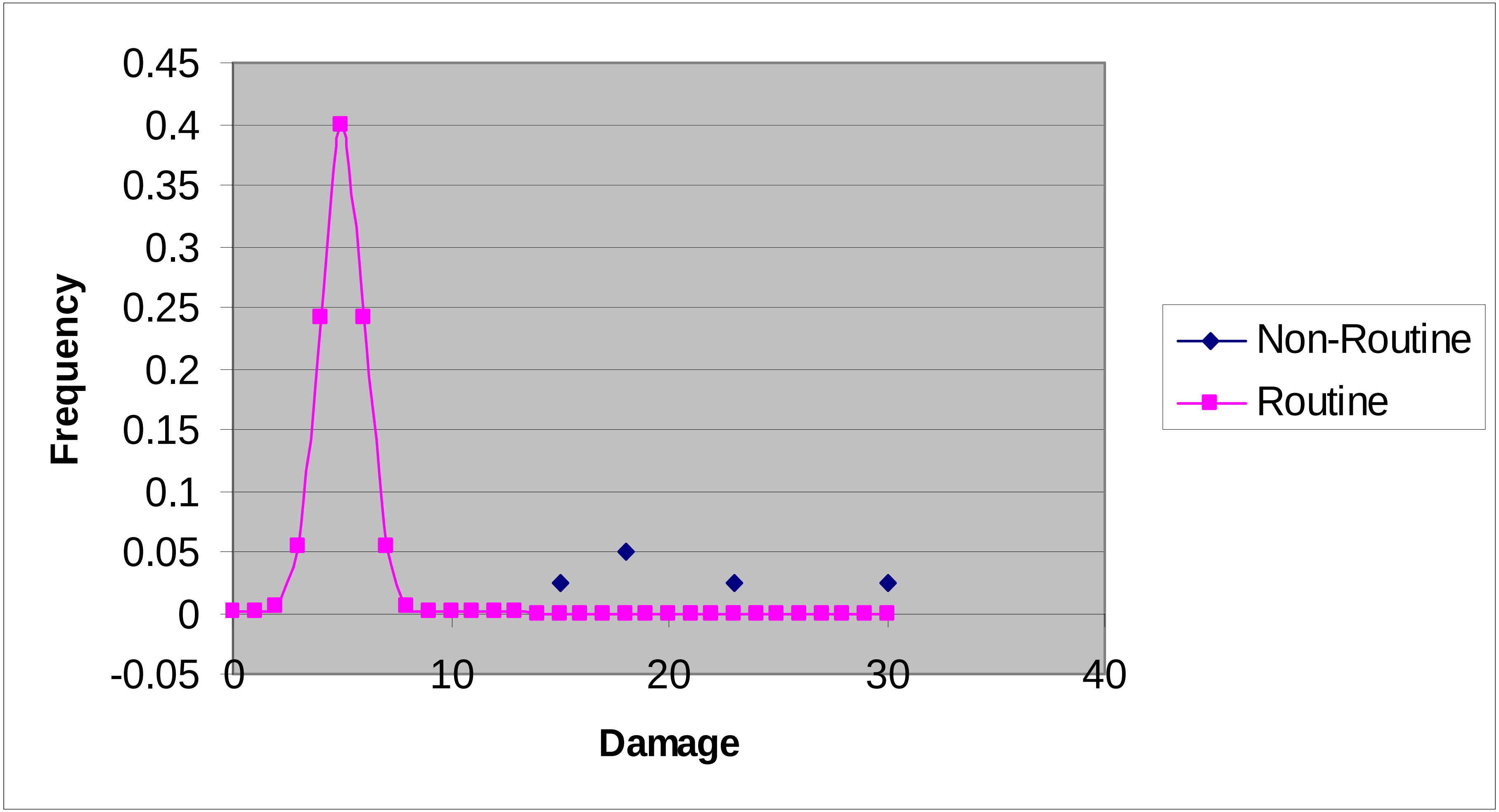
Nominal Damage From Cyber Incidents (FY 2000)



Probable Damage
Associated With Incidents

- *Probable Damage* **includes a factor for** *non routine* **incidents (none in FY 2000)**
  - **LBL Security Managers agreed 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**

Non-Routine Incidents



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 Damage, Probable Damage, and Damage Avoided

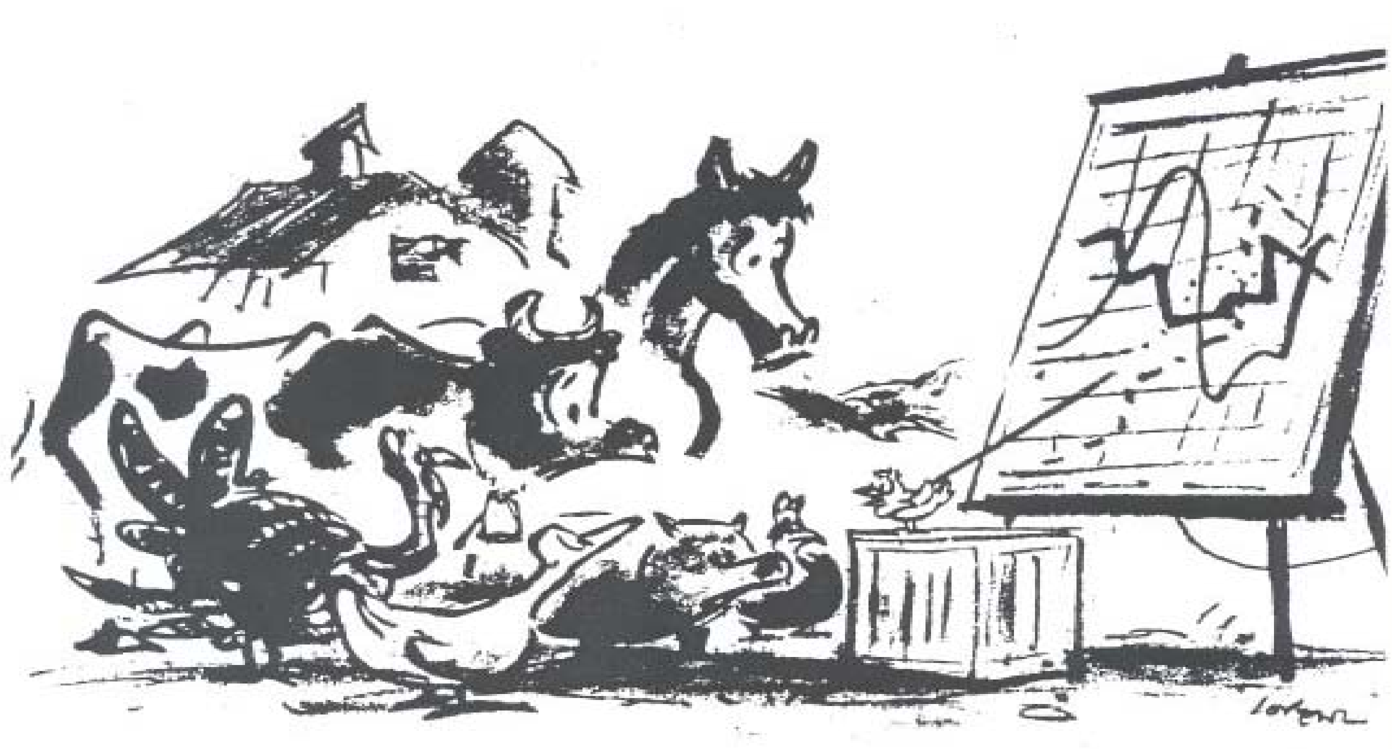
Incident Type	*Probable damage per hit*	*Total Unblocked Attacks*	*Total Blocked Attacks*	*Probable damage per year*	*Damage avoided per year*	*Total possible damage per*
Account Compromise	**$5,778**	**9**	**1490**	**$51,000**	**$8,607,000**	**$8,658,000**
Attack	**$2,407**	**12**	**633**	**$28,000**	**$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***

Protective Measures with Estimated Effectiveness

CounterMeasure	***Account compromise***	***Attack***	***Denial of Service***	***Ftp abuse***	***Napster-gnutella***	***Root compromise***	***Root login***	***Scan***	***Spam relay***
Warning banner program					2%				
Regular password cracking	10%			10%		10%			
LBNL firewall programs	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%
Level 1 vulnerability scanning	50%		50%	50%		50%			50%
BRO Intrusion detection sensors and analysis infrastructure	95%	95%	95%	95%	95%	95%	95%	95%	95%
"Crown jewels" intrusion detection program	40%	40%	40%	40%	40%	40%	40%	40%	40%
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%		

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%
Level 1 vulnerability scanning program	$35,000	$332,359	850%
"Crown jewels" intrusion detection program	$7,280	$263,930	3525%
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%



“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.”

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. 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)

- 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)

- **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

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`
- **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`
- **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`