Computer Protection at Open Scientific Facilities

- Presenters
  - Vern Paxson
    - ICIRI/LBNL
    - vern@{icir.org, ee.lbl.gov}
  - William T.C. Kramer
    - NERSC/LBNL
    - wtkramer@lbl.gov
  - James Rothfuss
    - LBNL
    - jsrothfuss@lbl.gov
  - Stephen Lau
    - NERSC/LBNL
    - slau@lbl.gov

Table of Contents

Title	Length	Slide
Intrusion Detection: Principles & Practice	1 hr	3
Components of Good Protection for Open Scientific Facilities	:30	32
Network Equipment Tracking System	:30	48
Deployment in an Open HPC Environment	1hr	66
Bibliography		91

Intrusion Detection: Principles & Practice

Vern Paxson
ICSI Center for Internet Research (ICIR)
International Computer Science Institute
and
Lawrence Berkeley National Laboratory
University of California
Berkeley, CA
vern@{icir.org, ee.lbl.gov}

A look at Network Intrusion Detection

- Why network intrusion detection? Why not?
- Styles of approaches.
- An example of a NIDS: BRO.
- Detecting activity: sniffers, stepping stones, backdoors.

LBNL is a large, diverse, network-intensive facility

- Each day, about 4,500 Lab computers communicate with 50–60,000 remote computers, using more than 100 distinct applications.

- Each day, about 100 remote hosts try to find a vulnerable Lab server to break into.
  - (Well, it's actually 300–2,000 remote hosts, thanks to CR + Nimda.)

- Without our network intrusion detection system, *Bro*, the *reactive firewall* it provides, and the Lab's *proactive scanning*, we would have break-ins *every day*.

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, con't:

- Appealing because it's *cheap* and gives broad coverage.
- You can have <u>multiple</u> 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.
- Assumes network-oriented (often “external”) threat model.
- For high-speed links, monitor may not be able to 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, con't:

- 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.

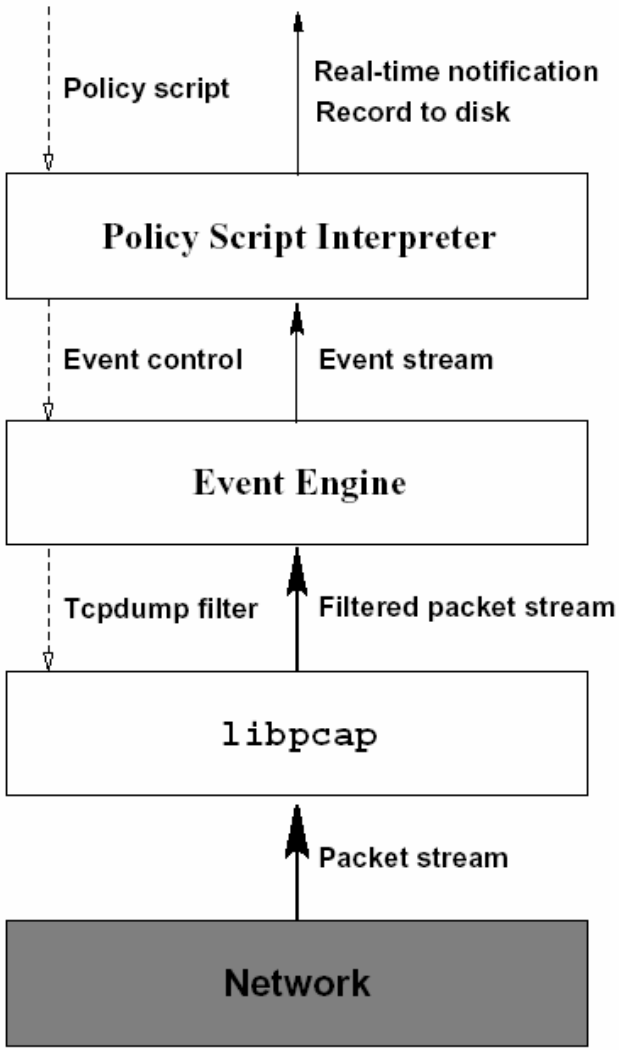
- Scaling works against us: as the volume of monitored traffic grows, so does its diversity.

- 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.
- <u>The monitor will be attacked.</u>

Bro Architecture



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
```

  - in first five months of operation,
    - 120 UCB break-ins (60 root compromises)

Teeth

- "`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, 100–200 scans dropped each day.
- Routers run with 1,000–4,000 ACL entries.
- When scan blocking fails, mean time to break-in: <u>4 hours</u>.

Status

- Operational since 1996.
- Two dozen monitor boxes deployed at LBNL, UCB, ICSI.
- LBNL boxes see about 2 billion packets/day (~23 Kpps).
- Avg: 500 filtered pps; peaks: 2,000+ pps.
- Connection logs: 1.3 GB/day.

Status, con't

- At LBNL/ICSI, "bulk trace" machines record some/all traffic for off-line analysis.
- Also monitor: NERSC, JGI, ESNET, internal nets.
- Break-ins: ~every 2–4 weeks at LBNL.
- 50,000 lines of C++. Unix/libpcap-based.
- 6,500 (generic) + 1,700 (site specific) lines of Bro scripts.

```
Oct 10 11:10:41 bw3-115pub202.bluewin.ch/3133 >
        winoto6.lsa.berkeley.edu/ftp
       #43 excessive filename:
       pppppppppppppppppppppppppppppppppp..[5115]..
Oct 10 12:23:15 ftp: 166.111.88.94/4188 >
       socrates.berkeley.edu/ftp #554
       SITE EXEC %020d|%.f%.f| (ok)
Oct 10 12:24:32 ftp: 166.111.88.94/1210 >
       vermi.berkeley.edu/ftp #559
       SITE EXEC %020d|%.f%.f| (502 SITE command not
Oct 10 12:27:34 dial-148-240-72-140.zone-2.dial.net.mx has tried
   100 username/password combinations
       (latest: brandon@cs3.cs.berkeley.edu)
Oct 10 12:34:38 dial-148-240-72-140.zone-2.dial.net.mx
       has tried 1000 username/password combinations
Oct 10 14:41:44 ftp: outbound.tumbleweed.com/7772 >
   conquest.ocf.berkeley.edu/ftp #3185
       RETR mypasswd (complete)
```

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 (!).

Detecting activity — sniffer detection

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

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

- Another form: sniffer detection.
  - 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 <u>idle periods</u> 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 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 surreptitously 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, con't:

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

- Algorithms also amenable to <u>filtering</u> for large perf. gain:
- e.g., `tcp[(tcp[12]>>2):4] = 0x5353482D and (tcp[((tcp[12]>>2)+4):2] = 0x312E or tcp[((tcp[12]>>2)+4):2] = 0x322E)`

Detecting backdoors, con't:

- 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.

U.S. Department of Energy

Office of Science

Components of Good Protection for Open Scientific Facilities

William T.C. Kramer

NERSC/LBNL

November 17, 2003

SC 2003, Phoenix, AZ

Agenda

- Components of good protection
  - Policy and Procedures
  - Good Systems Protection
  - Response teams
  - Other “best practices”

Primary Protection Concern

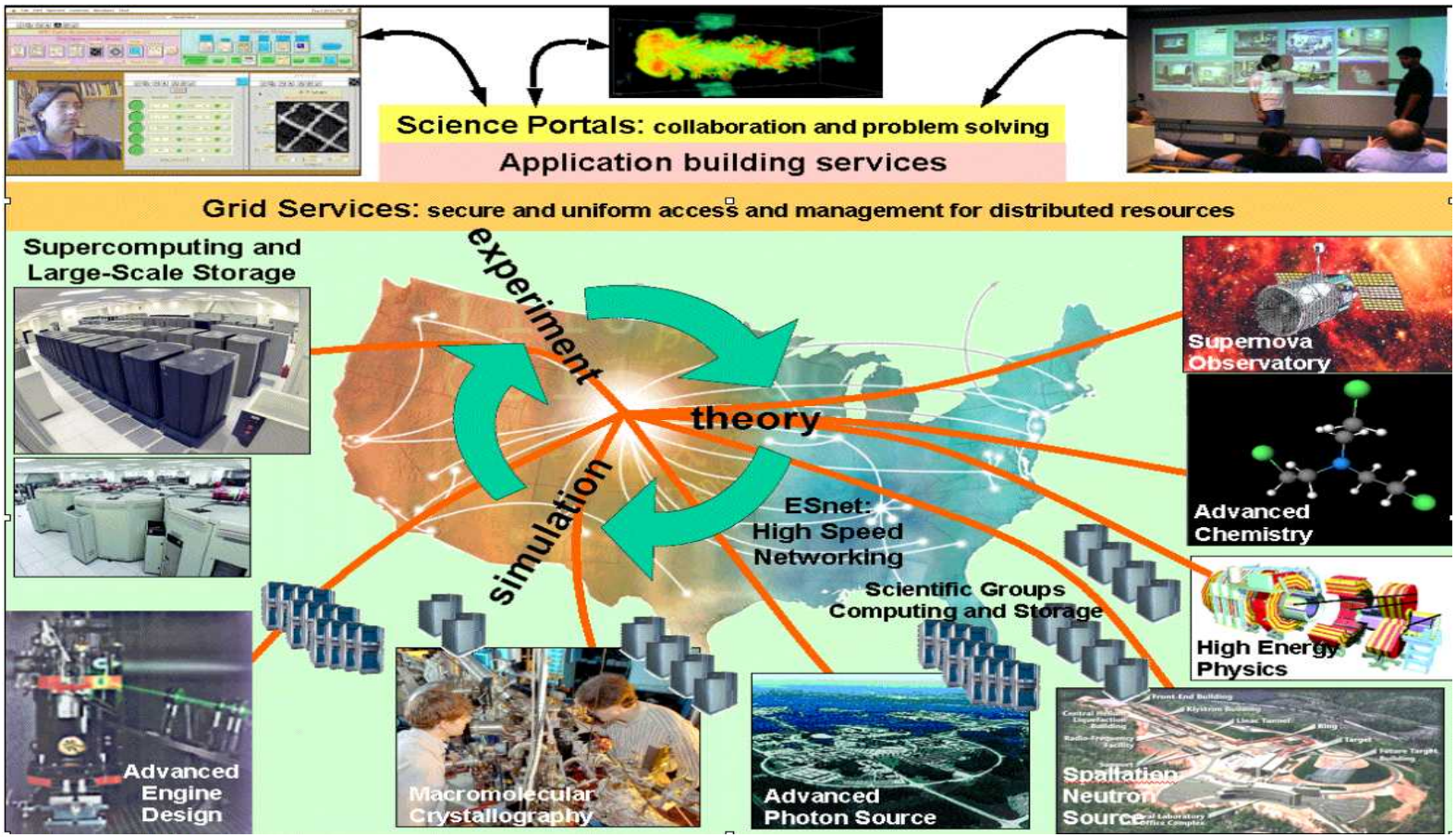
Service Protection

Information Protection

U.S. Department of Energy

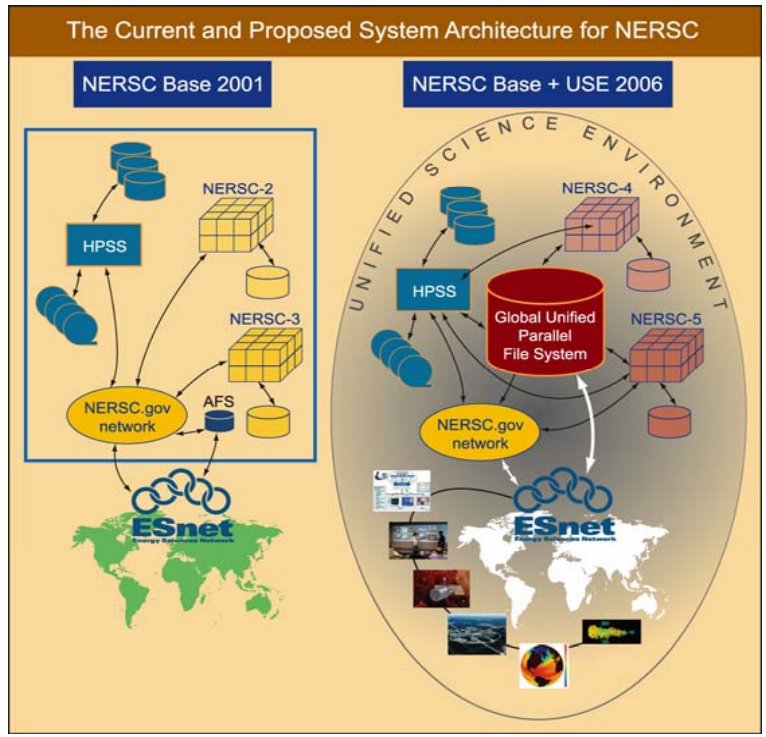
Office of Science

Everything is Integrated in the Future



What This Means

- There will be a lot more people and things “inside the walls”
- Cooperative processing will increase
- The patterns of access and connectivity will have much more variety
- The protection practices of the past may not work in the future



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

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

Good systems protection

- 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
  - 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
  - All of us have to push to change the mind set

Response teams

- Every site should have an incident response team for cyber protection
- Consists of
  - Cyber protection personnel (typically the technical lead)
  - Staff from all the major expertise groups
    - System Admin, Networking, application layers, …
  - Staff from all the major groups that have responsibility for protection
    - Any group running systems, any group developing s/w that run on systems, network groups, …
  - Management focal point
    - Formally responsible for the protection of the site
    - At a level that can allocate/get further resources
    - At a level that and deal with higher levels of management

Response teams

- Use the team in several ways
  - Reactively to deal with an incident when needed
  - For communication of protection/protection issues and ideas
  - As a way to assign responsibility to carry out policy and practice
  - For education of other staff
  - To advise on changes and strategy
    - To represent the viewpoints of others
  - To explain to other staff the reasons for policy and practice so the entire organization is protection aware

Relationships with Other Organizations

- Should know how to contact
  - Protection functions within your organization – both cyber and physical
  - Protection organizations in sponsoring agencies
  - Local law enforcement
  - FBI
  - Etc.
- Often organizations have a designated path to contact outside agencies
  - Example – in at least some government organizations only the Inspector General is authorized to contact the FBI.
  - There is often a pre-arranged relationship established
    - E.g. law enforcement probably will respond better to the protection staff they may have met rather than Joe ("grad student") Sysadmin calling.
- Need to remember the more people invited to the party means the more people looking around your house and maybe making a mess
  - Consider before inviting

Other "best practices"

- 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

Dedicated Staff

- Cyberprotection is a distinct discipline that is interdisciplinary in nature
  - Have parts of system administration, networking, coding, analysis, application understanding, operations
  - But also needs unique skills – forensics, policy interpretation and maybe creating , investigation, synthesis of information
- There has to time and effort devoted to protection
  - Need a clear responsibility assigned to led the effort
  - Need to commit specific resources
- Protection typically crosses organizational boundaries
  - Need authority to provide direction – not just advice
- Size of dedicated effort determined by scope, threat and policy

Getting and Keeping Good People

- Want people with a broad background
  - They should have done more than just protection at some point
  - Demonstrated flexibility
- People whose experiences map on to the sites mission and goals
  - Getting someone who has spent their entire career in highly classified work to deal with protection in an open environment (or vis a versa) may not be the best idea
- Consider hiring the most qualified – the most broadly qualified
  - That is not the highest level per se but the people who can be flexible
  - People who won't spend 100% of their time just doing the core activities
- Need good communicators

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
- Organized ways of discussing and addressing protection issues
- Excellent people with enough time to spend on protection

COMPUTING SCIENCES

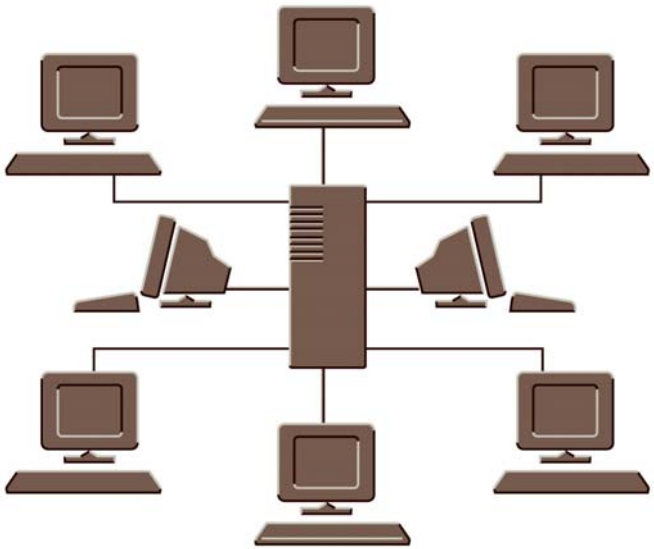
ITSD

INFORMATION TECHNOLOGIES & SERVICES DIVISION

James S. Rothfuss, Computer Protection Program

NETS

Network Equipment Tracking System



BERKELEY LAB

Network Equipment Tracking System

NETS Presentation

- Protection Concepts
- NETS Vision
- Current Implementation
- Future Development and Integration

Views of Protection

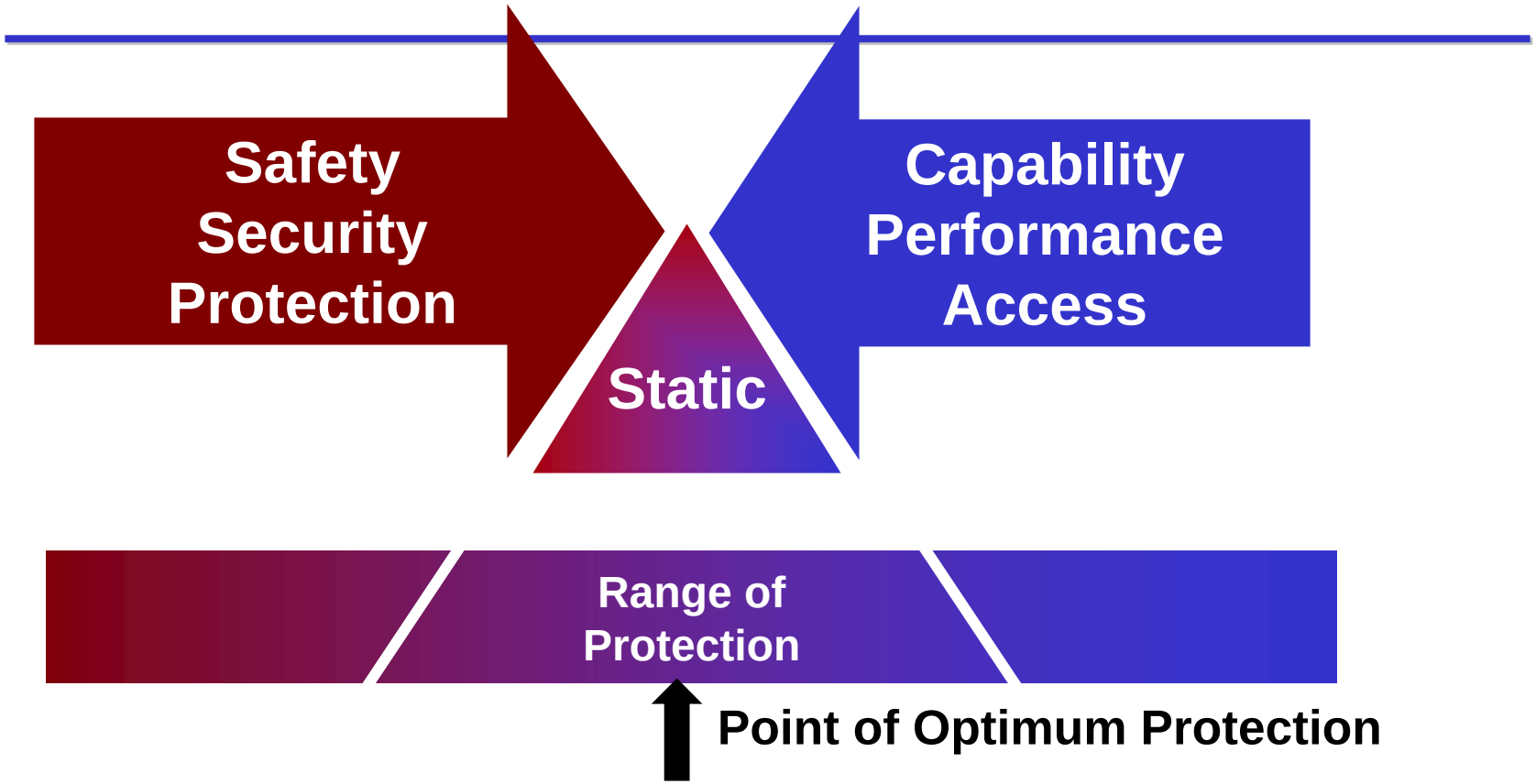
"Threat" Based Protection

"Vulnerability" Based Protection

Protective measures are based on the known attacks.

System weaknesses are identified and protected.

Current Methods of Vulnerability Based Protection



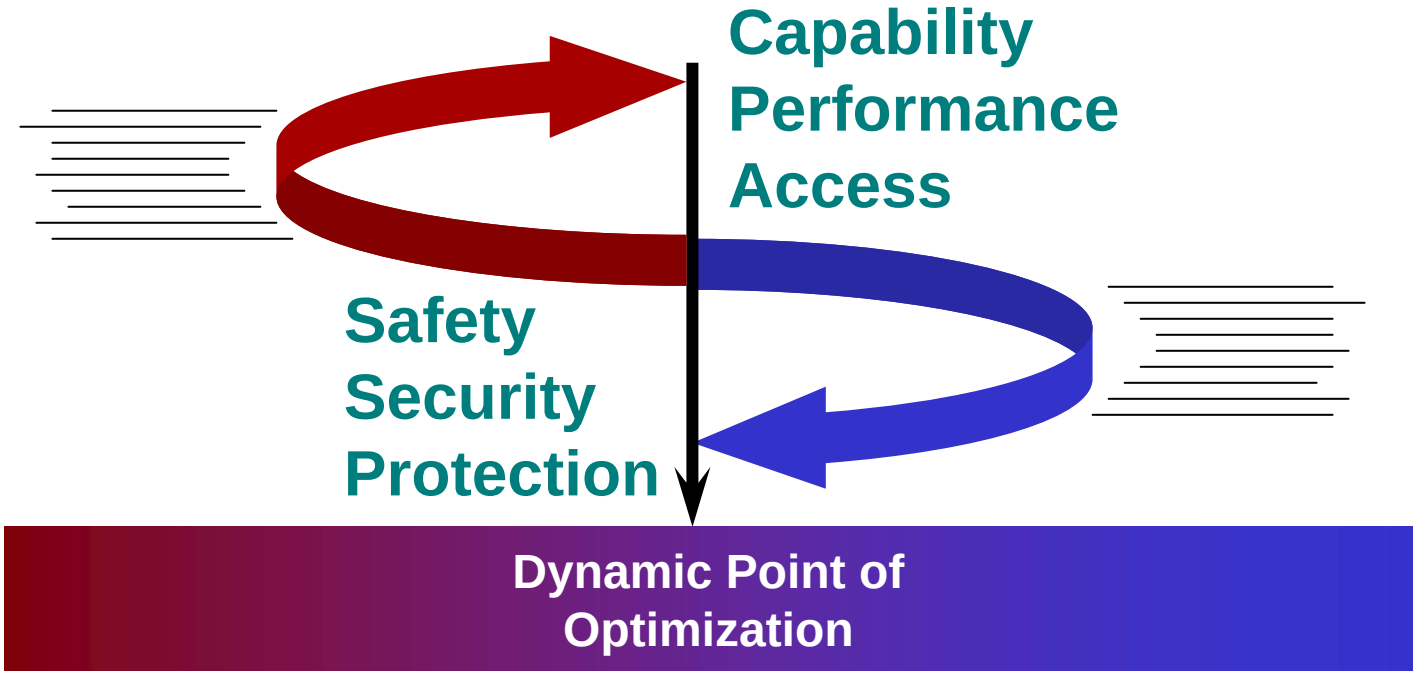
- Analyze network
- Guess at “reasonable” firewall rules
- Hope the rules stay current (assume a static network)

Current Methods of Vulnerability Based Protection

Patching Vulnerabilities

	Calendar Time	Effort
• Scan for Vulnerability	1/2 day	4 hours
• Notify System Admin	1/2 day	4 hours
• Wait	5 days	1 hours
• Scan for Vulnerability	1/2 day	4 hours
• Block	1/2 day	4 hours
Totals	7 days	17 hours

Continuous Optimization



- Constant analysis of network
- Protection measures adapt

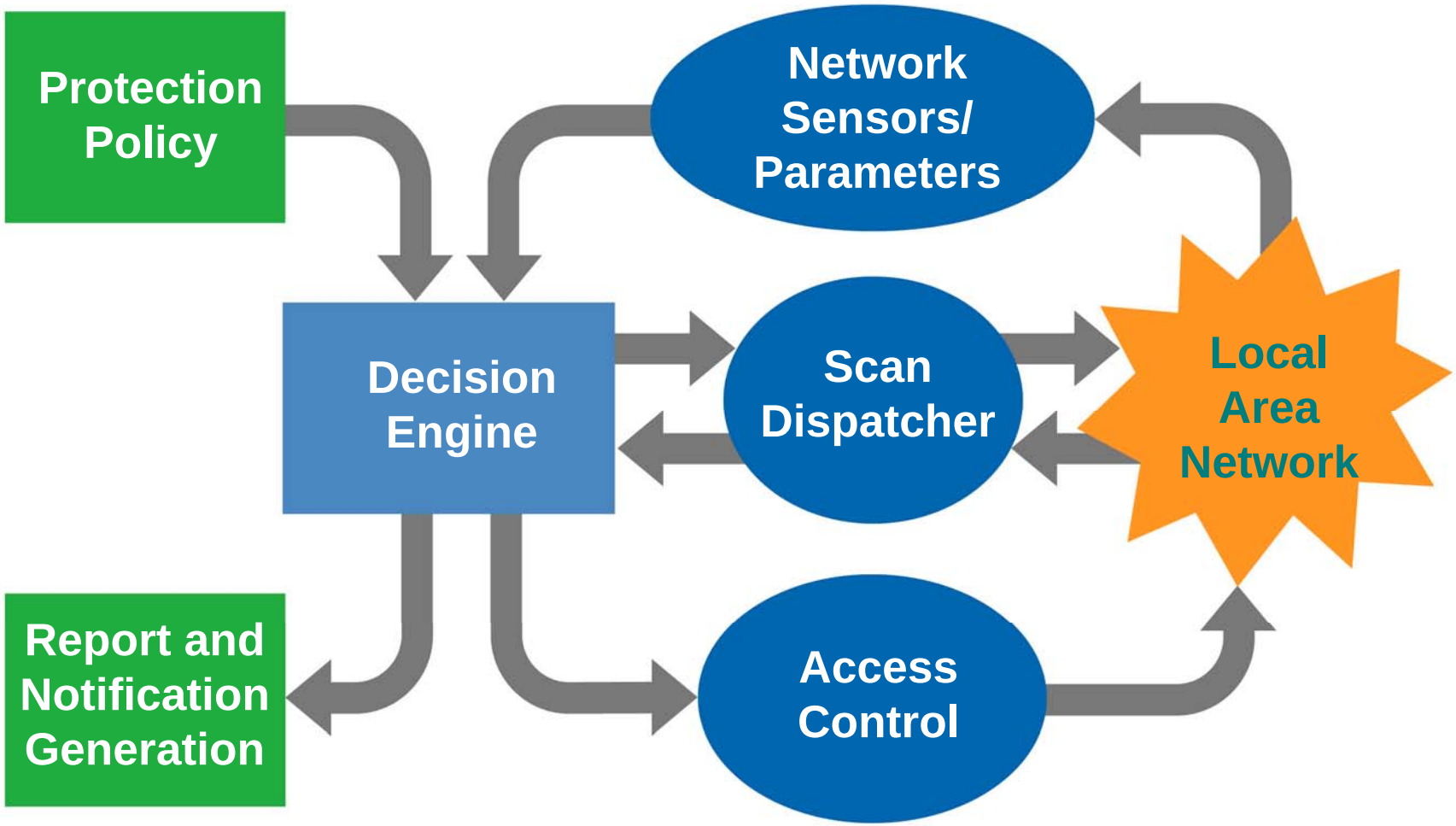
Optimum balance between protection and access

NETS Vision

Fully automated vulnerability discovery and elimination

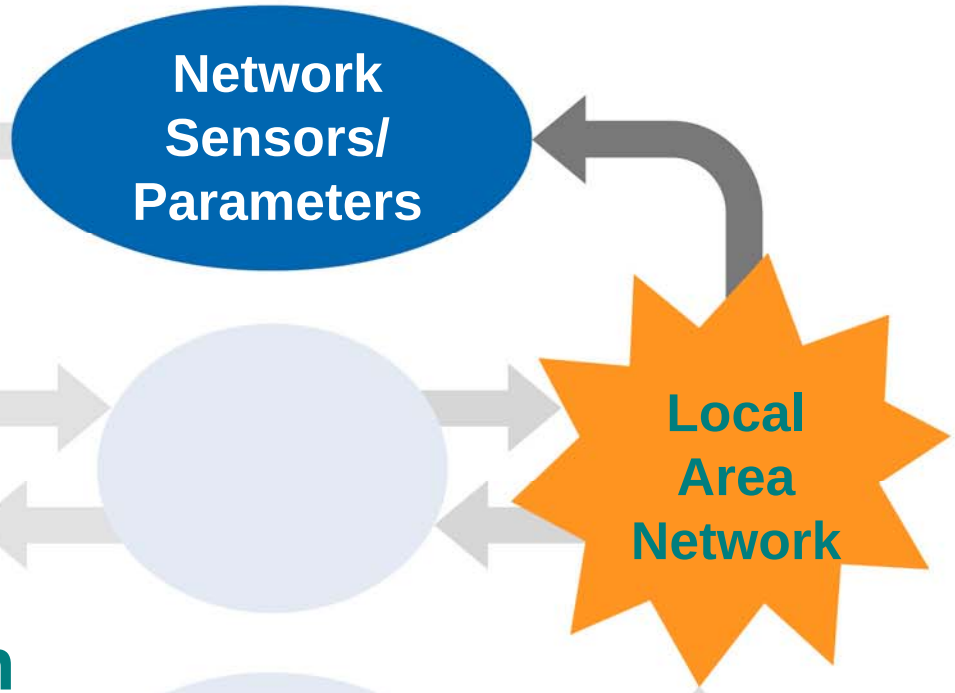
- Network information continuously collected
- System connections are detected
- New systems scanned
- Network vulnerabilities evaluated as they appear
- Vulnerabilities blocked
- System admins automatically alerted
- Automatic block removal as vulnerabilities are fixed
- Safe systems given full access
- Internet access is maximized

Adaptive Network Protection



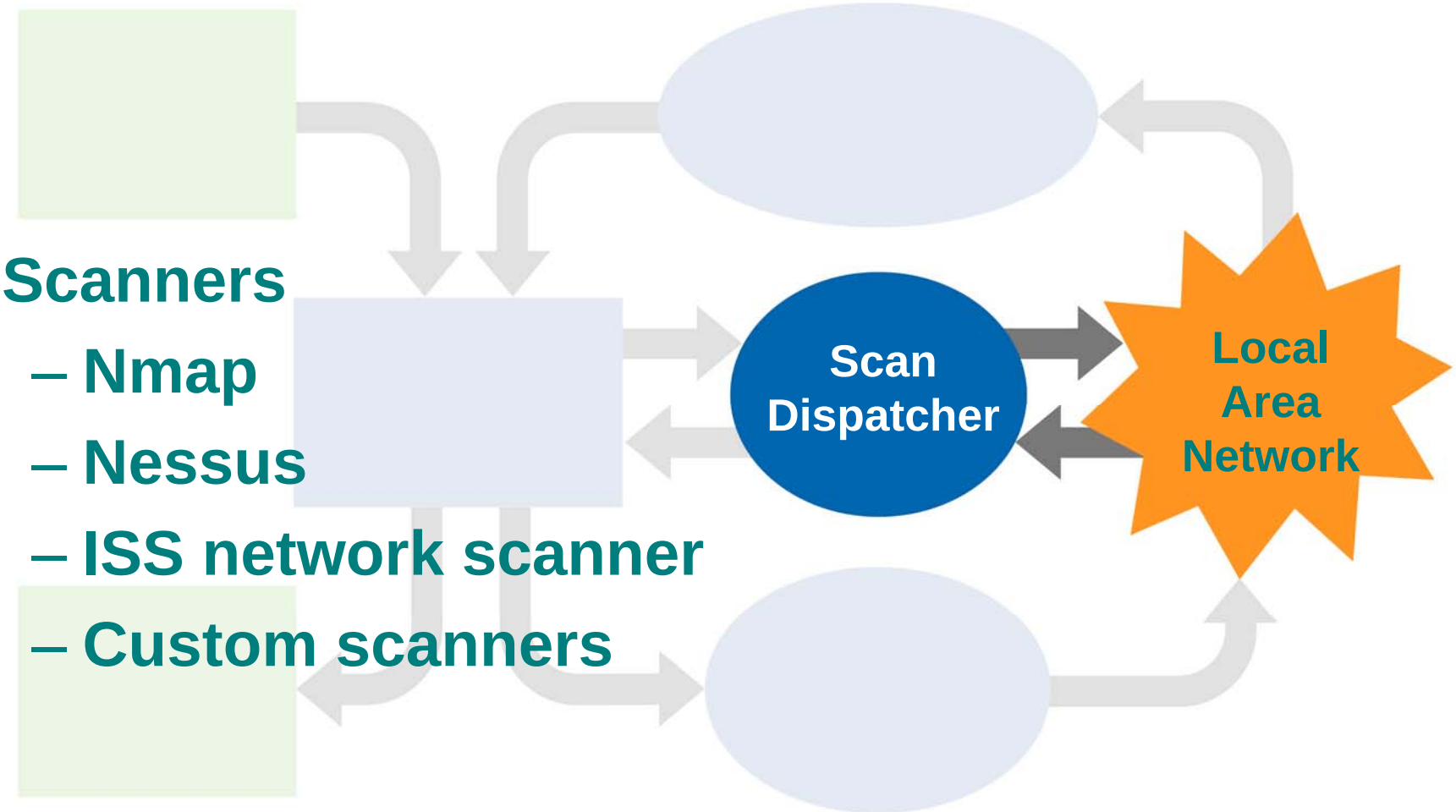
Sensors/Parameters

- DNS
- DHCP
- ARP data
- Router data
- NIS service
- Microsoft domain
- User input (registration)
- Network traffic
- Intrusion detection (Bro)



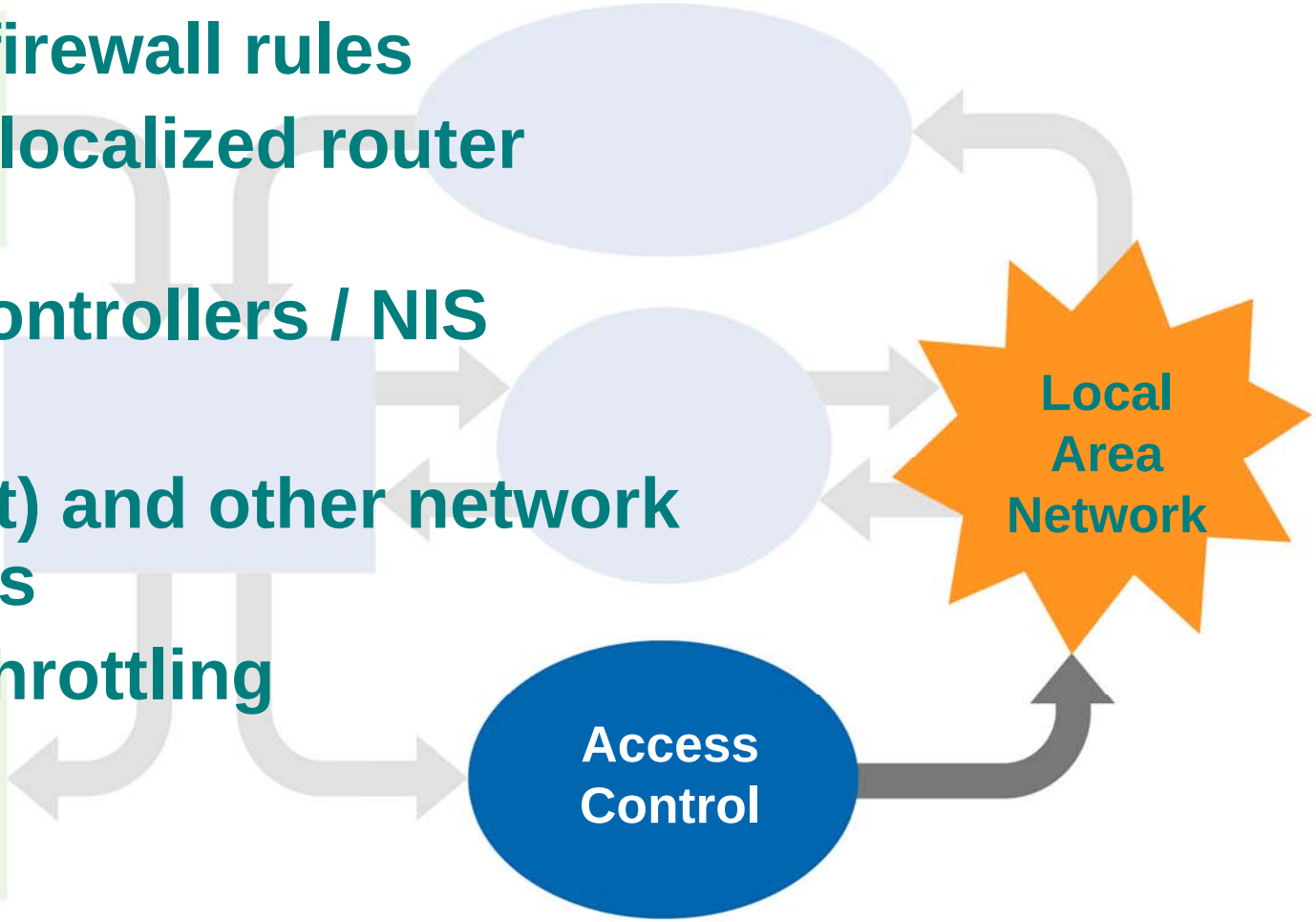
Probe Dispatcher

- Scanners
  - Nmap
  - Nessus
  - ISS network scanner
  - Custom scanners



Access Control

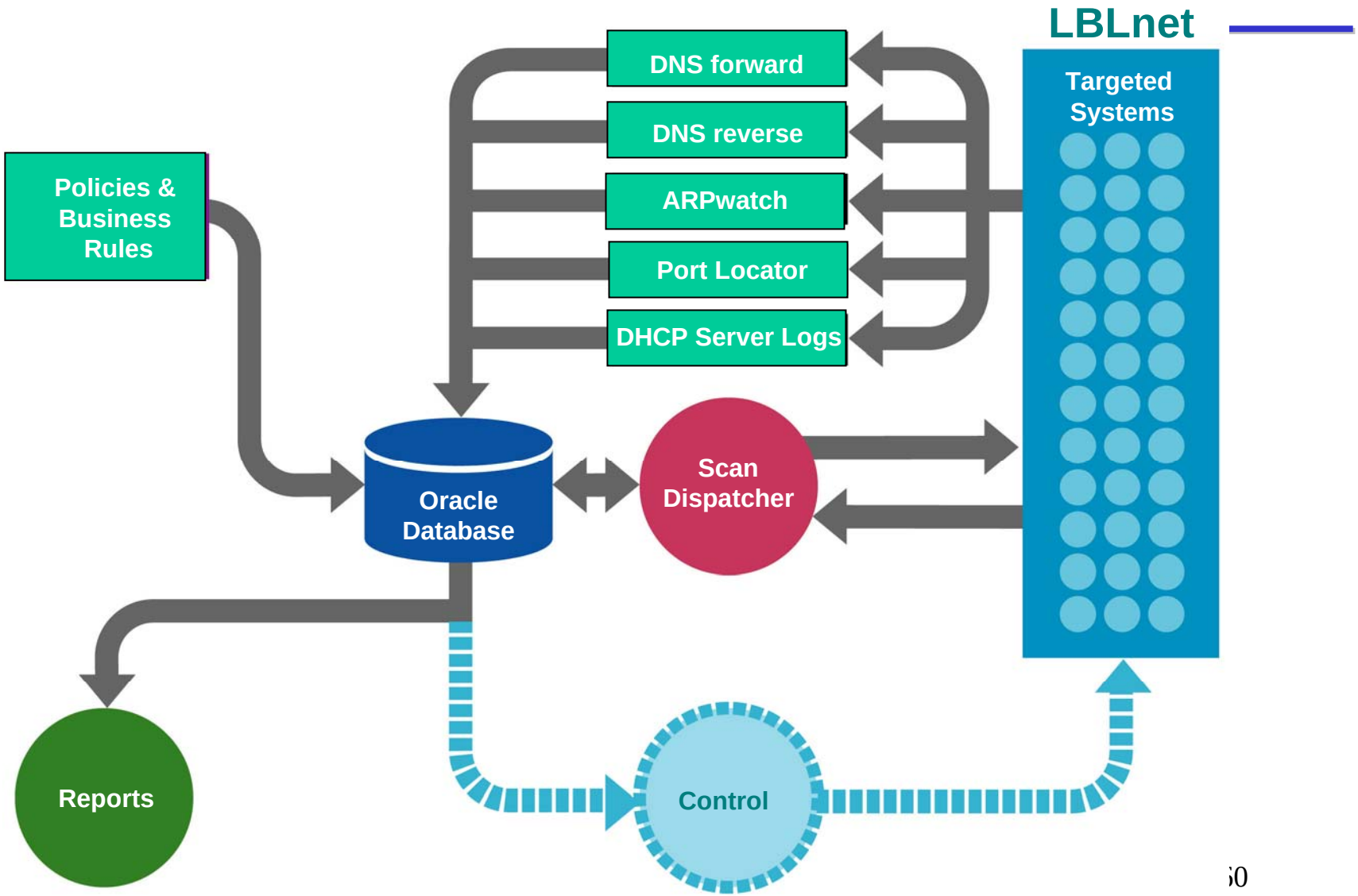
- Router access controls
- Dynamic firewall rules
- Blocks at localized router switches
- Domain controllers / NIS
- DHCP
- RST (reset) and other network commands
- Network throttling



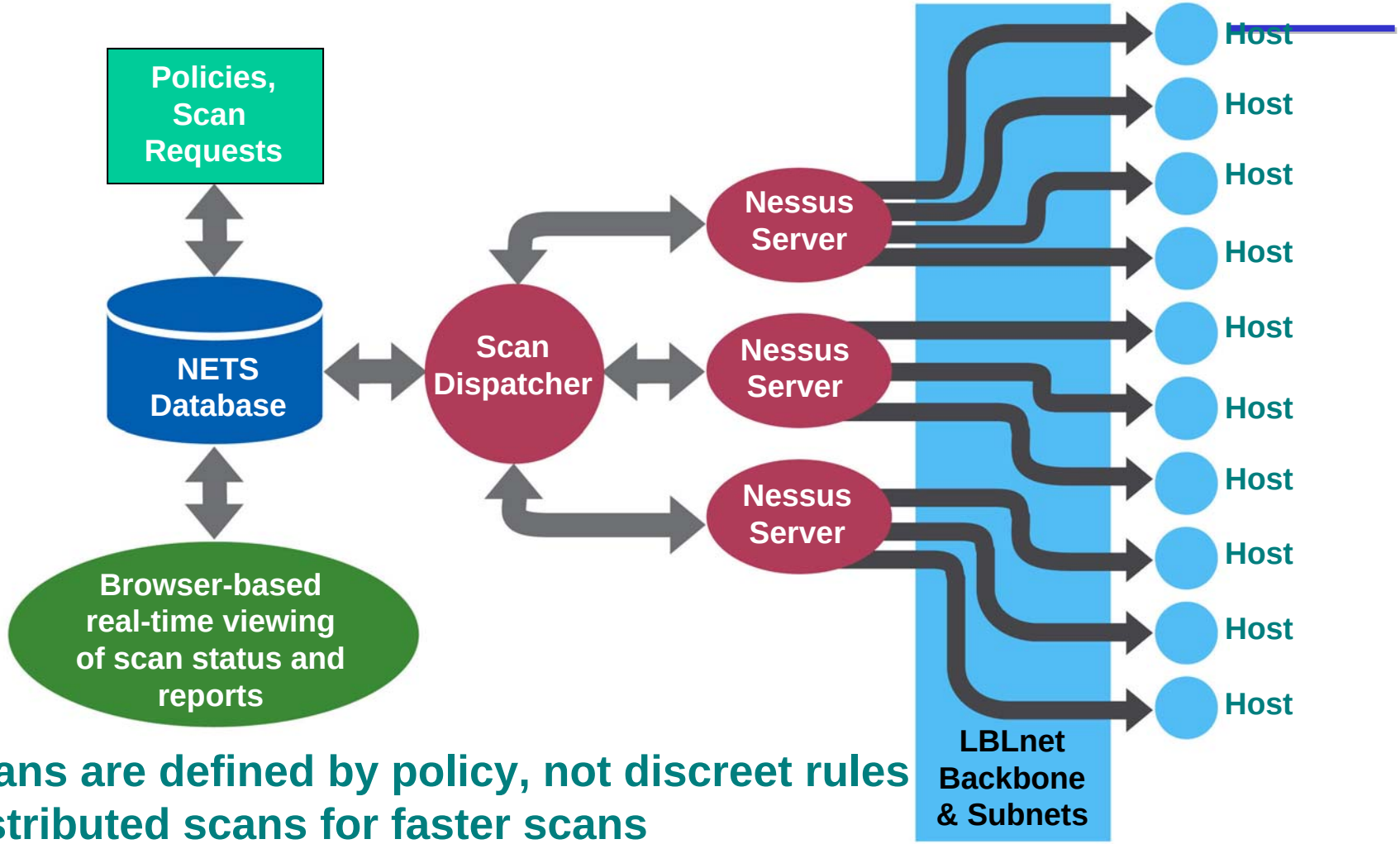
LBNL Network Equipment Tracking System (NETS)

- √ Network information collected
- √ System connections are detected
- √ Systems are probed
- o Vulnerabilities blocked
- o 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)

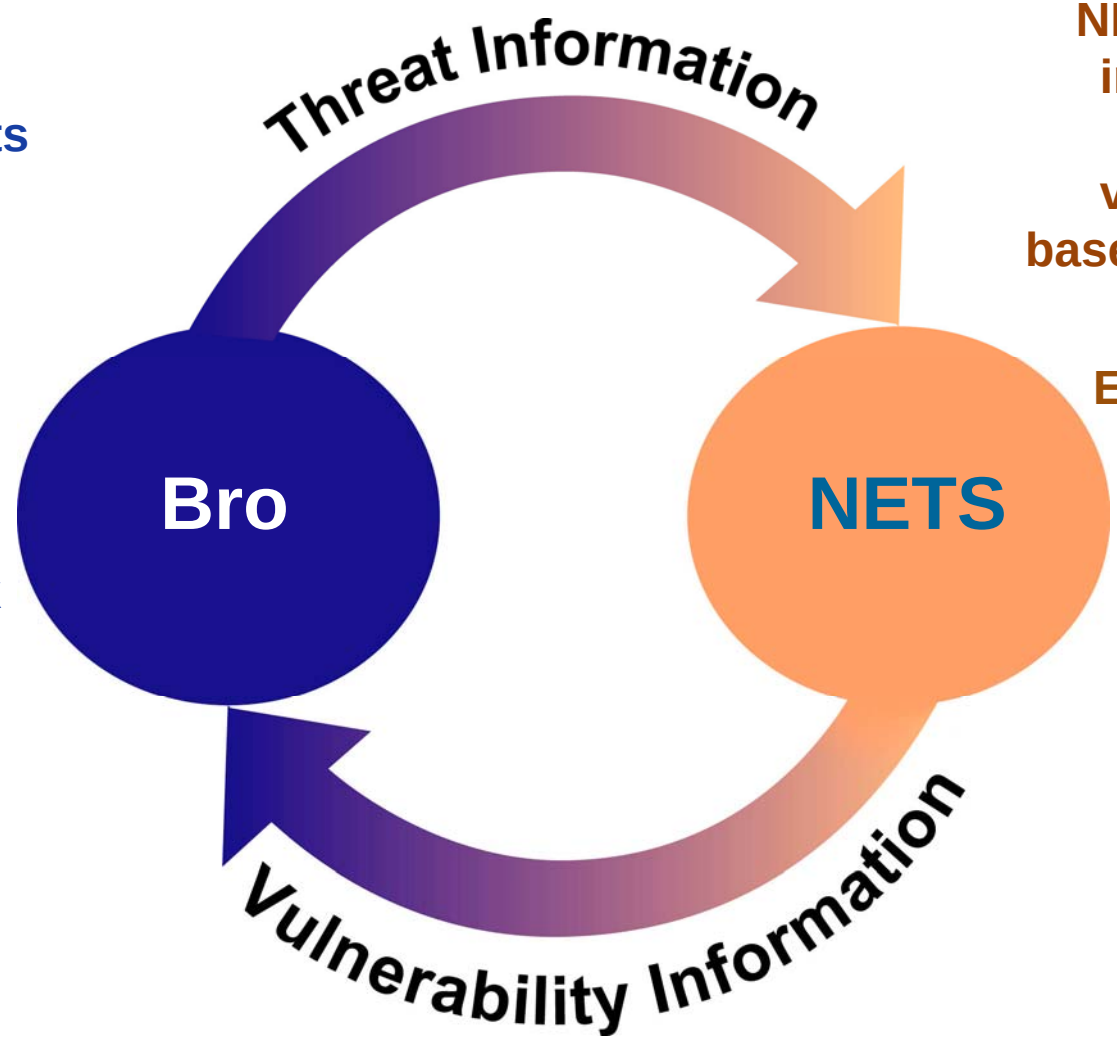
Future Advances

- **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

Future Integration With Bro

Bro uses NETS information to prioritize threats based on vulnerabilities

Extra attention given to vulnerabilities with a high risk of attack



NETS uses Bro information to prioritize vulnerabilities based a on threat

Extra attention to attacks against known weaknesses

Views of Protection

"Threat" Based Protection

"Vulnerability" Based Protection

NETS and Bro Integration

Network protection adapts based on both threats and vulnerabilities

U.S. Department of Energy

Office of Science

Computer Protection in Open Scientific Facilities Deployment in an Open HPC Environment

Stephen Lau

NERSC/LBNL

SC 2003 Phoenix, AZ

Overview

- Deploying security in an HPC environment
  - Defense in Depth
- Tools for deployment
  - Firewalls, scanning, virus protection, etc.
  - Limitations / benefits
- Incident Response

Disclaimer

- Examples are in use at various facilities
  - Site specific information not disclosed (for security reasons)
- Your mileage will vary
  - Implementation does **NOT** guarantee success
- Do NOT use this information to test your site's security measures

Computational 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
    - e.g. at NERSC: Mix of DOE 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

U.S. Department of Energy

Office of Science

Network Traffic Pattern Over 3 Week Period

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%

Problem

- How do we secure a site where users are everywhere, staff is scattered and systems have to be open enough so as to not disrupt users and staff so they can perform science unhindered.

- Balancing act between security and availability

Restrictive Environment

Unrestrictive Environment

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

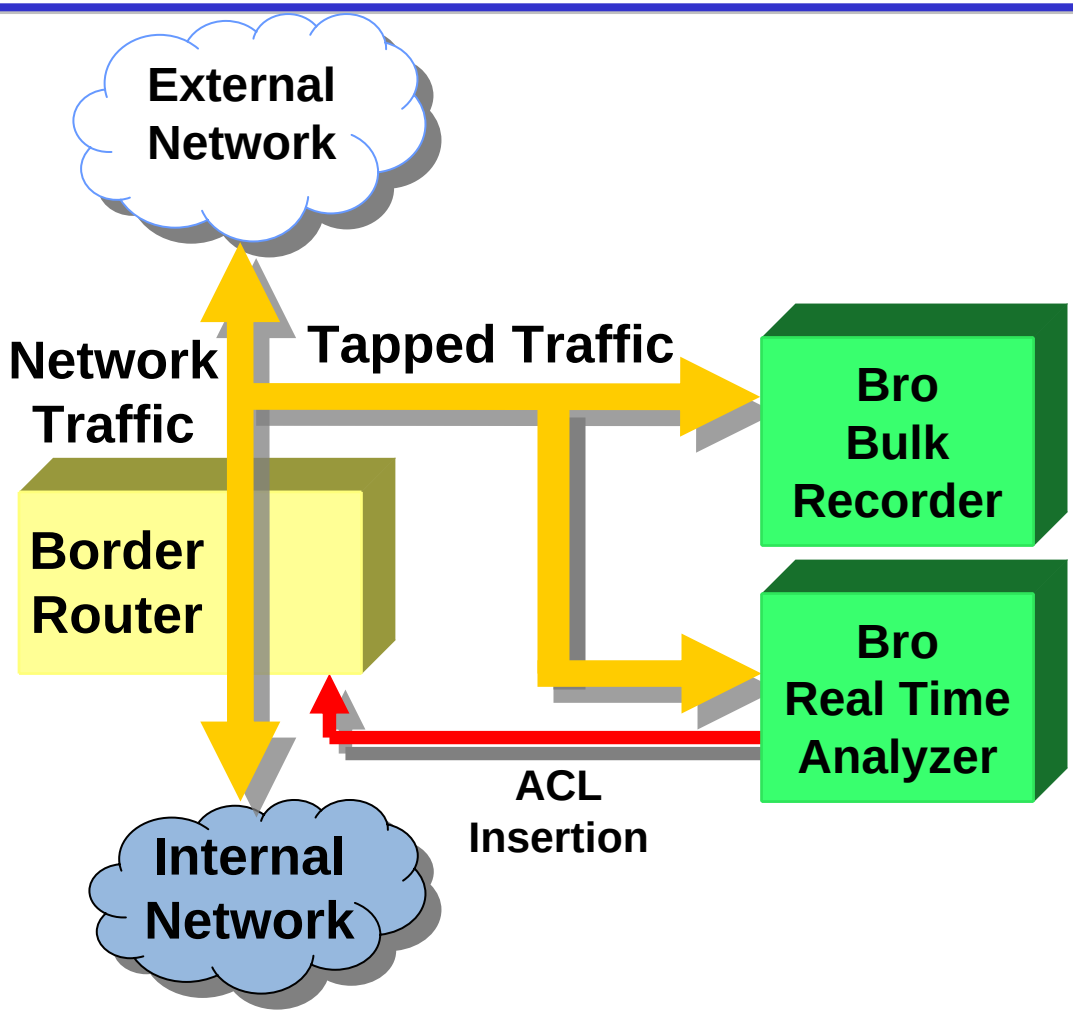
Defense in Depth Layers

- 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

Bro DMZ Implementation



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.

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

Summary

- Open Scientific Facilities are a unique challenge to computer protection
- Defense in Depth relies on multiple tools and techniques
  - Protects against single points of failures
  - Provides multiple methods to evaluate an incident
- Prepare for an incident before it happens

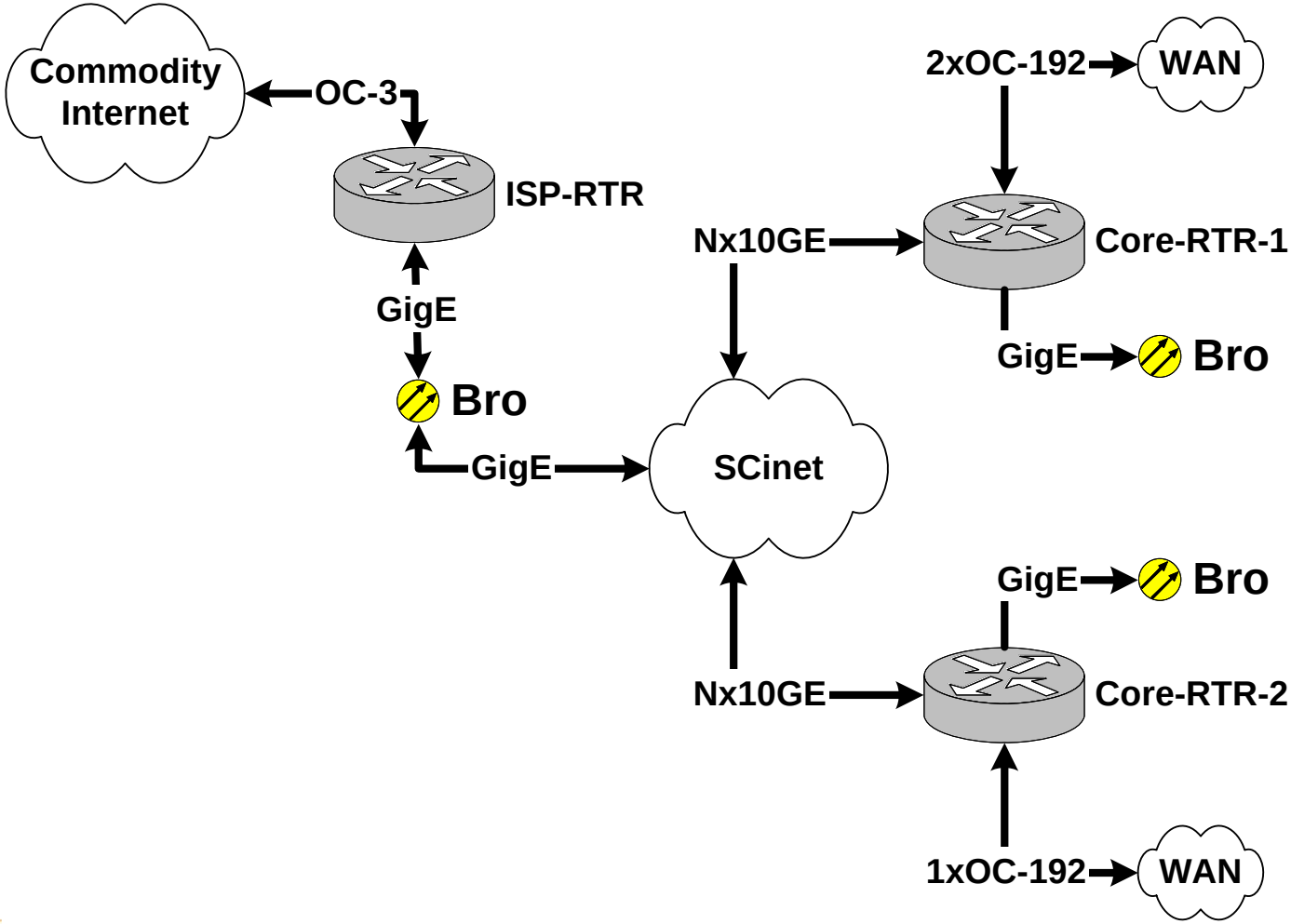
Bro at SC03

- Bro primary IDS for SC conference since SC00
  - Used to monitor SCinet traffic
- Maximum observed bandwidth
  - 16.8Gbps at SC2002 (Bandwidth Challenge)
  - Used router hardware BPF
- Passive monitoring only
  - Automatic countermeasures disabled

Bro at SC03

- IDS for SCinet
  - Ensure conference network does not get taken down by attacks
  - Detect 0wned systems
  - Monitor for “odd” behavior
- Educational tool for attendees
  - Password capture and display
  - Alert exhibitors to “risky behavior”
    - i.e. .rhosts with root enabled

SCinet Bro Infrastructure



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, con't:

- 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, con't:

- 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:

- *Bro* is available from
  - `http://www.icir.org/vern/bro.html`.

- *Snort* is available from `www.snort.org`. Snort is a widely used freeware signature-based NIDS.

- *Tcpdump* and *Libpcap* are available from `www.tcpdump.org`. Tcpdump is a widely used freeware network traffic packet dumper. Libpcap is the portable packet-capture library on which Tcpdump (and Bro) is built.