Intermediate SCADA Security

SS-2 SANS SCADA Summit

September 28,29, 2006

Homeland Security

U.S. DEPARTMENT OF ENERGY

Disclaimer

References made herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government, any agency thereof, or any company affiliated with the Idaho National Laboratory.

Use the described security tools & techniques at "***your own risk***" – i.e. carefully evaluate any tool prior to using it in a production SCADA Network.

Agenda

- Introduction
- Demonstration
- “Chalk Talk”
- Network Layers Review
- Vulnerability Reduction Process
- 10 Minute Break
- Application Security Identification and Remediation
- Lunch
- SQL Injection Identification and Remediation
- 10 Minute Break
- Unauthorized Control Identification and Remediation
- 10 Minute Break
- Interactive Test Discussion

Instructors

Kenneth Rohde
Communications & Cyber Security
Critical Infrastructure Division

James Davidson
SCADA & Control Systems
Critical Infrastructure Division

May Permann
Communications & Cyber Security
Critical Infrastructure Division

John Hammer
Communications & Cyber Security
Critical Infrastructure Division

email: scadasummit@inl.gov
toll-free phone number: 866-495-7440

Why This Class?

- **To develop an understanding of tools & methods that can be used to:**
  - **Discover & identify vulnerabilities in your system**
    (CIP-005 R3.2, CIP-005 R4, & CIP-007 R8)
  - **Develop mitigation strategies for resolving these issues**
    (CIP-007 R3)
  - **Fix the problems you find**
- **Ask yourself**
  - **If I don't know I have a problem, how can I fix it?**

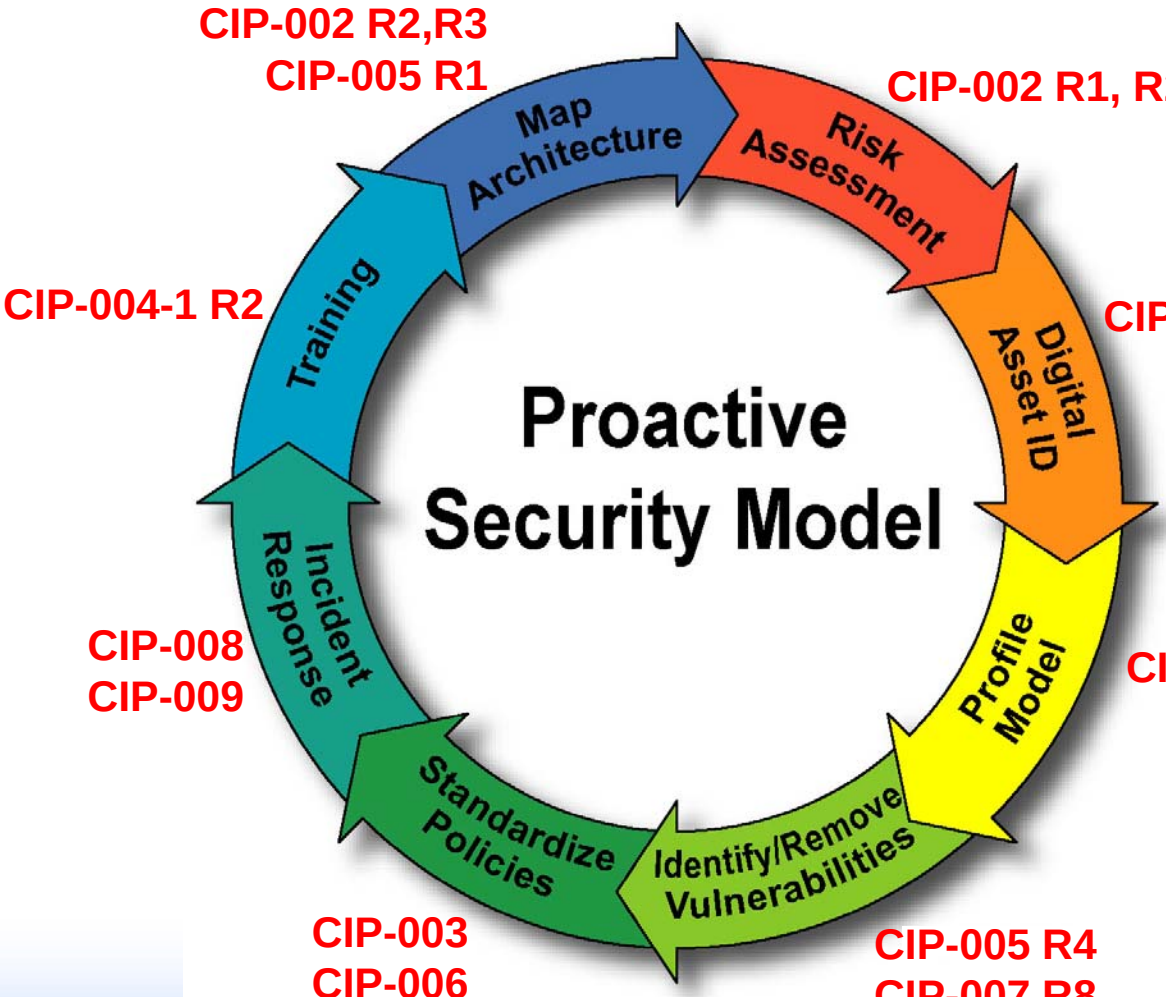
NERC Top 10 Vulnerabilities

1. Policies, procedures & culture governing control system security are inadequate & lead to lack of executive management buy in. In addition, personnel routinely ignore or lack training in policies & procedures to protect the control systems.
2. Poorly designed control system networks that fail to employ sufficient defense-in-depth mechanisms.
3. Remote access to the control system through means which do not provide identity control.
4. Prescribed system administration mechanisms are not part of control system implementation.
5. Use of wireless communication
6. Lack of a dedicated communications channel for command & control in applications such as Internet based SCADA, & inappropriate use of control system network bandwidth for non control purposes.
7. Lack of quick & easy tools to detect & report on anomalous or inappropriate activity. Non existent forensic & audit methods.
8. Installation of inappropriate applications on critical systems.
9. Software used in control systems is not adequately scrutinized, & newer systems include extraneous vulnerable software.
10. Control systems data sent in clear text.

These are not in any order of importance
Items in blue are covered by this course

There is No Such Thing as a Secure System!

Security is a Never Ending Process

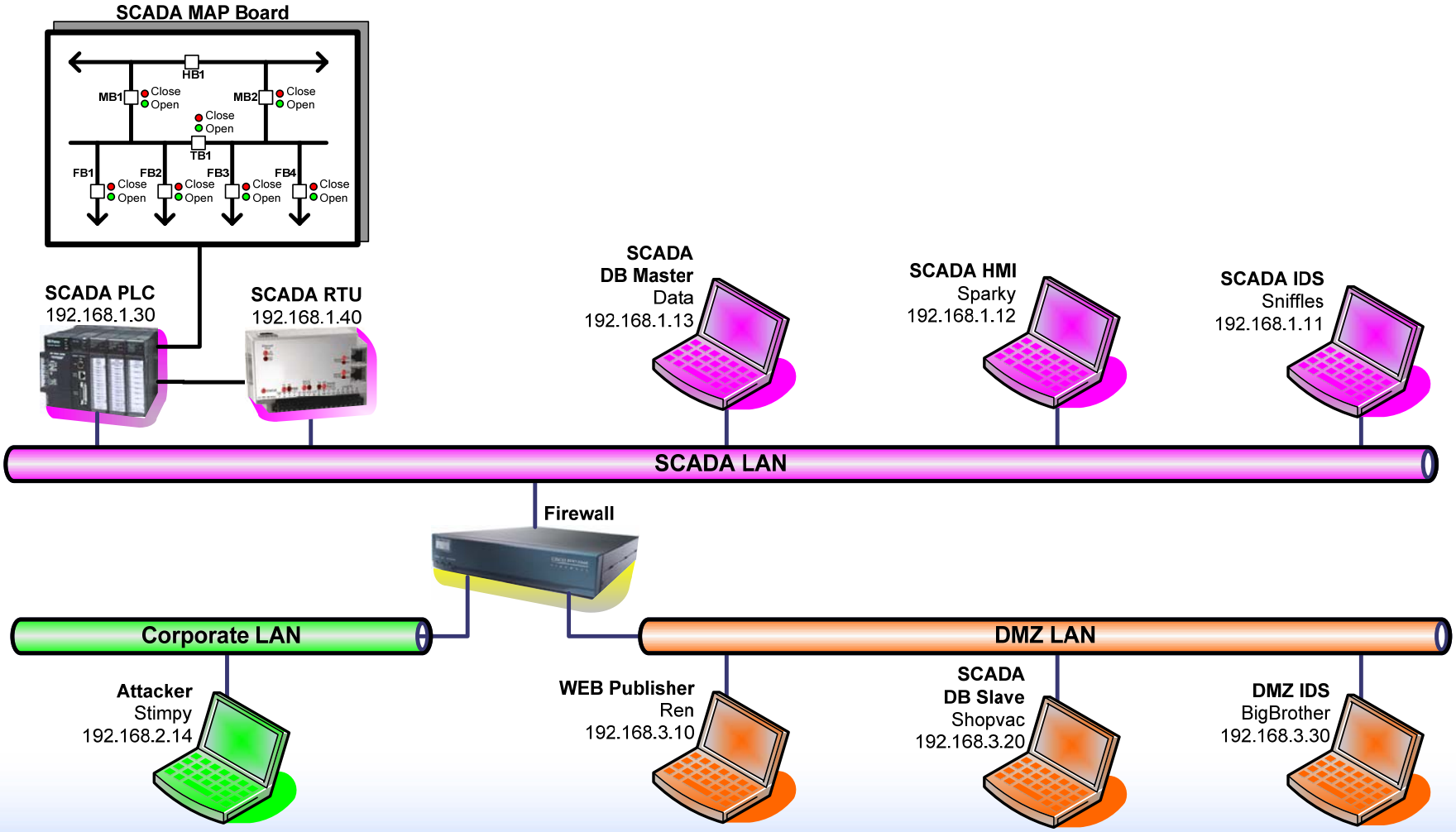


Idaho National Laboratory

Demo

SCADA Exploit Demonstration

Demo Network Layout

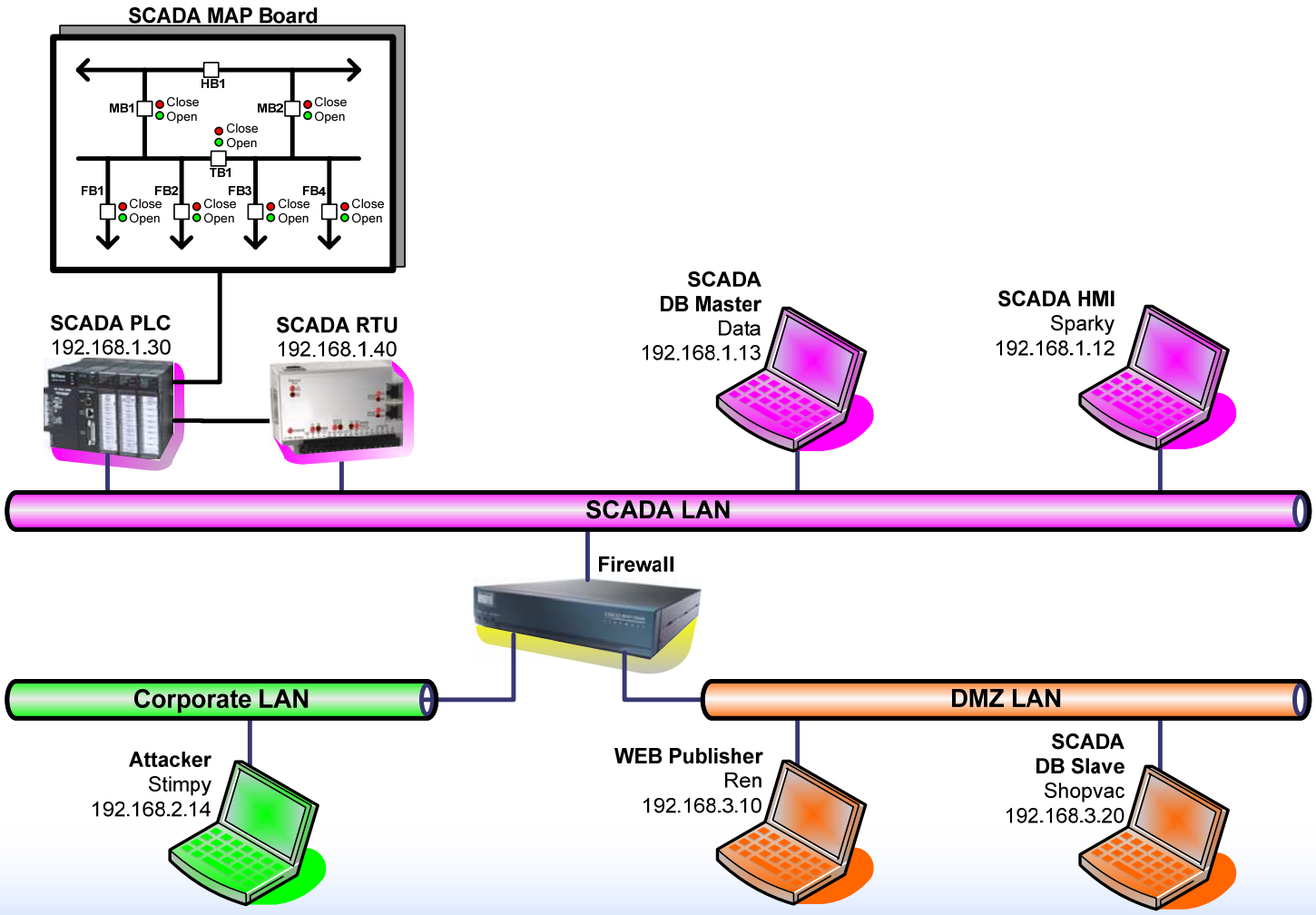


Idaho National Laboratory

Demo System Vulnerabilities

- ✓ Clear Text Communications
- ✓ Network Switch Configuration Flaws
- ✓ Dynamic ARP Tables
- ✓ Poorly Defined Firewall Policy
- ✓ Intrusion Detection System (IDS) Configured Poorly, Unusable
- ✓ Poor Application Coding Practices
- ✓ Improper Application & Service Privileges

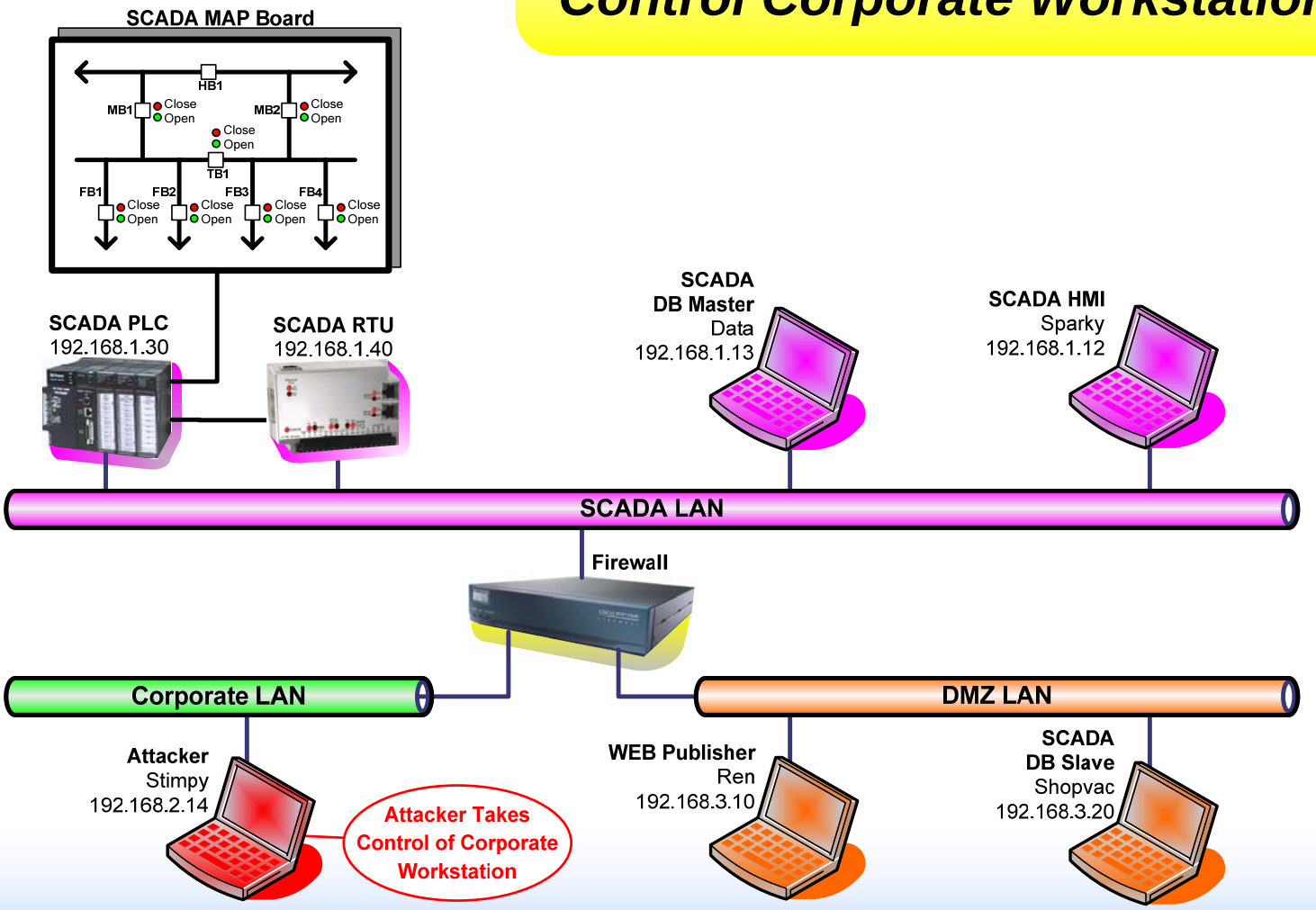
Starting Configuration



Idaho National Laboratory

Attack

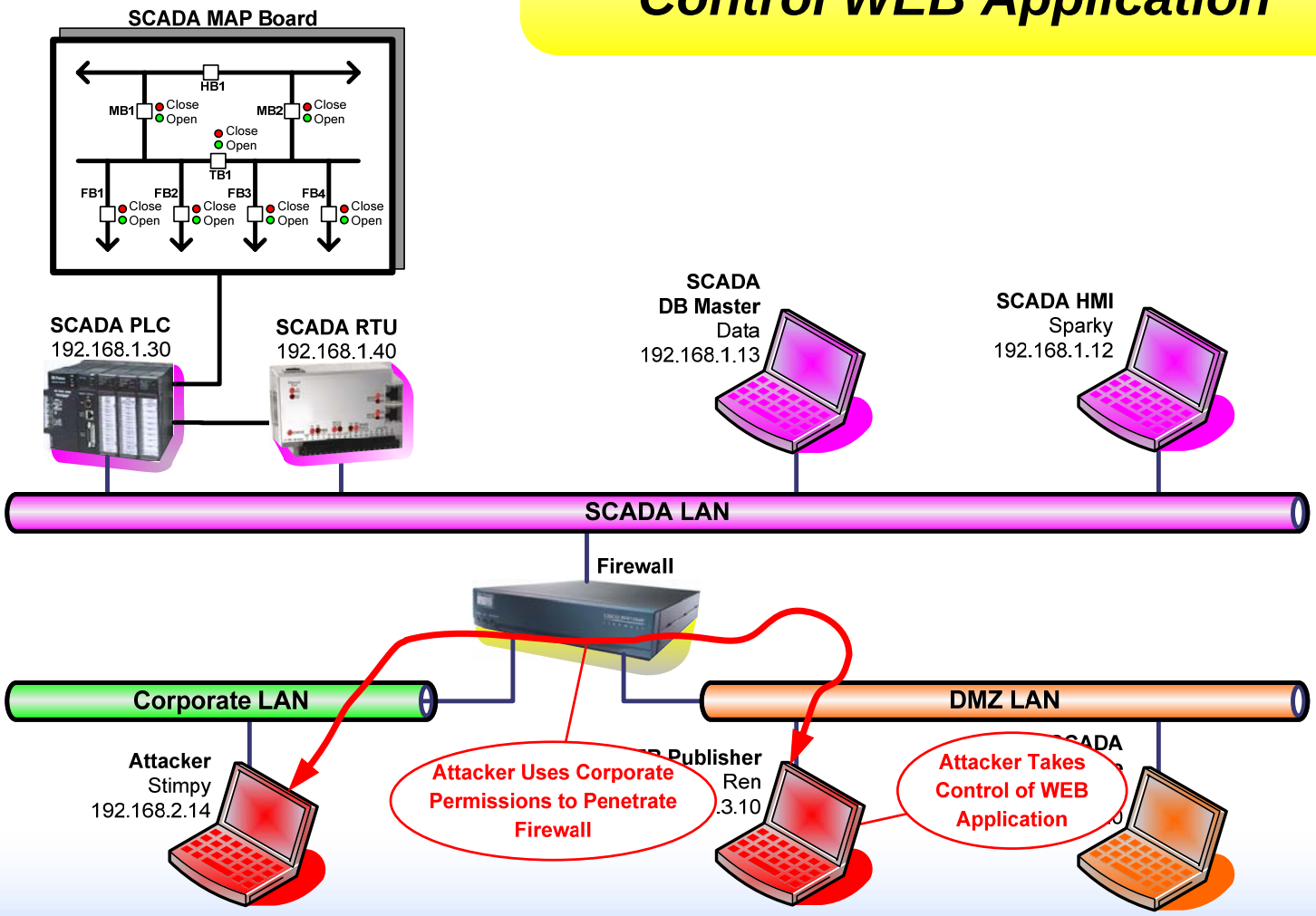
Control Corporate Workstation



Idaho National Laboratory

Attack

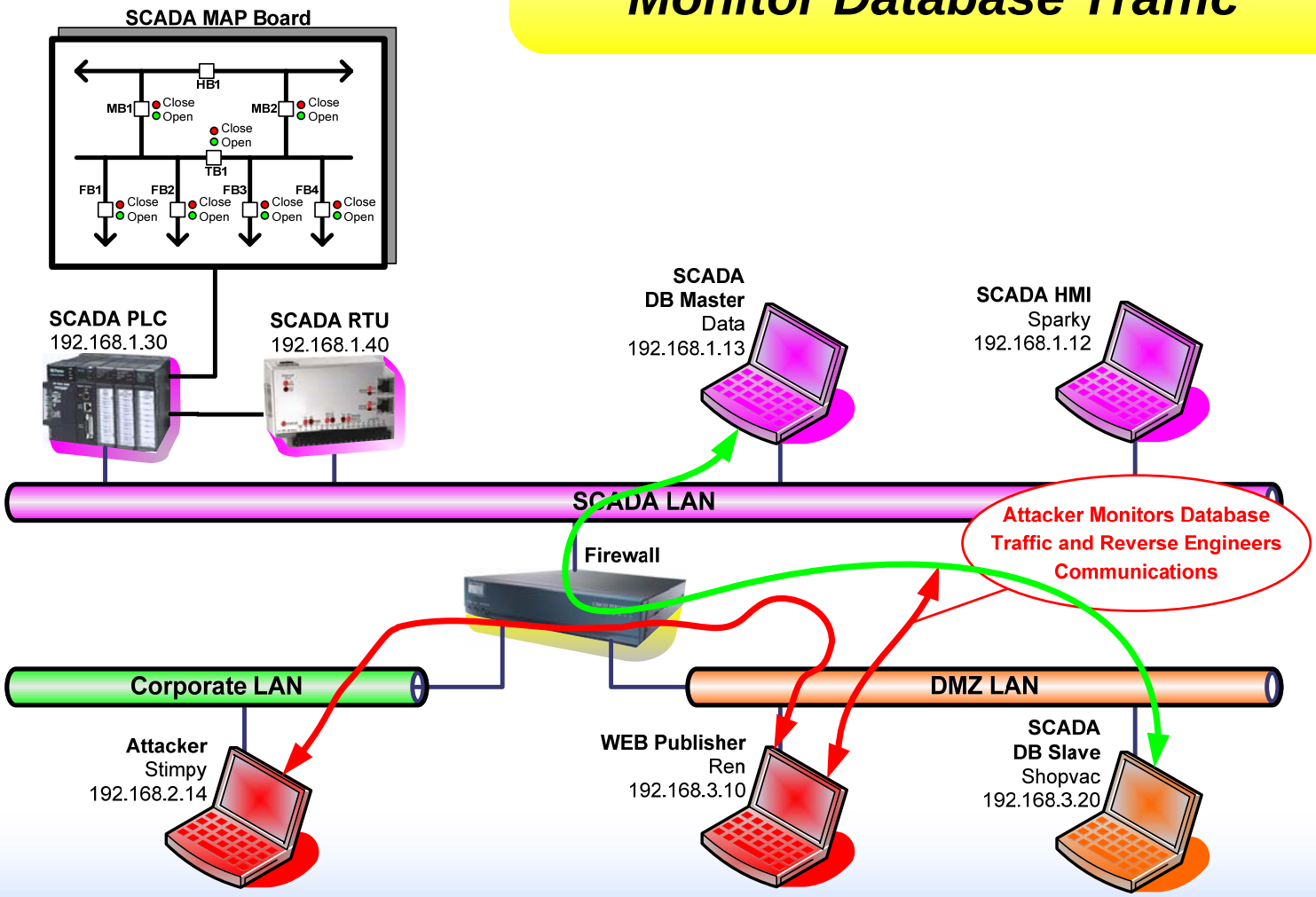
Control WEB Application



Idaho National Laboratory

Attack

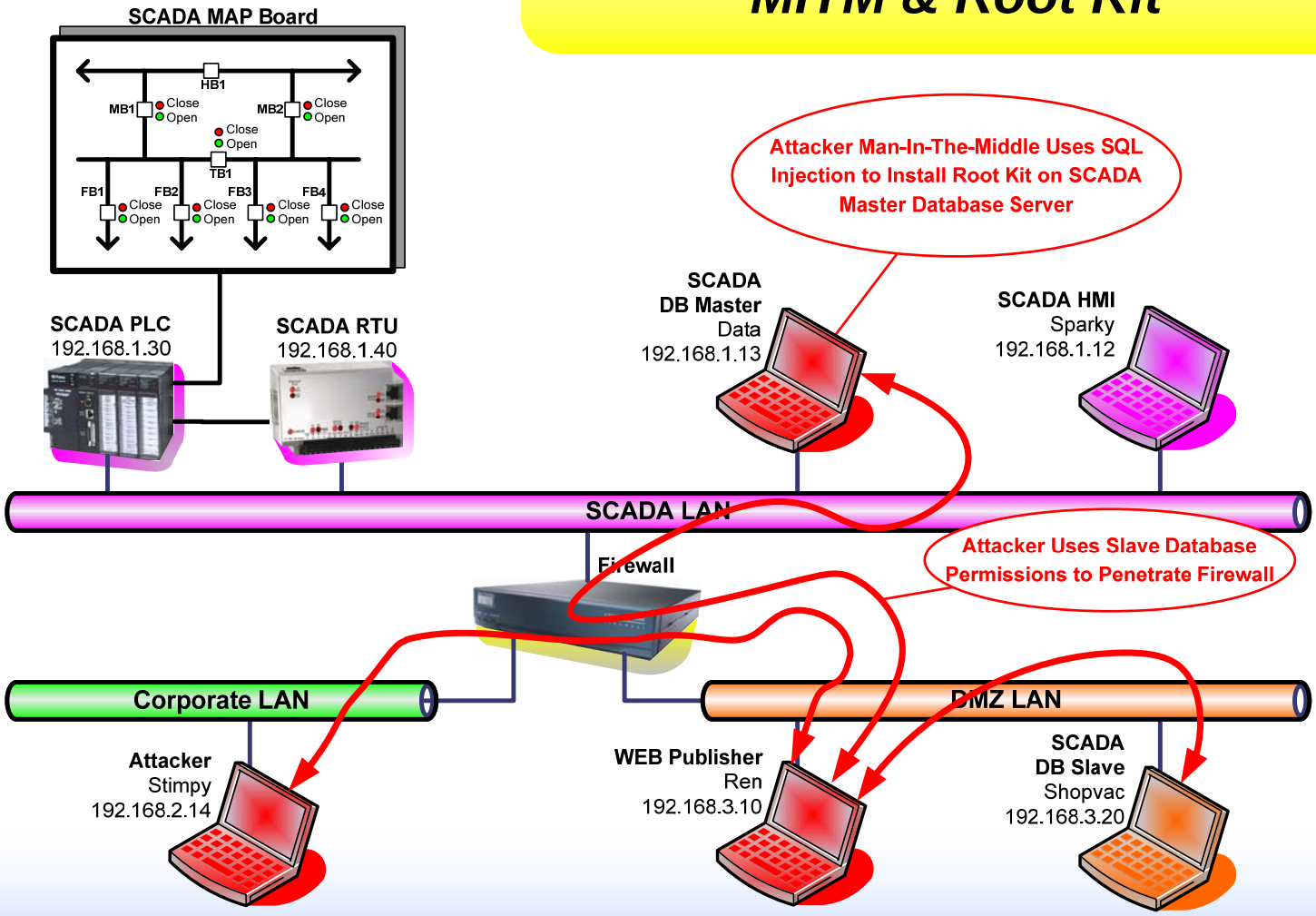
Monitor Database Traffic



Idaho National Laboratory

Attack

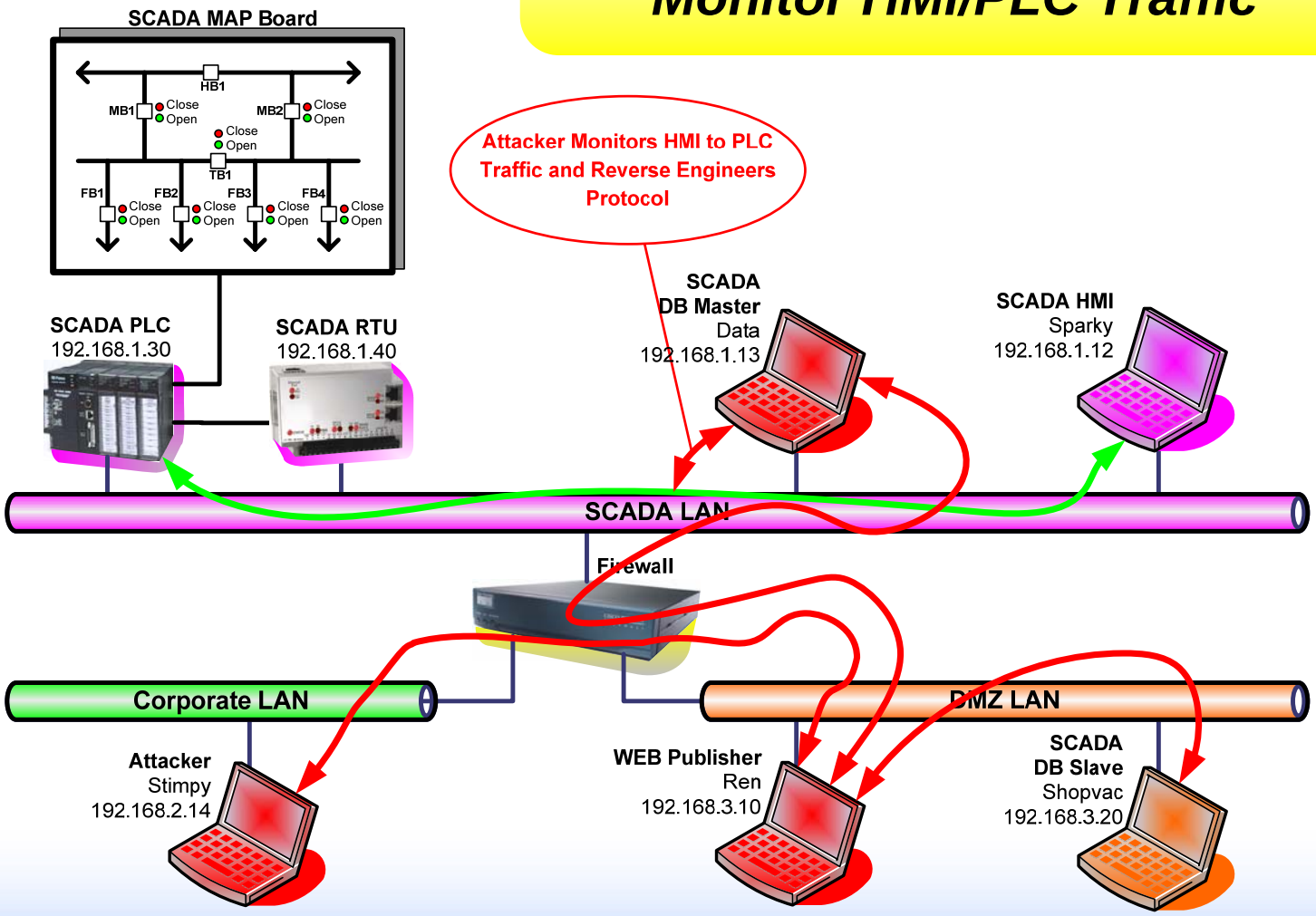
MITM & Root Kit



Idaho National Laboratory

Attack

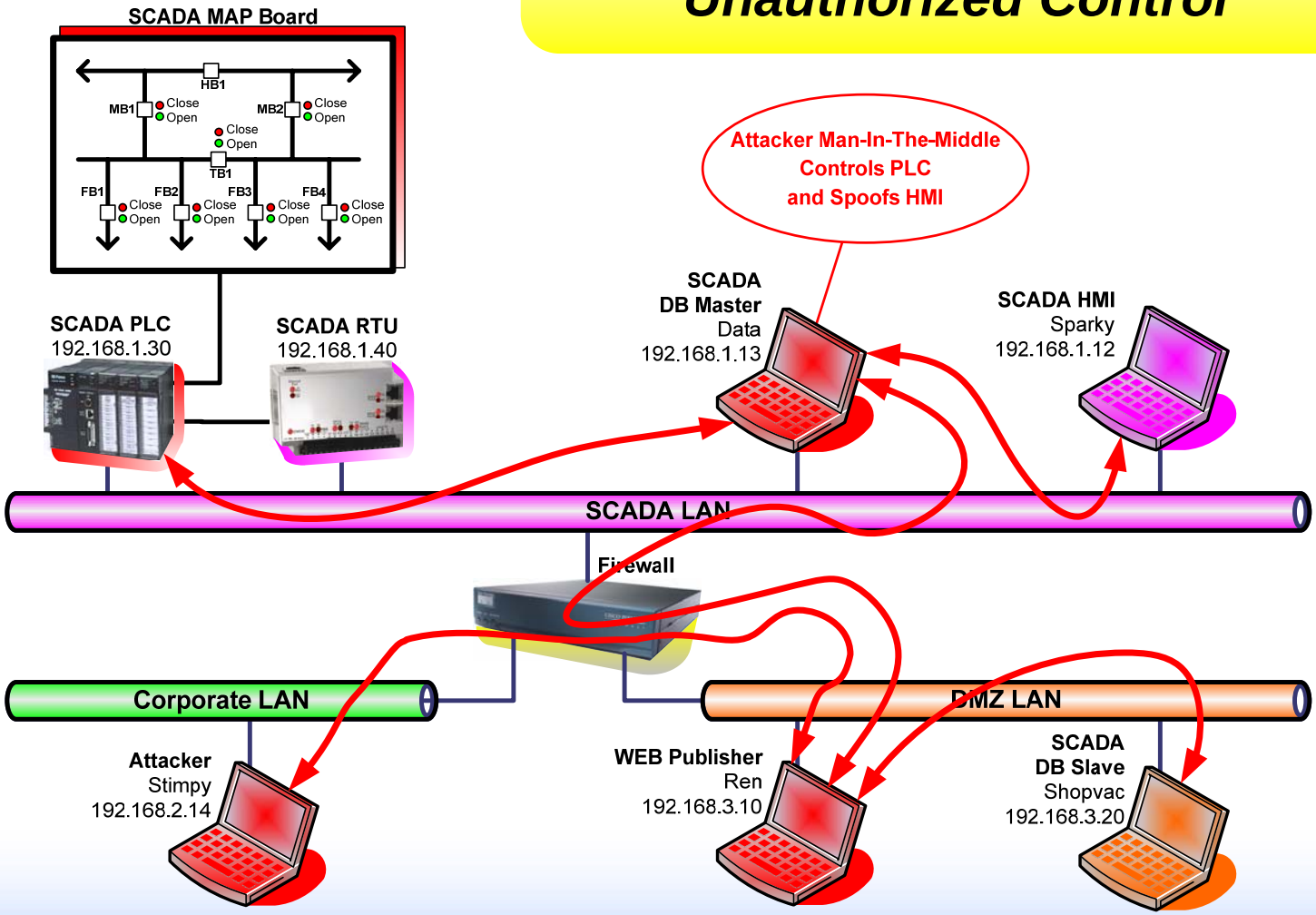
Monitor HMI/PLC Traffic



Idaho National Laboratory

Attack

Unauthorized Control

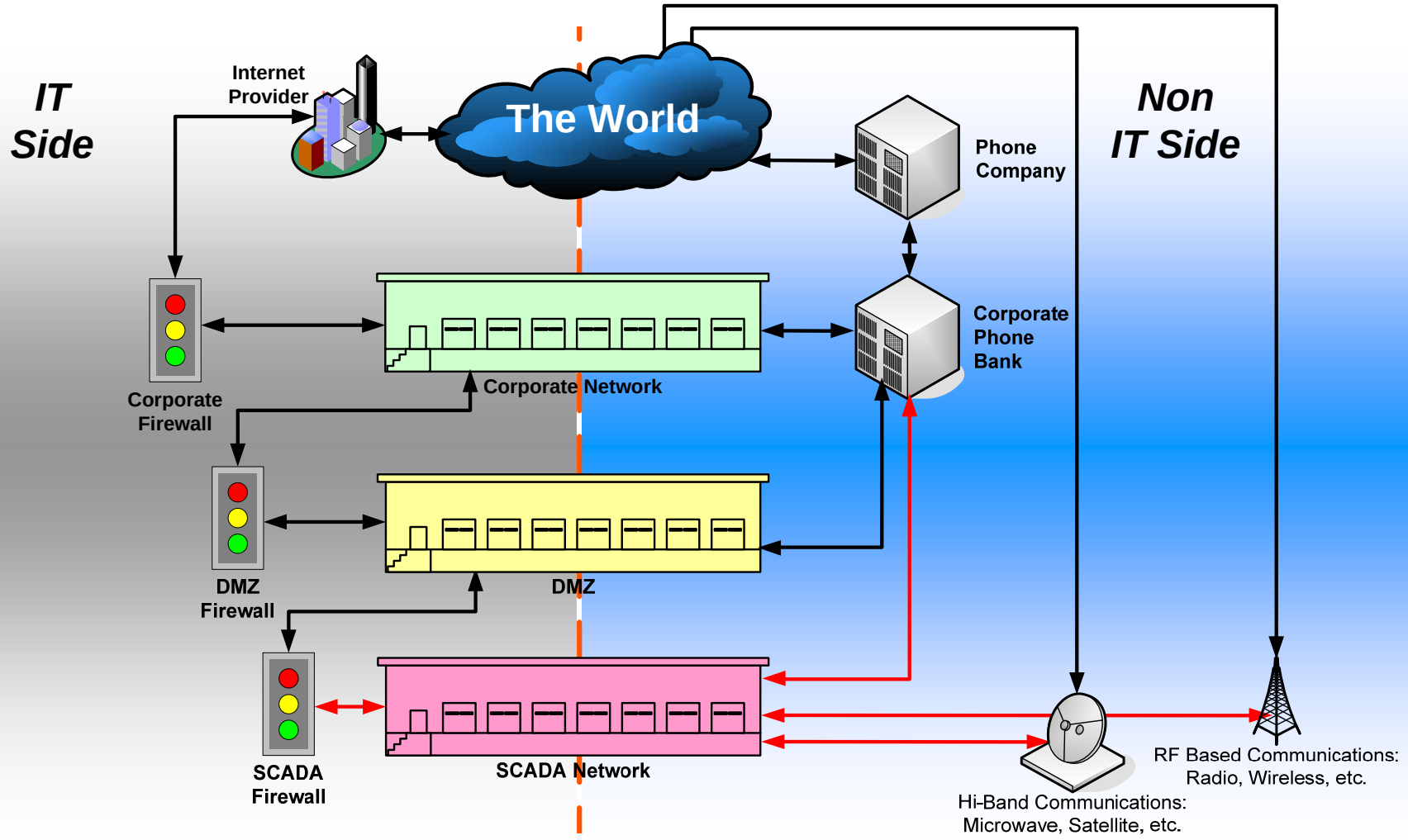


Idaho National Laboratory

SCADA Security

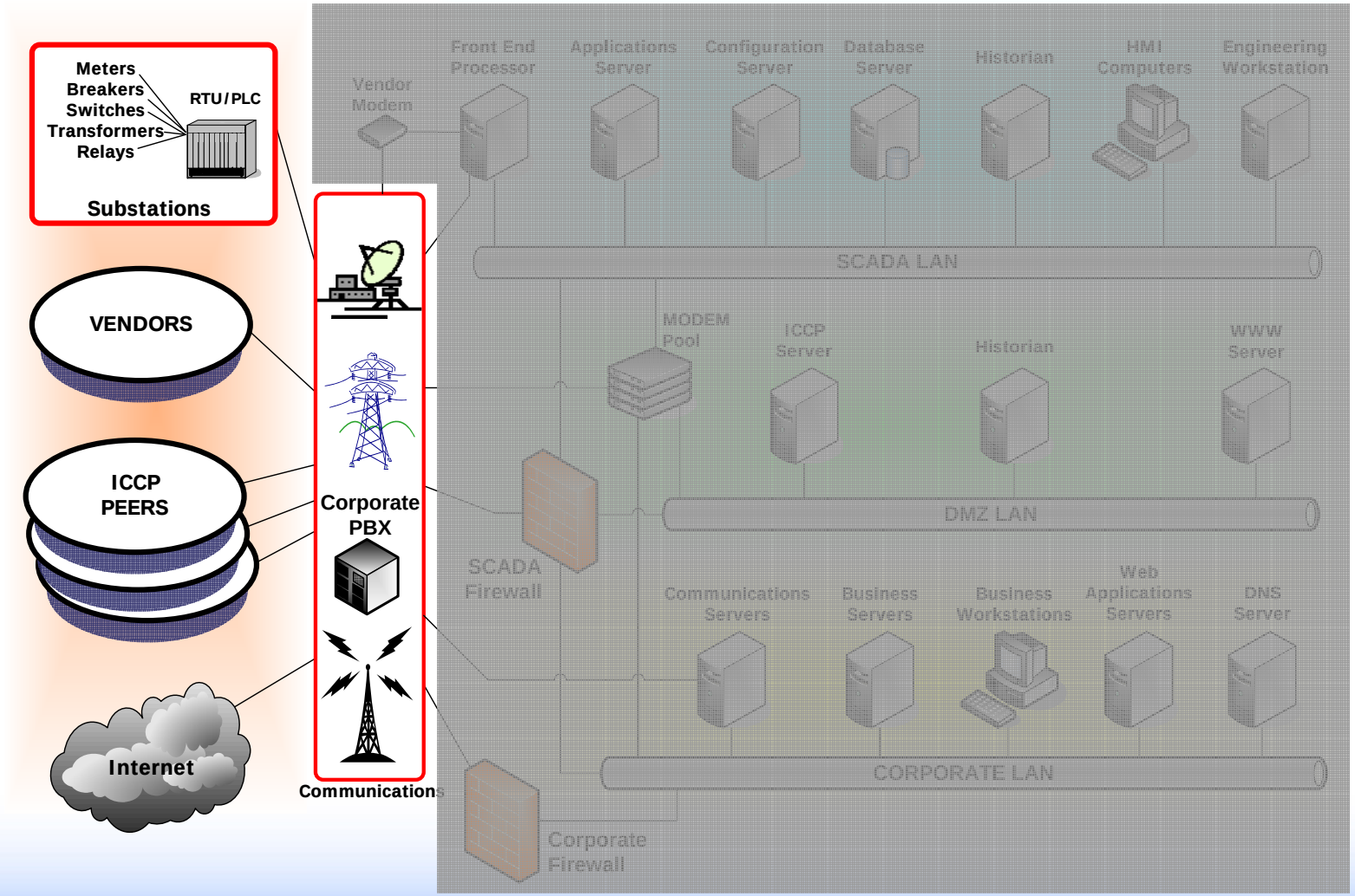
“Chalk Talk”

Electronic Perimeter



Routes of Entry

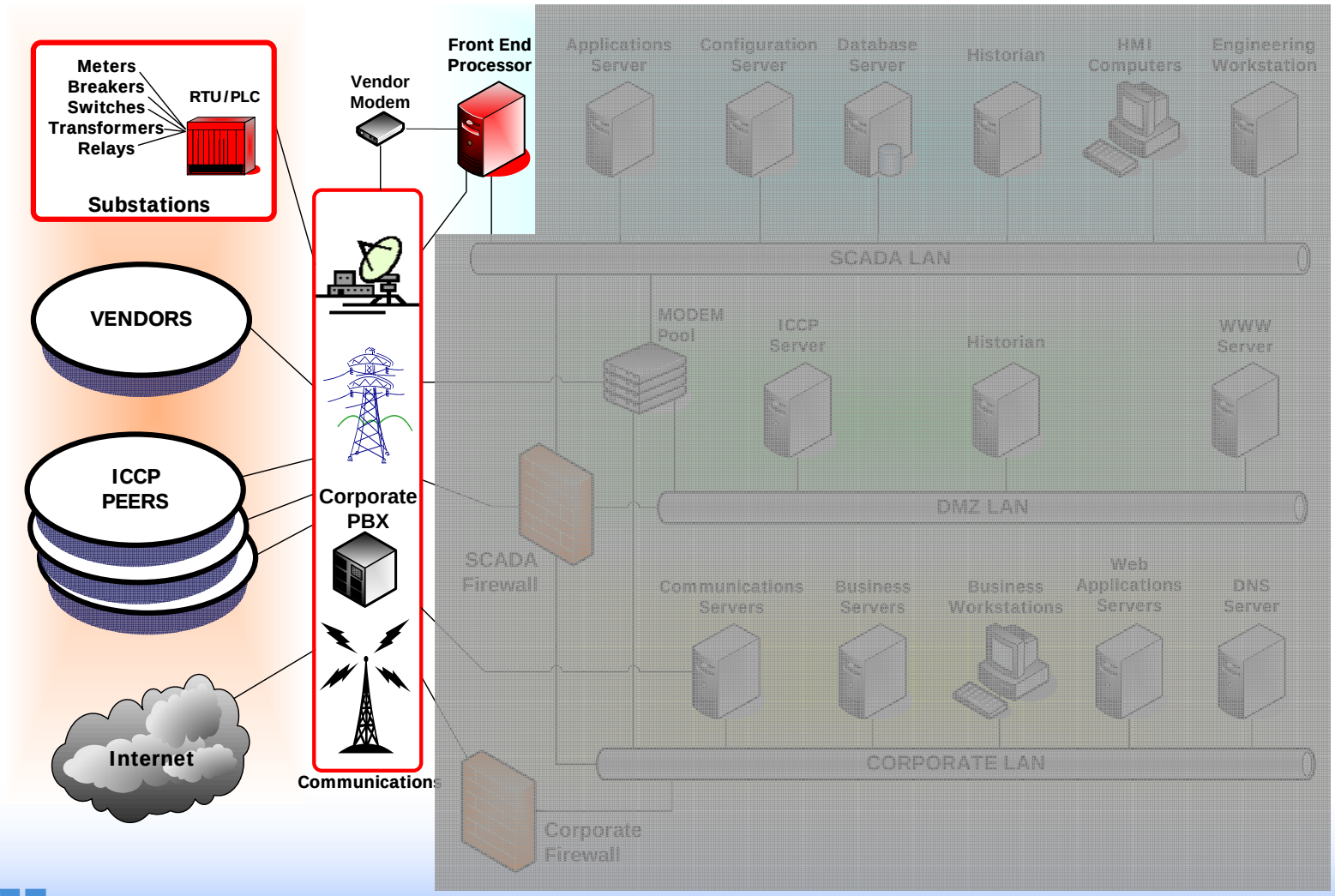
RTU Modems



Idaho National Laboratory

Routes of Entry

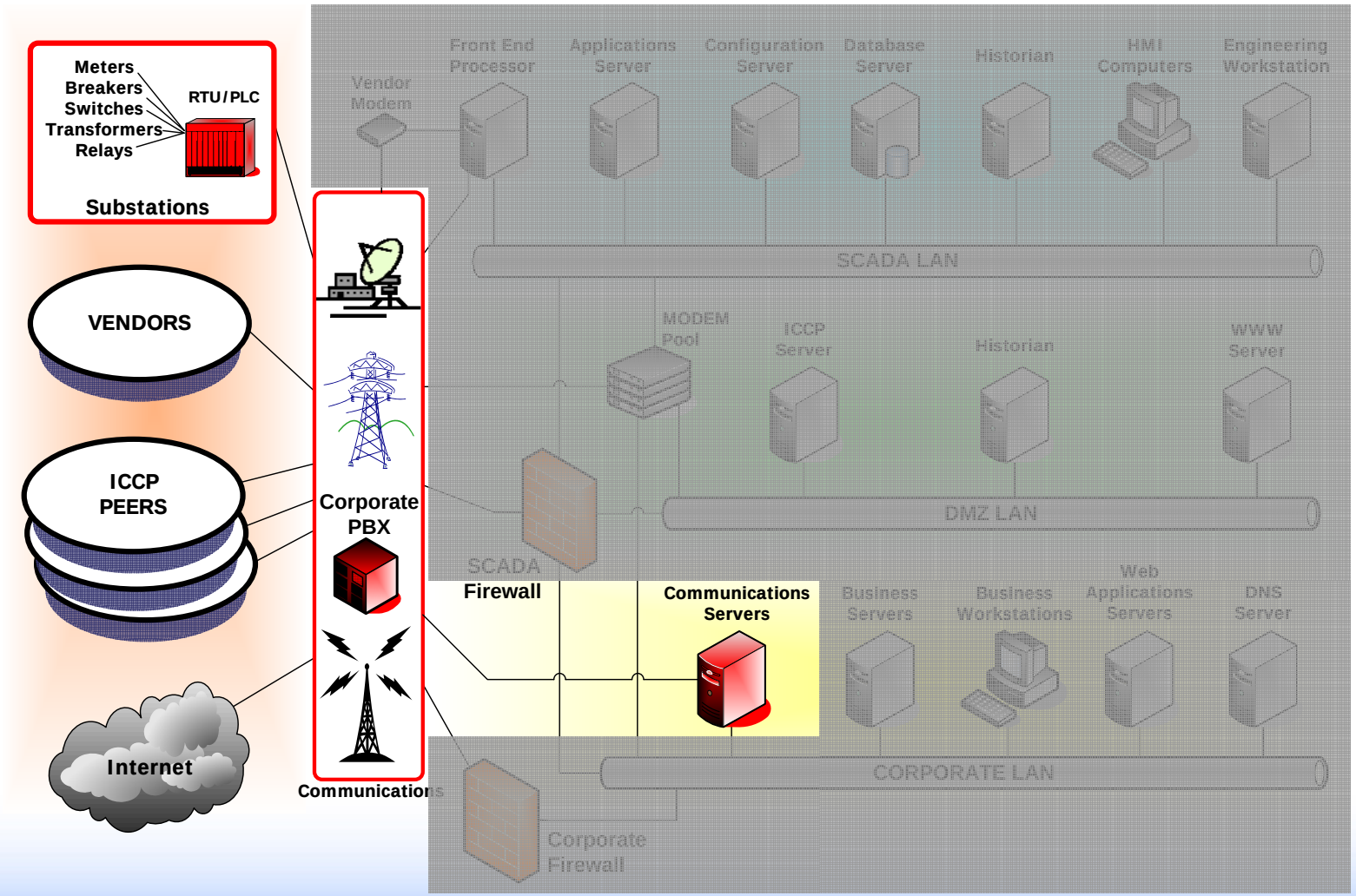
Communication Lines



Idaho National Laboratory

Routes of Entry

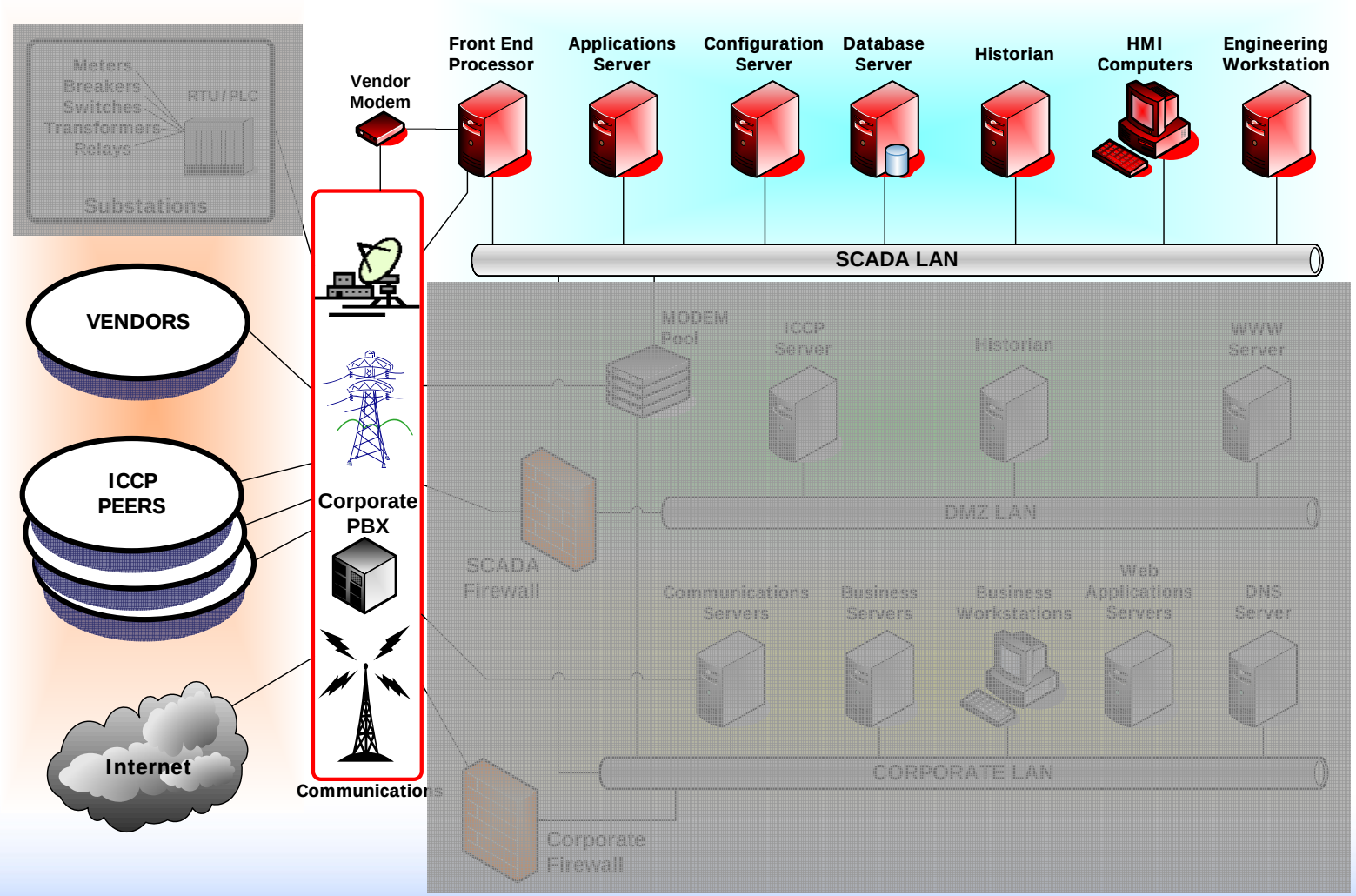
IT Controlled Communications Equipment



Idaho National Laboratory

Routes of Entry

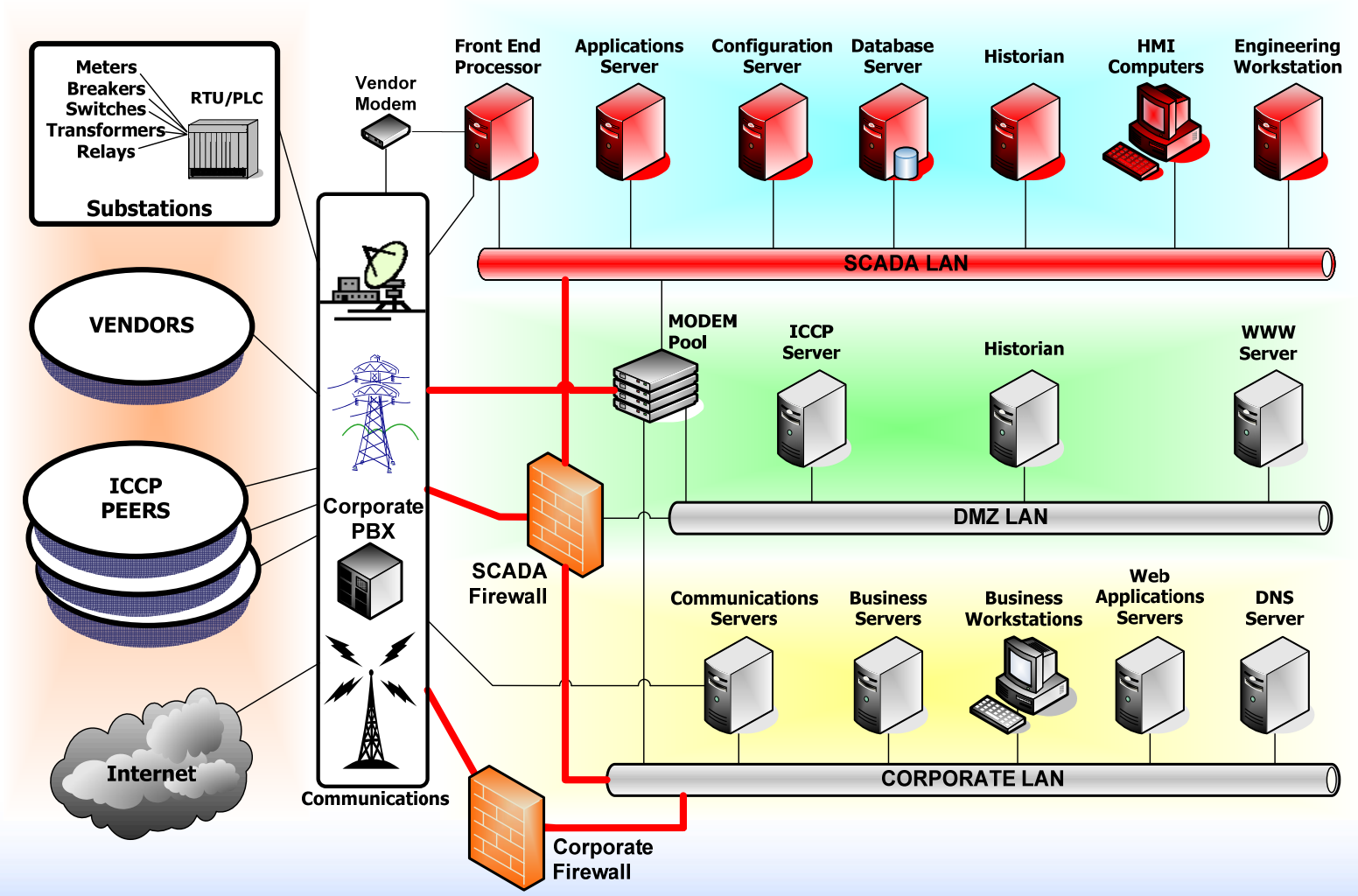
Vendor Support



Idaho National Laboratory

Routes of Entry

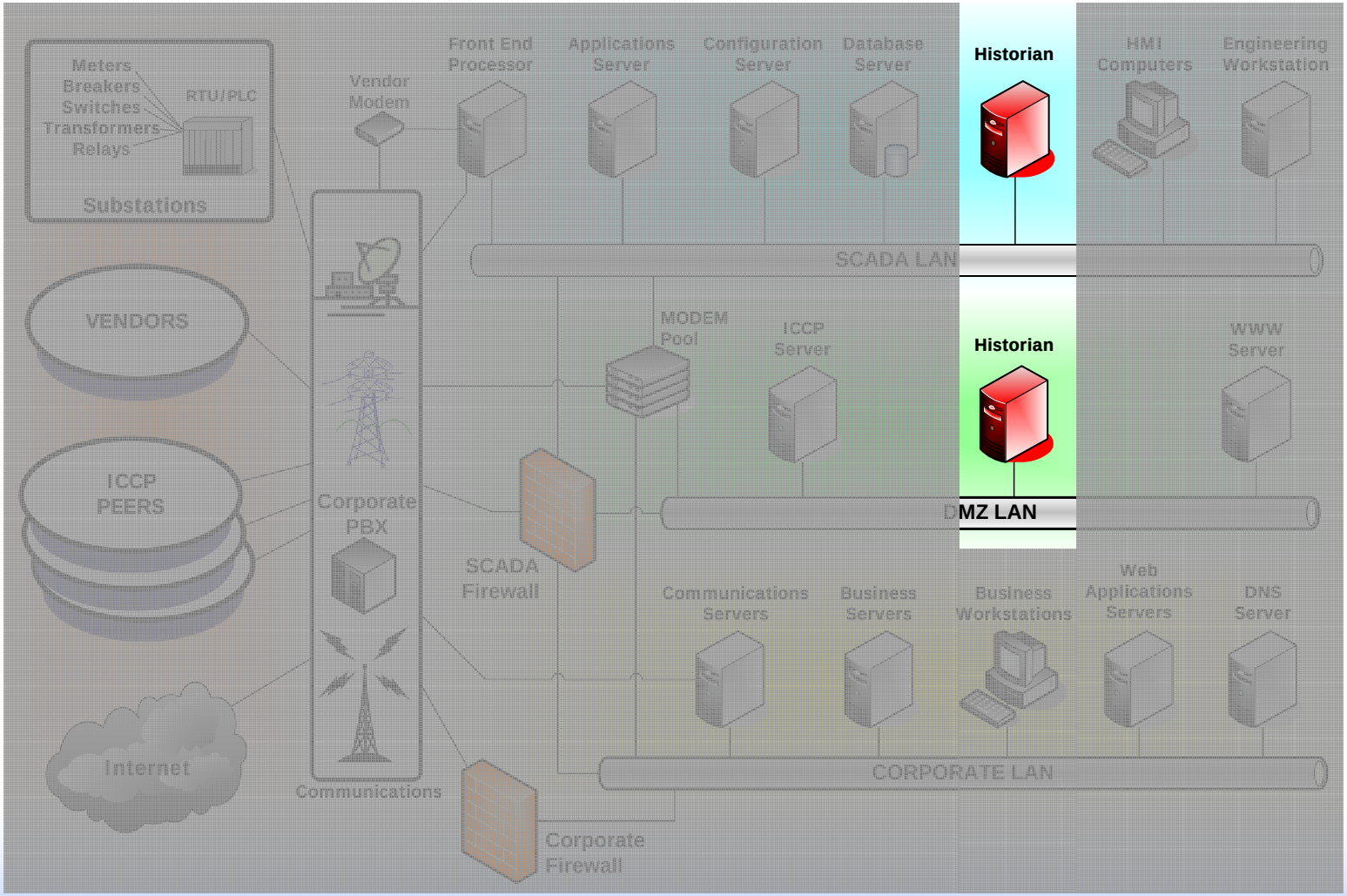
VPNs



Idaho National Laboratory

Routes of Entry

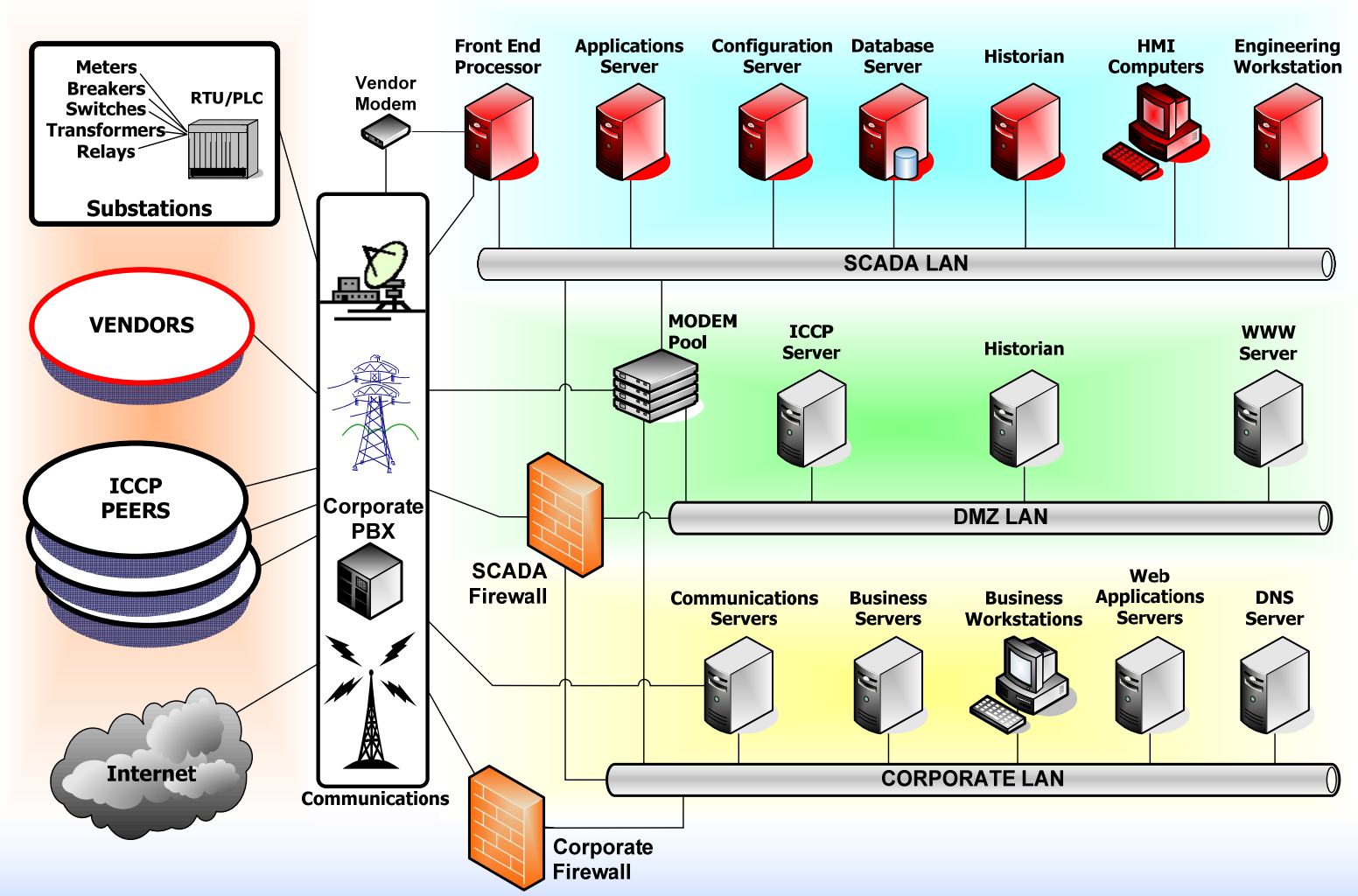
Database Links



Idaho National Laboratory

Routes of Entry

Vendor Patches



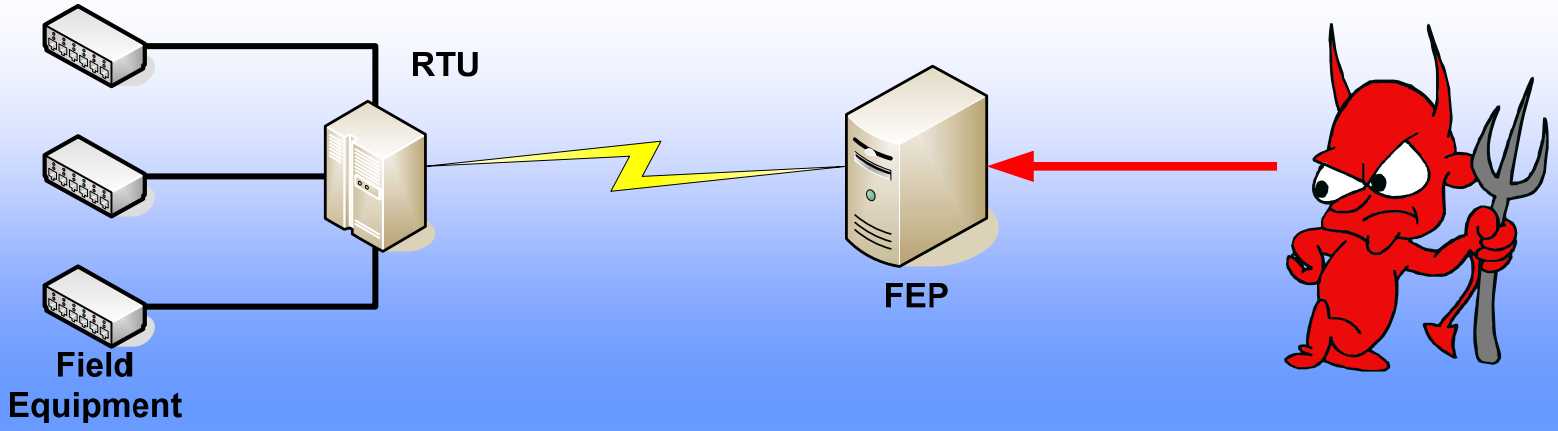
Idaho National Laboratory

Considerations

- **Knowledge of the process is key for long term or surgical disruption**
- **Field equipment generally doesn't contain process knowledge**
  - **Breaker 17A**
  - **Valve 4**
- **Direct access to field equipment without additional knowledge generally only results in nuisance disruption**

Manipulation of the System

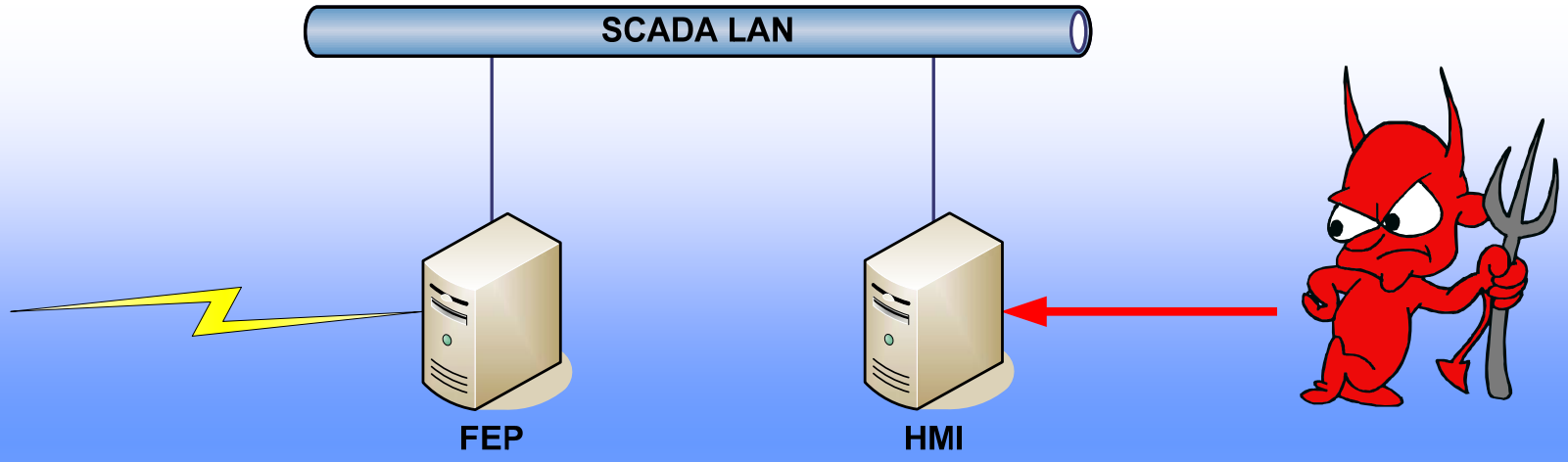
Talk Directly to the Front-End Equipment



- **Often no userid/passwords required**
- **Undocumented vendor protocols are common**
- **Commands are generally not logged**

Manipulation of the System

Export the HMI Screen

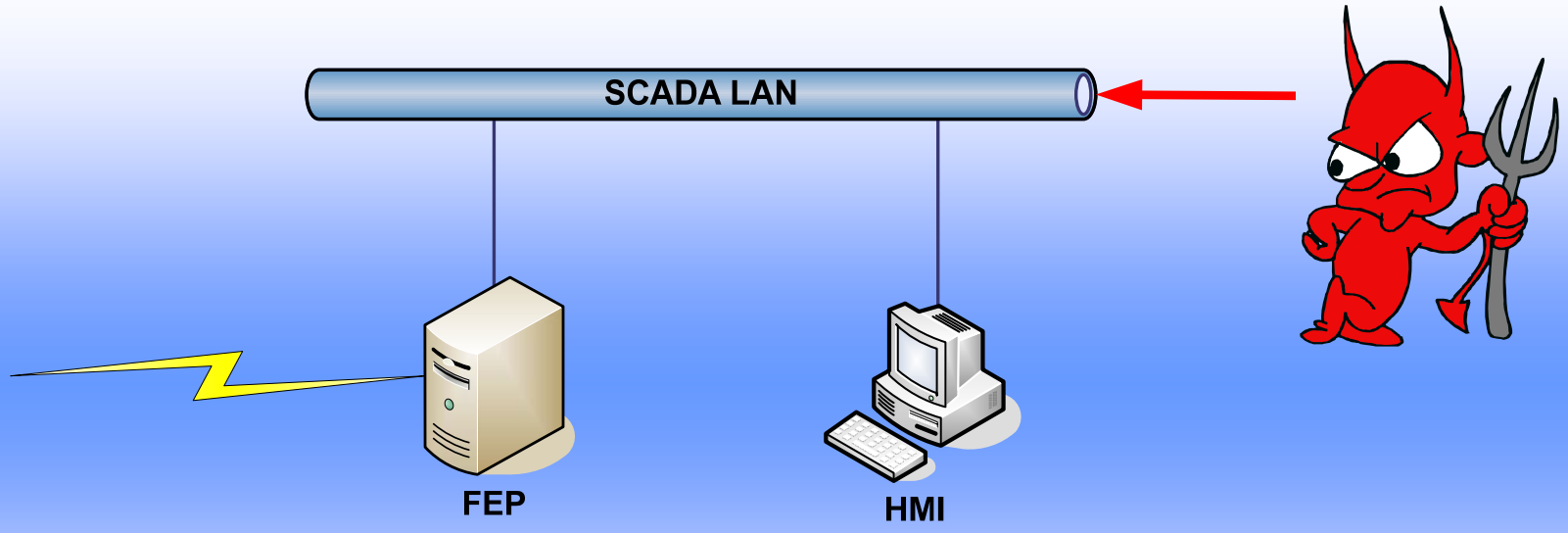


- **Pretty pictures to describe the process**
- **Noticeable by the operator**
- **Can use your off-the-shelf tools**
- **Have credentials of logged in user**
- **May not be able to manipulate to failure**

Idaho National Laboratory

Manipulation of the System

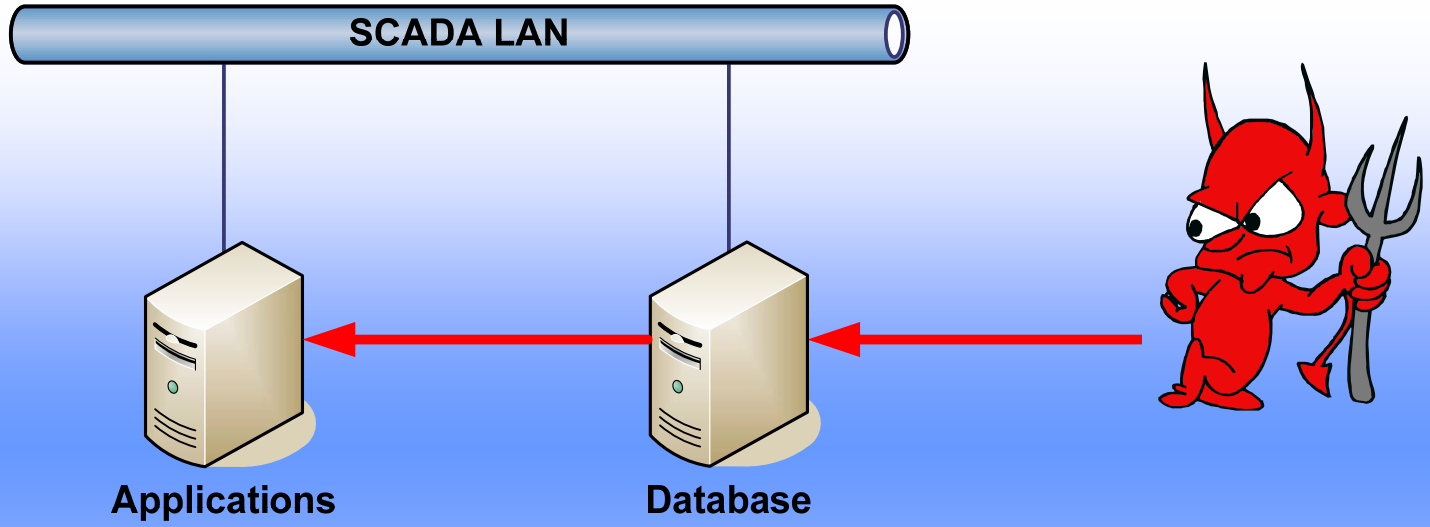
Peer Utility



- **Often the least secured link**
- **Necessary for operation in electric power**
- **Peers often have limited rights on peer's system**

Manipulation of the System

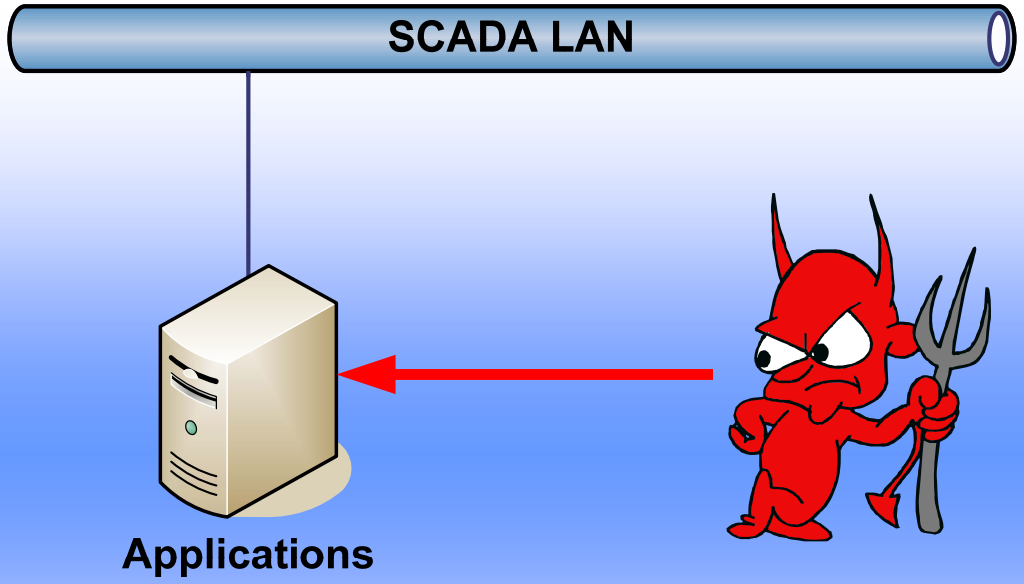
Changing Data in the Database



- **Application Server will make decisions based on bad data**
- **Not all vendor systems vulnerable**

Manipulation of the System

Insert Commands in the Application Stream

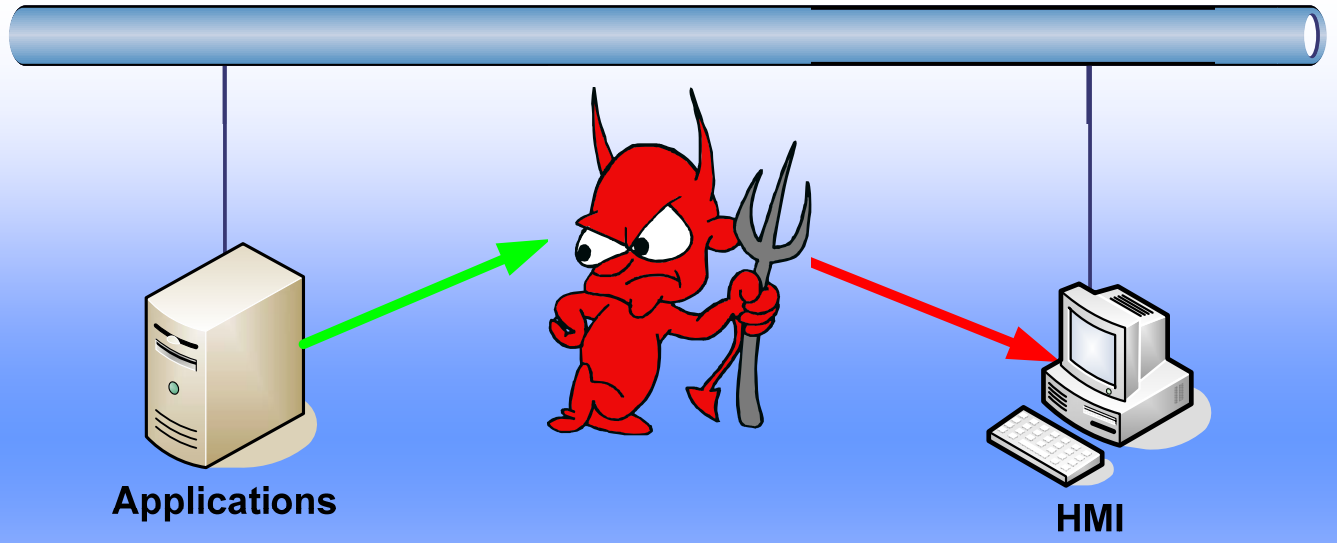


- **Must understand vendor protocols**
- **Logged as actions by the operator**
- **Generally can bypass failure logic**
- **May or may not need credentials**

Idaho National Laboratory

Manipulation of the System

Change Operator's Display



- **If presented with an out-of-control system, operator will take steps to shut down**
- **Logs will reflect operator actions & true state of system**
- **Detailed knowledge of process needed to make believable**

Observations from the Field

"We have no outside communications….except for that one…and that one…and that one…"

"Hackers don't understand process control."

"Patches have historically broken SCADA systems."

"Fear of regulation is greater than fear of attack."

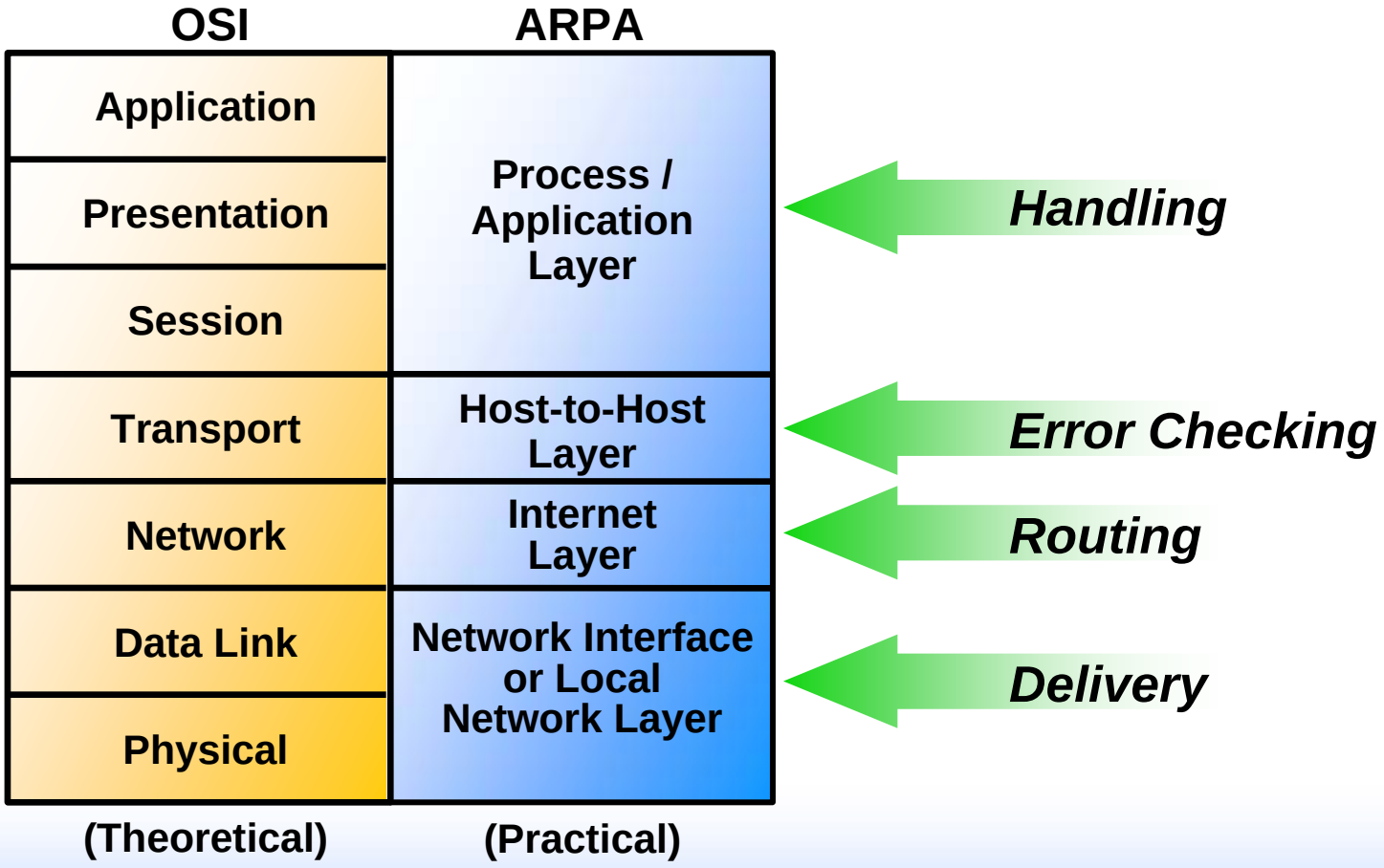
"I'm only going to tell NERC about a couple of assets, they don't need to know about my whole system."

"It's only one-way traffic, my vendor says he only writes to the database."

Network Layers Review

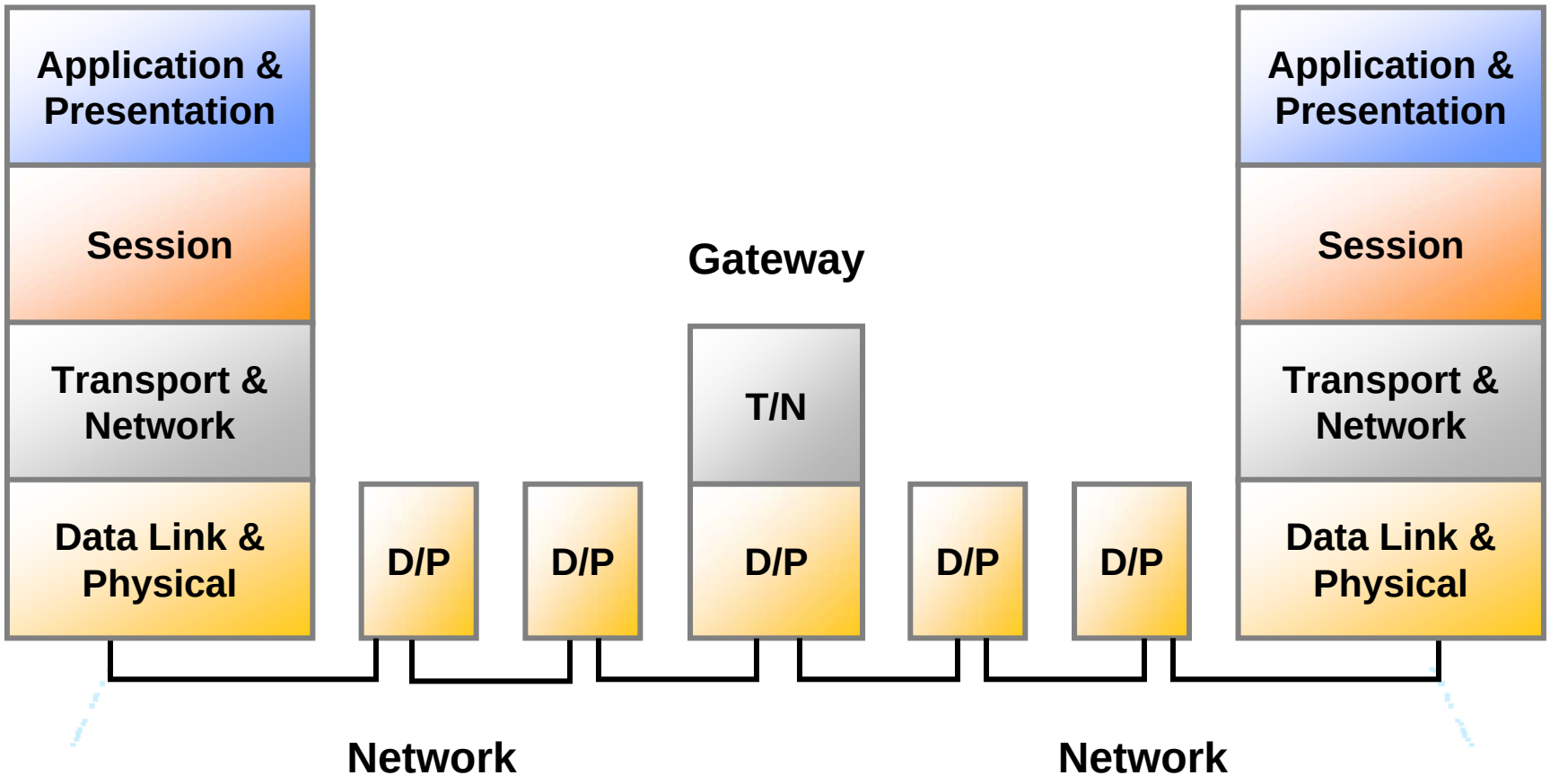
Network Layers

The OSI & the ARPA Layered Architecture

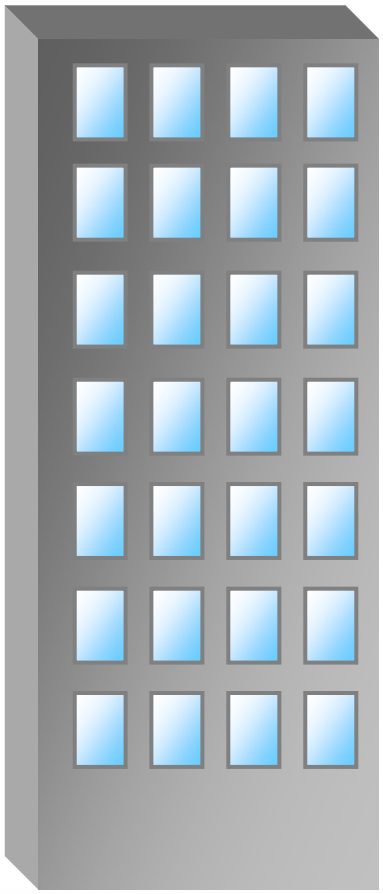


Idaho National Laboratory

A Packet in Time



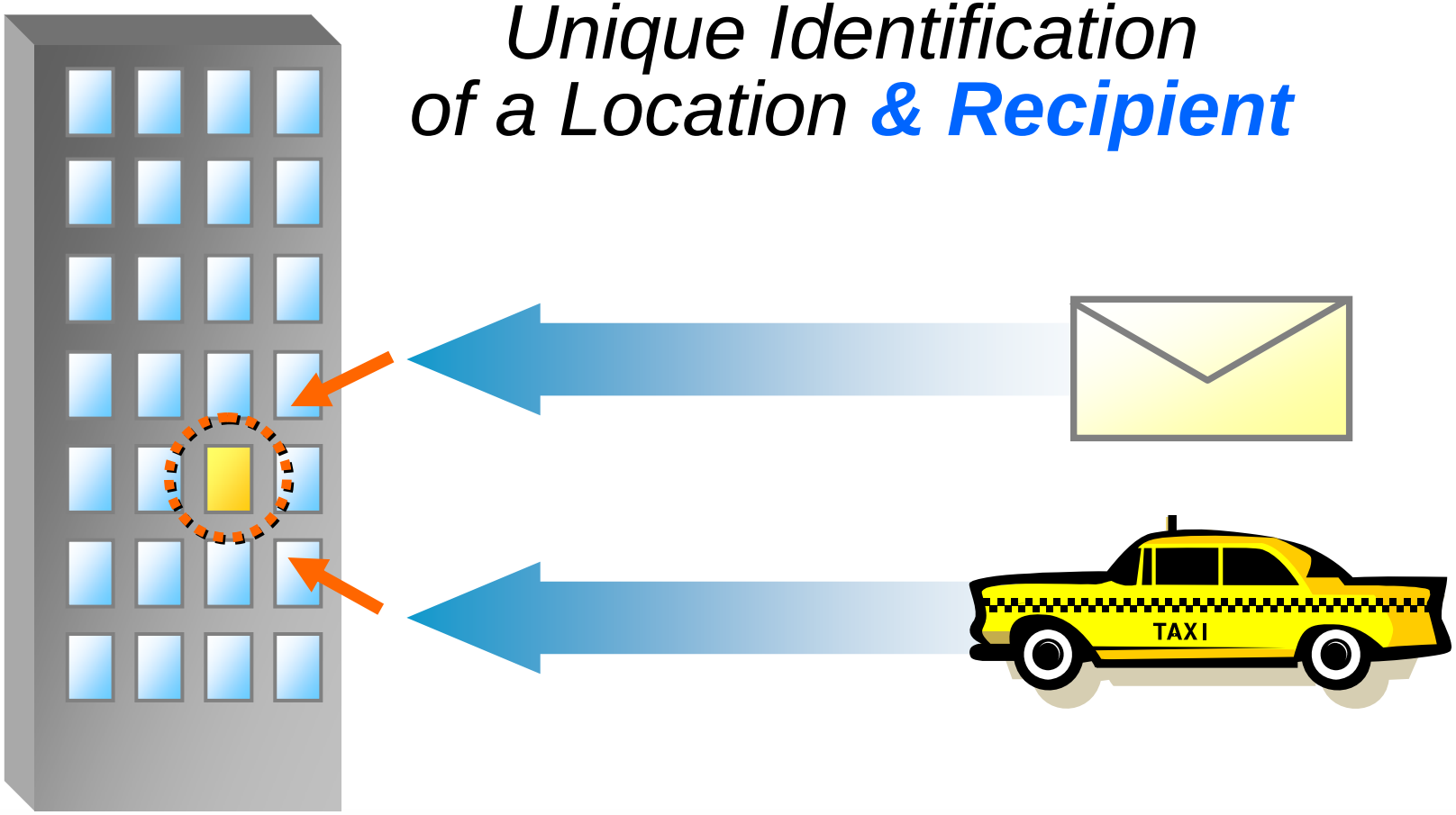
IP Addresses



Unique Identification of a Location

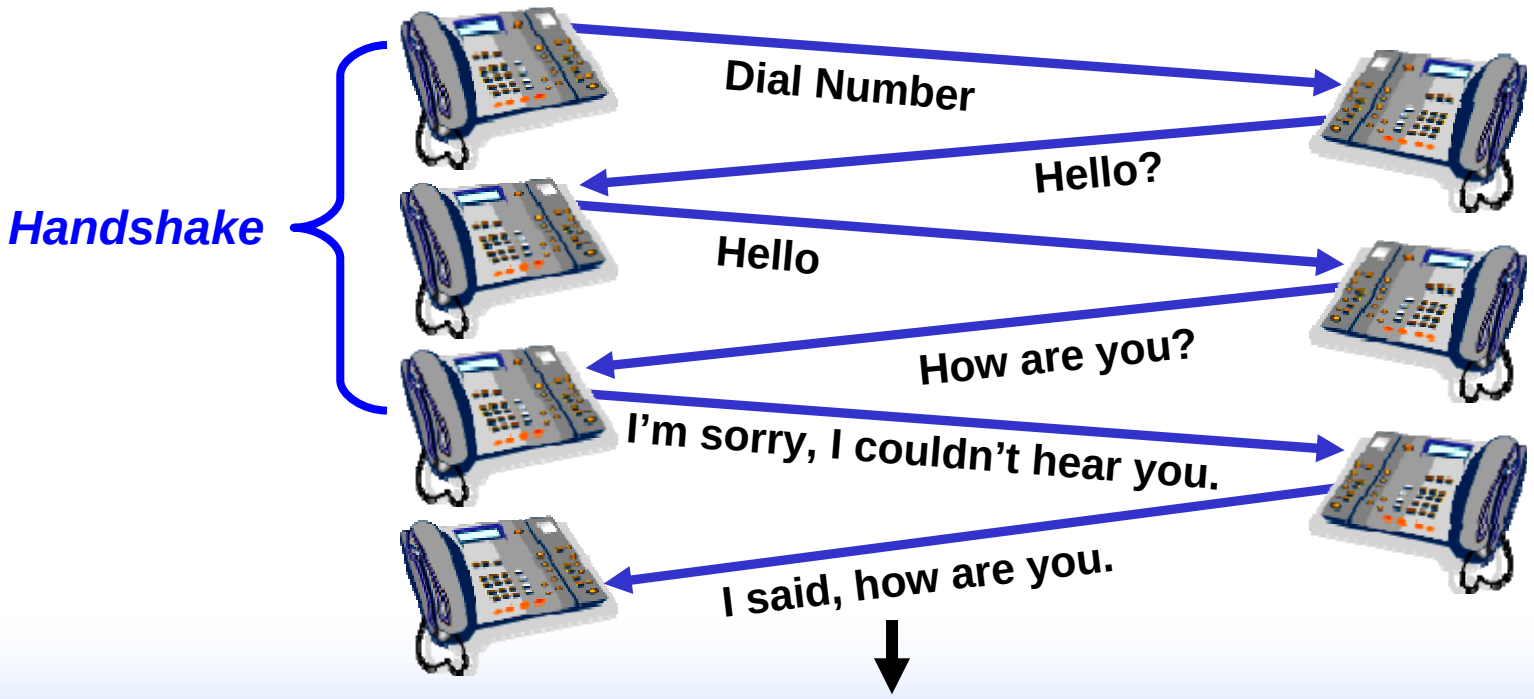
IP Ports

Unique Identification
*of a Location **& Recipient***



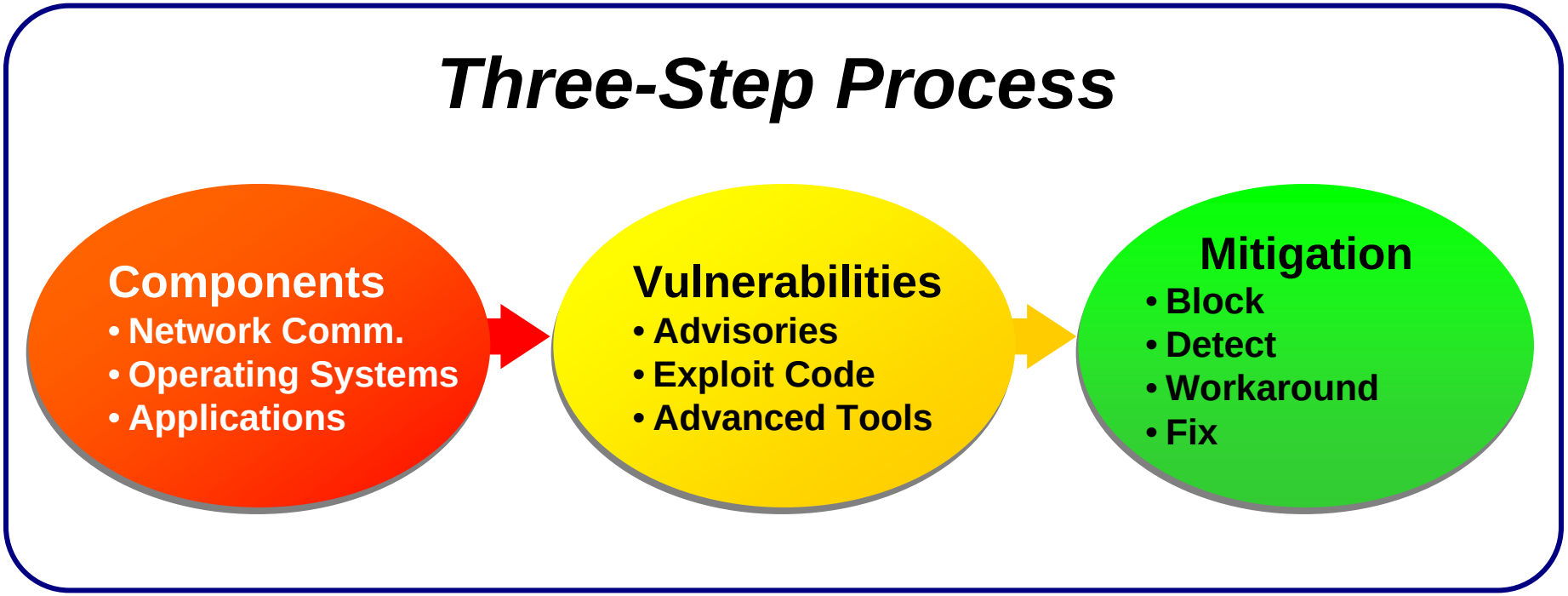
TCP Communication Basics

- **TCP Is a Reliable, Stateful Communications Protocol**
- **Three-Way Handshaking**
- **There Are No One-way Communications with TCP**

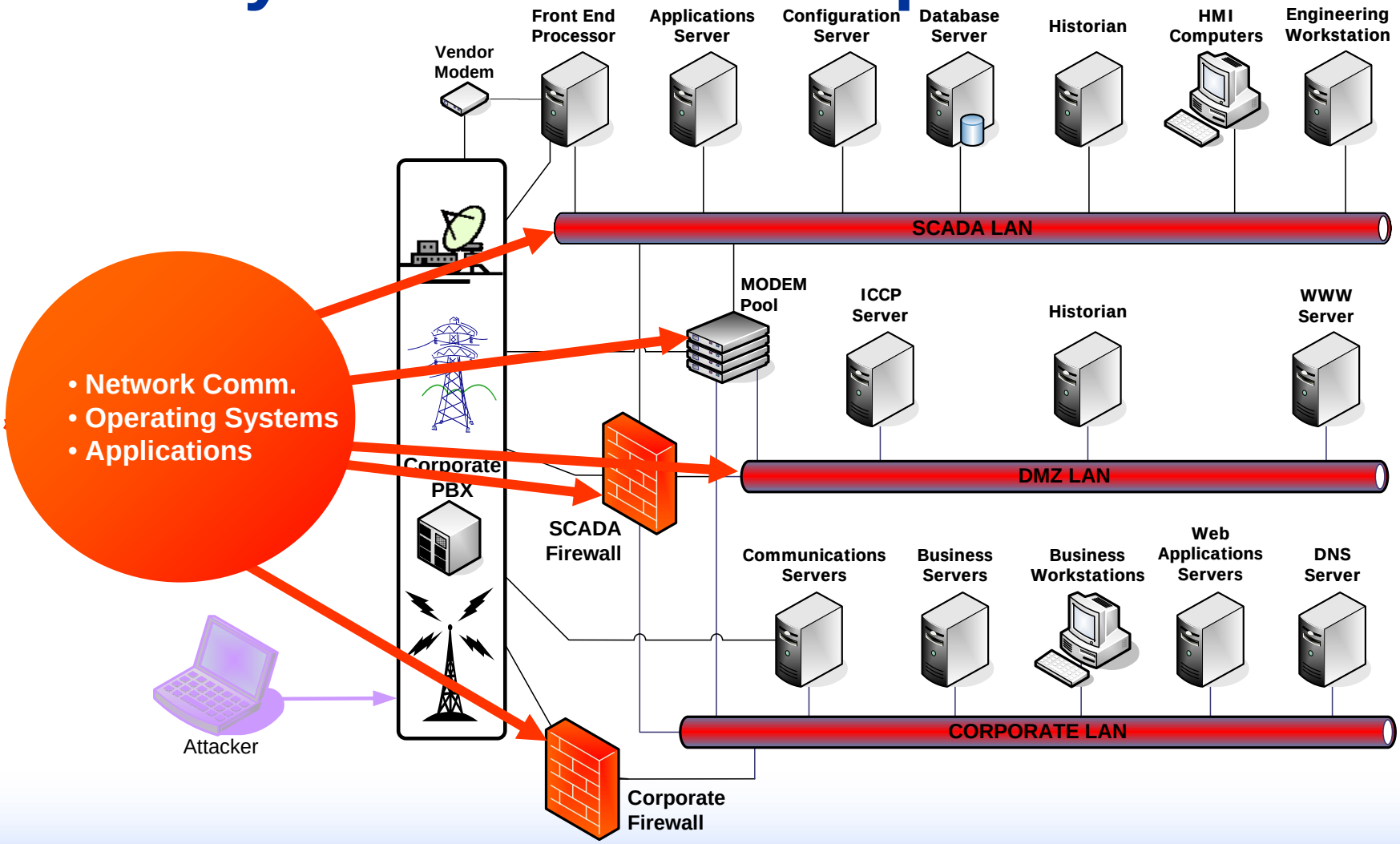


Vulnerability Reduction Process

Understanding Exposure

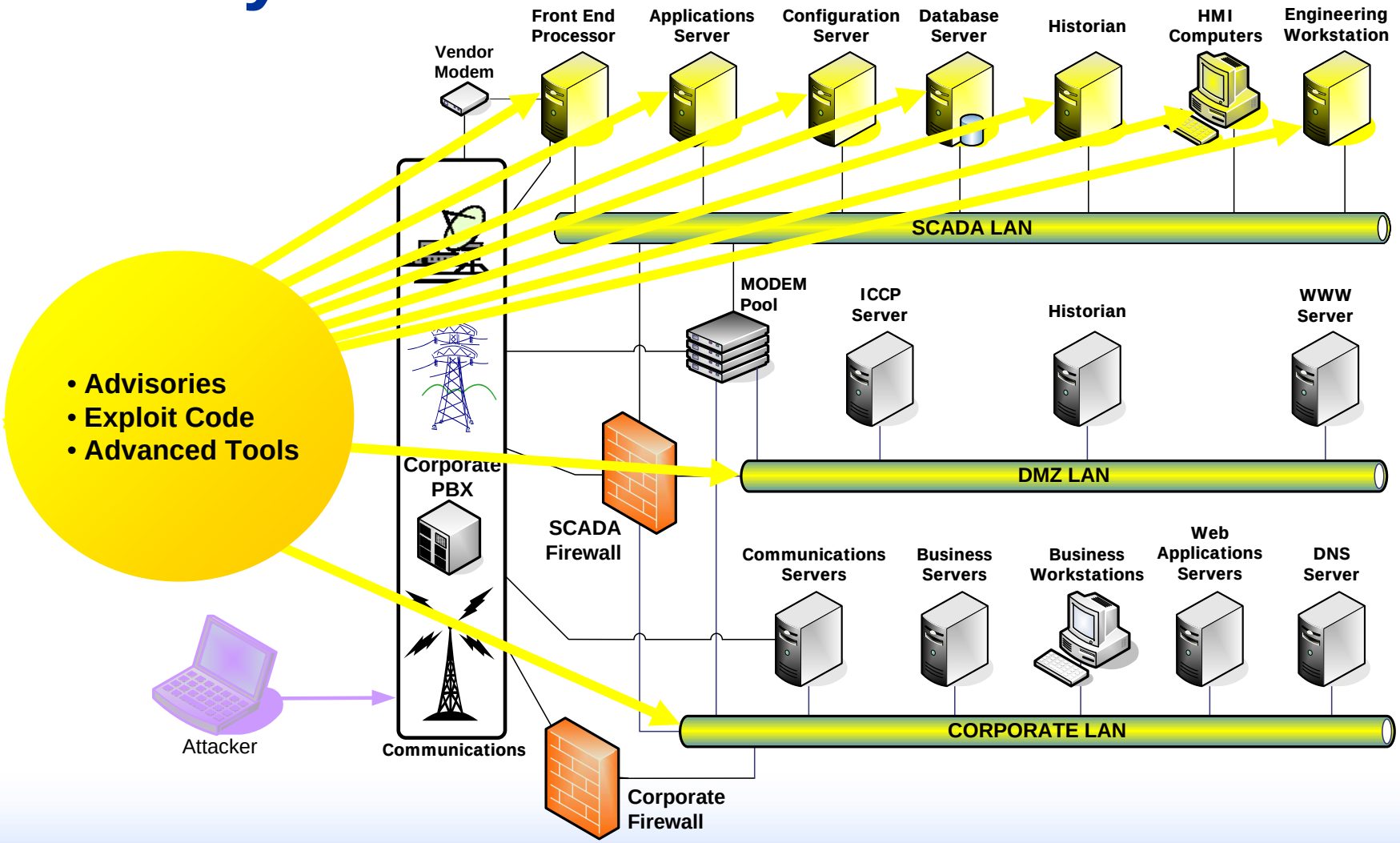


Identify Vulnerable Components



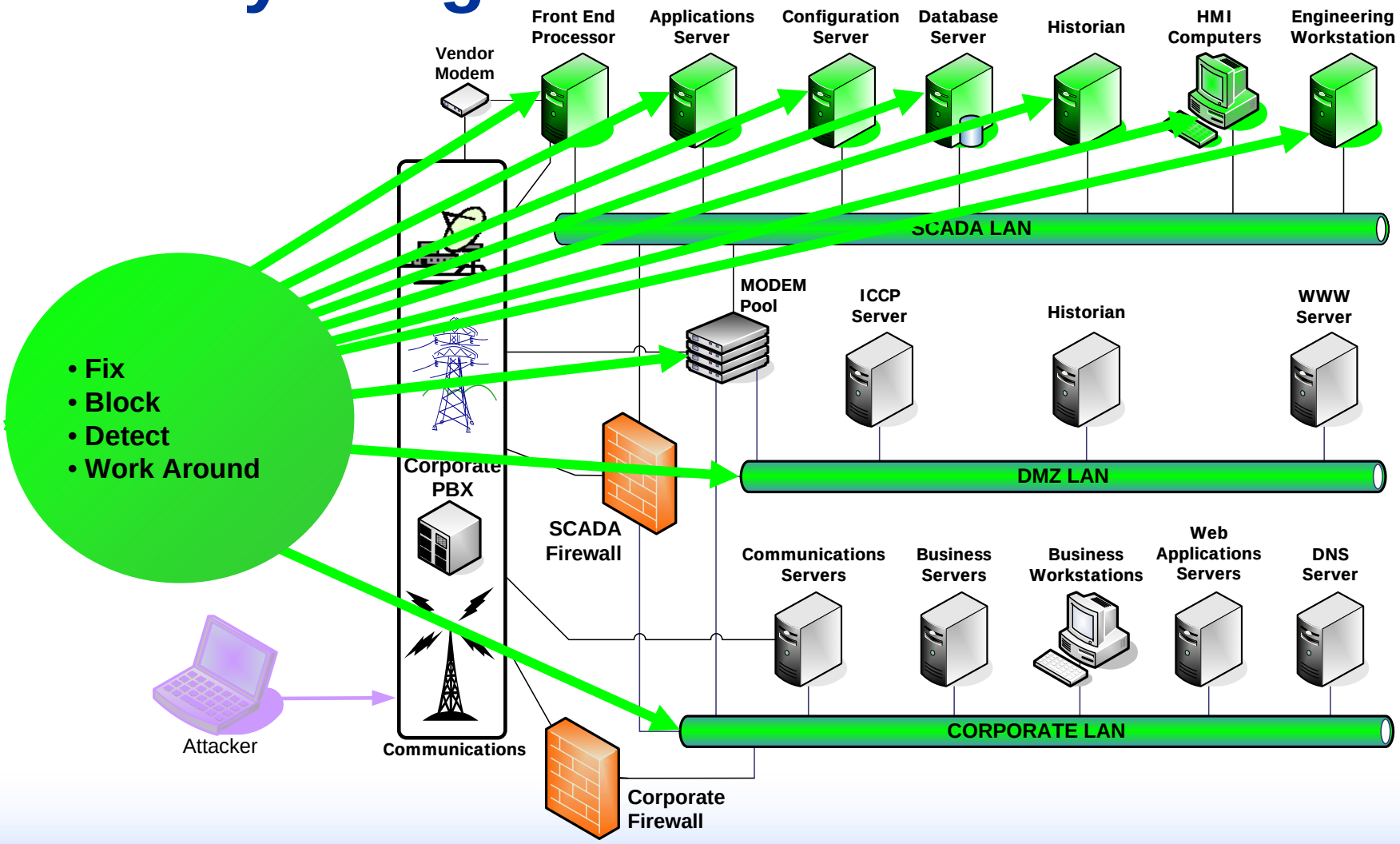
Idaho National Laboratory

Identify Threat Vectors



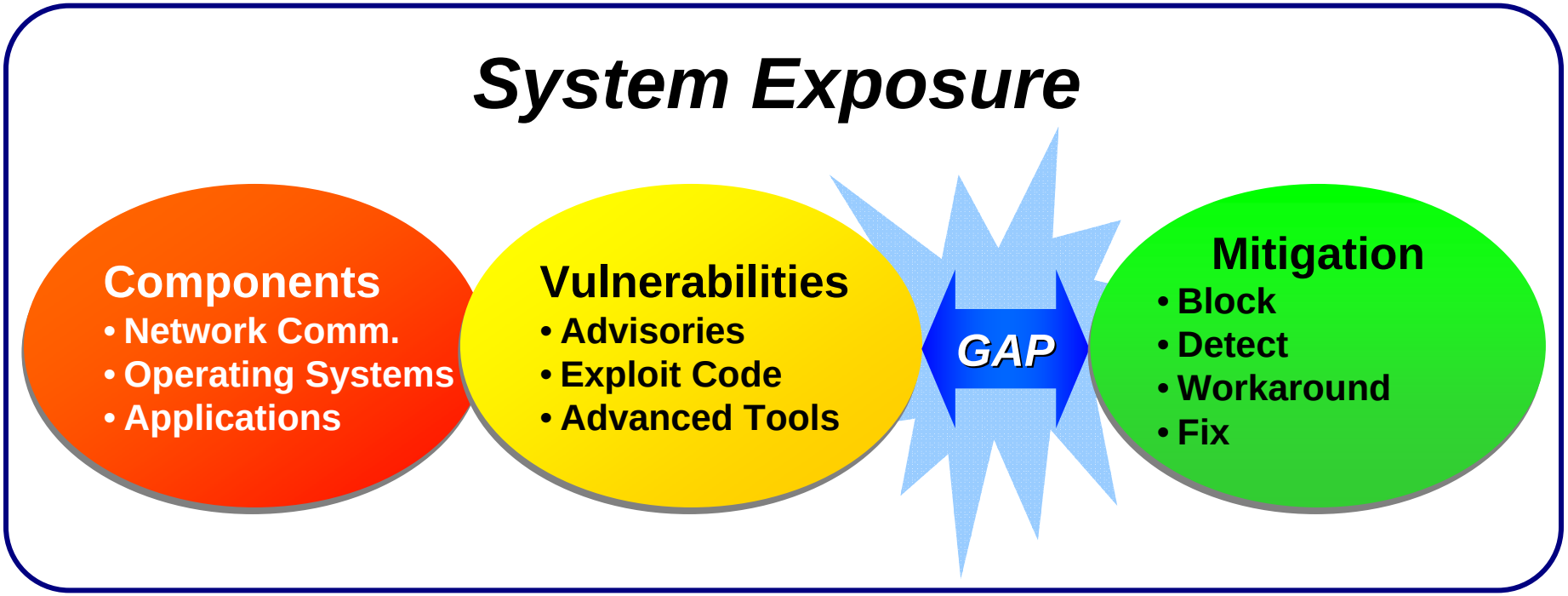
Idaho National Laboratory

Identify Mitigations



Idaho National Laboratory

Exposure


System Exposure
Components
• Network Comm.
• Operating Systems
• Applications
Vulnerabilities
• Advisories
• Exploit Code
• Advanced Tools
GAP
Mitigation
• Block
• Detect
• Workaround
• Fix

Break

Application Security Identification & Remediation (Corporate Network)

First Step – Identify Reality

Nmap is designed to allow system administrators & curious individuals to scan large networks to determine which hosts are up & what services they are offering.

A Fast & Informative Network Scanner that CAN Be Safely Used on isolated non-production SCADA/EMS Networks. *

Nmap Network Exploration

- Nmap was originally designed to be run from the command line (i.e. A Bash or DOS prompt)
- Some common Nmap options:
  - -sS TCP SYN Stealth Scanning (Default for root)
  - -sF TCP FIN Stealth Scanning
  - -sX Nmap Christmas Tree Scan (All TCP Flags Set)
  - -sN Null Stealth Scanning (No TCP Flags Set)
  - -sP Ping Sweep
  - -sV Enable Version Probing
  - -O OS Detection
  - -Tx Timing Mode (Polite & Sneaky are good)

Nmap Network Exploration

- Target hosts can be specified in many ways:
  - 10.4.4.1-255
    - All 255 possible IP addresses on this subnet
  - 10.4.4.0/24
    - Equivalent to the above but signifying a class C address block
  - 10.4.1-4.1-10
    - Ranges are allowed for subnets as well
  - 10.4.1.0/16
    - The 16-bit netmask will scan the entire class B address block

Nmap Network Exploration

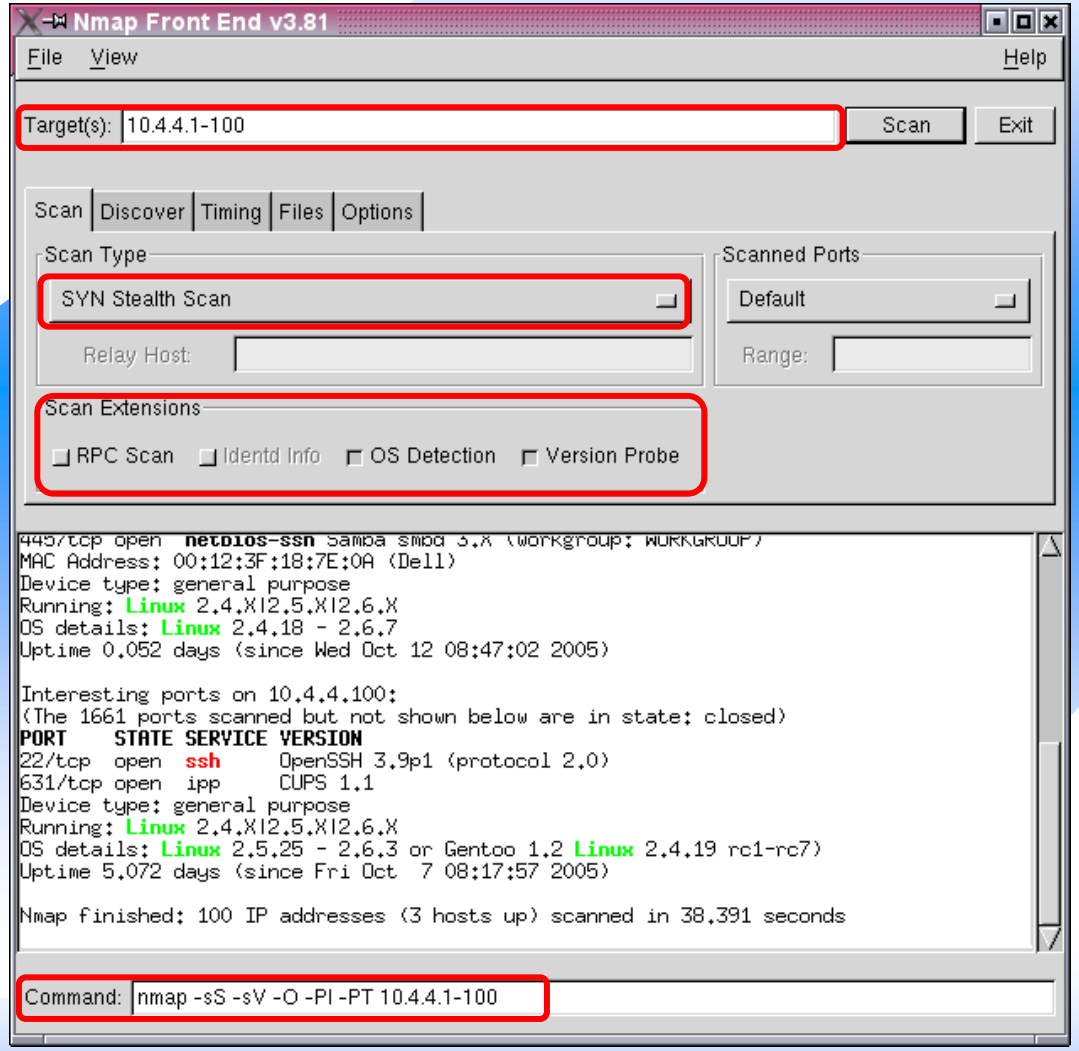
```
stimpy rohdkw # nmap -sP 10.4.4.1-100

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-10-12 09:48 MDT
Host 10.4.4.20 appears to be up.
MAC Address: 00:60:B0:03:C8:70 (Hewlett-packard CO.)
Host 10.4.4.50 appears to be up.
MAC Address: 00:12:3F:18:7E:0A (Dell)
Host 10.4.4.100 appears to be up.
Nmap finished: 100 IP addresses (3 hosts up) scanned in 3.610 seconds
stimpy rohdkw # nmap -sS -sV -O 10.4.4.1-100

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-10-12 09:49 MDT
Interesting ports on 10.4.4.20:
(The 1660 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE    VERSION
23/tcp    open  telnet     HP JetDirect printer telnetd
515/tcp   open  printer?
9100/tcp open  jetdirect?
MAC Address: 00:60:B0:03:C8:70 (Hewlett-packard CO.)
Device type: printer
Running: HP embedded
OS details: HP printer w/JetDirect card

Interesting ports on 10.4.4.50:
(The 1660 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 3.9p1 (protocol 2.0)
139/tcp open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
MAC Address: 00:12:3F:18:7E:0A (Dell)
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.18 - 2.6.7
Uptime 0.044 days (since Wed Oct 12 08:47:02 2005)

Interesting ports on 10.4.4.100:
(The 1661 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 3.9p1 (protocol 2.0)
631/tcp open  ipp     CUPS 1.1
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.5.25 - 2.6.3 or Gentoo 1.2 Linux 2.4.19 rc1-rc7)
Uptime 5.064 days (since Fri Oct  7 08:17:57 2005)

Nmap finished: 100 IP addresses (3 hosts up) scanned in 37.592 seconds
stimpy rohdkw #
```

Idaho National Laboratory

Nmap Network Exploration

- **Fortunately there is a GUI for Nmap so that you don't need to memorize all of the options**
- **Sorry Windows users... You are stuck with the command line version**

Nmap Network Exploration



Idaho National Laboratory

Nmap Network Exploration

- **Nmap is on the live CD**
- **Execute Nmap with the following commands:**
  - **nmap -sP 192.168.1.0/24**
  - **nmap -sS -sV -O 192.168.1-100**

Second Step – Tcpdump Discover Communications

"Tcpdump prints out the headers of packets on a network interface that match the boolean expression. It can also be run with the -w flag, which causes it to save the packet data to a file for later analysis, and/or with the -r flag, which causes it to read from a saved packet file rather than to read packets from a network interface. In all cases, only packets that match the expression will be processed by tcpdump." www.tcpdump.org

A Very Efficient & Clean Way for Creating a Customized "Wire Tap" on Your Network.

Second Step – Ethereal Discover Communications

Ethereal is a GUI network protocol analyzer. It lets you interactively browse packet data from a live network or from a previously saved capture file. Ethereal's native capture file format is libpcap format, which is also the format used by tcpdump & various other tools.

Ethereal is THE Standard for Performing Network Protocol Analysis.

TCPDump & Ethereal

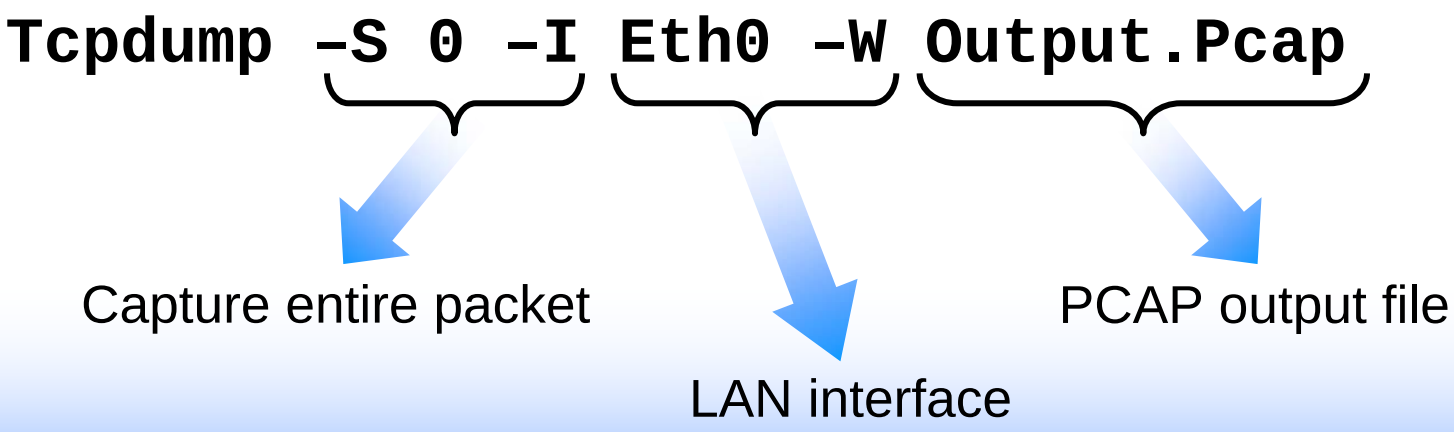
- Some common options for TCPdump:
  - -s <len>
    - The snap length of the packet capture
  - -C <size>
    - Limit output file to size (in MB)
  - -F <file>
    - Input filter file
  - -i <lan>
    - Network interface to sniff
  - -w <file>
    - Output PCAP file

TCPDump & Ethereal

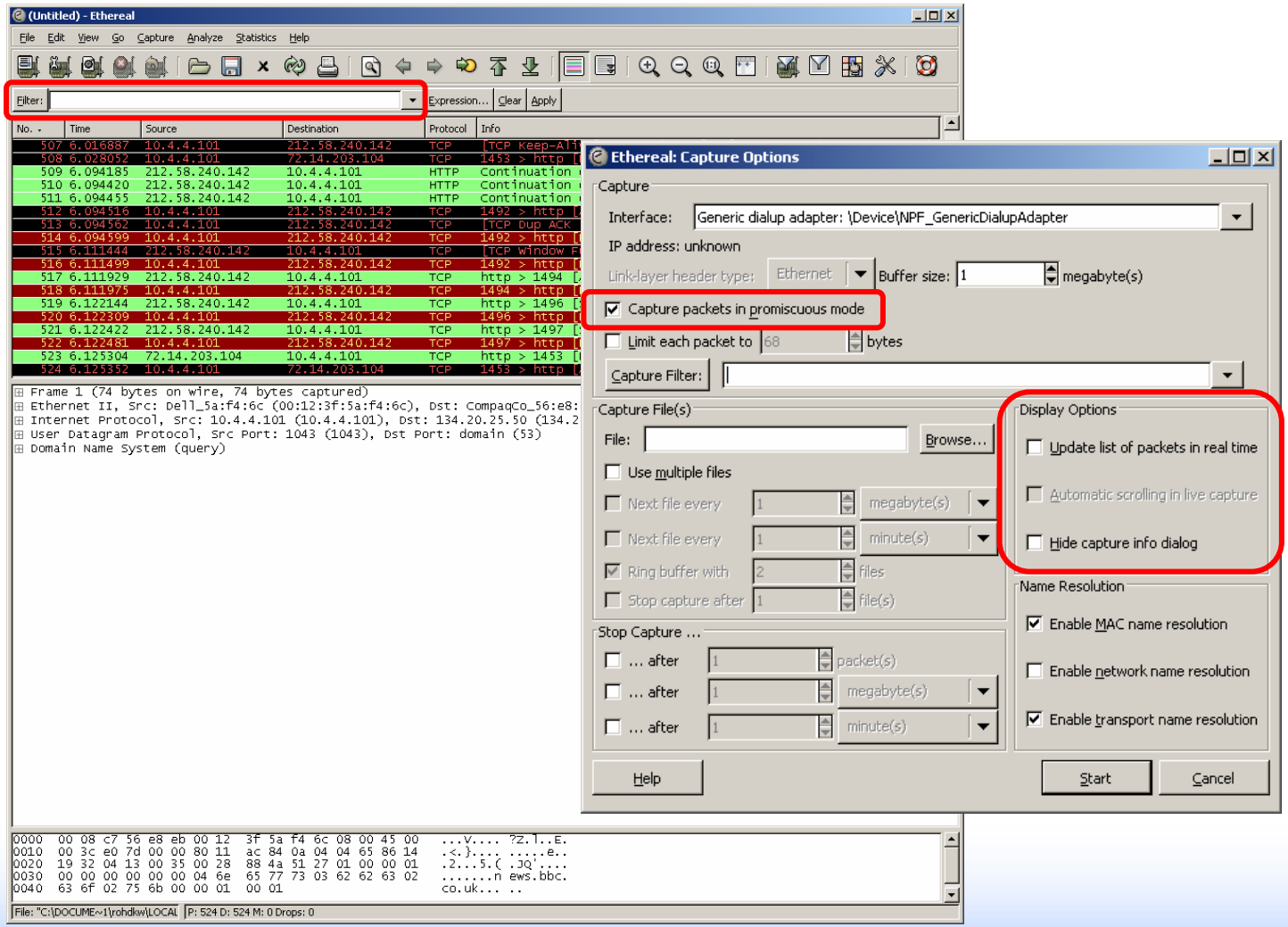
- **Some common options for Ethereal:**
  - **Promiscuous Mode**
  - **Update list of packets in real time**
  - **Automatic scrolling in live capture**
  - **Hide capture info dialog**

TCPDump & Ethereal

We'll First Run A TCPdump Session & Capture a Few Minutes Worth of Data



TCPDump & Ethereal



Idaho National Laboratory

Nessus Security Scanner

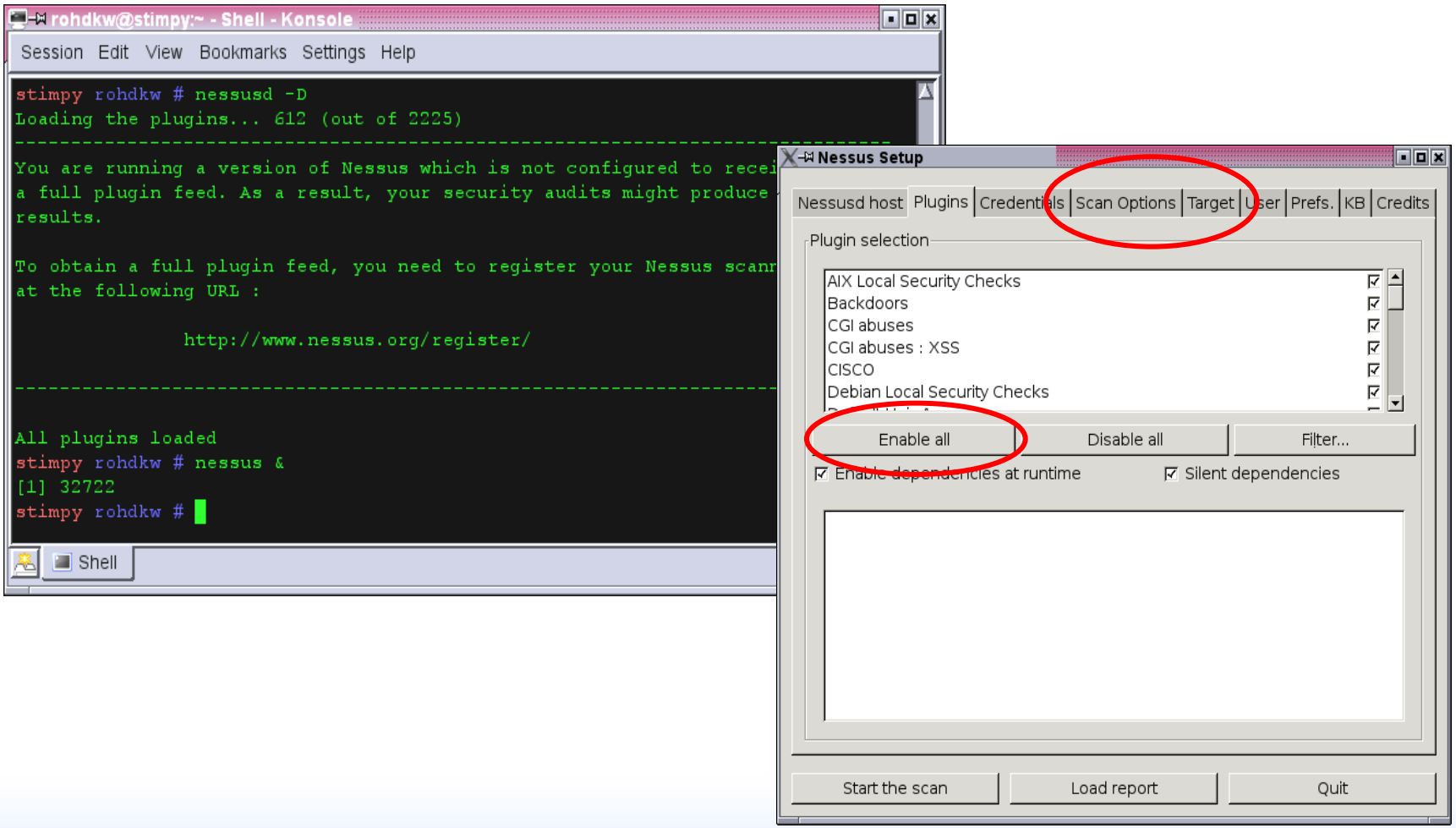
The Nessus Security Scanner
is a Security Auditing Tool Made Up of Two Parts:

Server	***Client***
The Server, Nessusd is in Charge of the Attacks	The Client Nessus Provides An Interface to the User.”

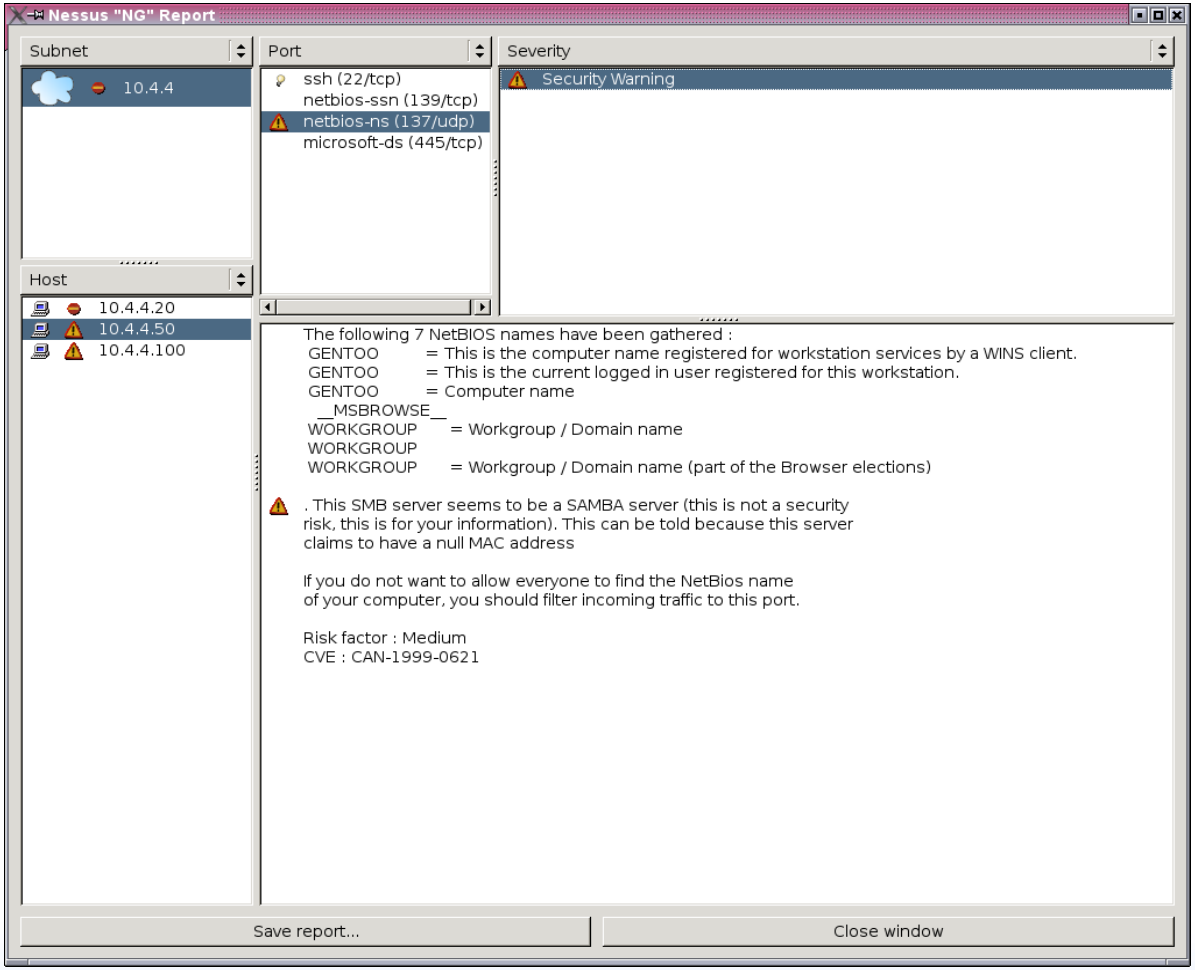
Nessus is the Defacto Standard for (Free)
Open Source Network Vulnerability Scanners

This tool can be dangerous *

Nessus Security Scanner



Nessus Security Scanner



Application Security

Find It

- Be “curious” about software used on your systems
  - play the idiot or the bad guy
- Analyze what applications & services are available on your critical networks
- Check database user privileges & database service configuration
- Examine firewall rules for replication

Application Security

Find It

- Common tools used when analyzing application security:
- Curiosity
- Ethereal &/or TCPDump
- Strings
- Source Code Analysis like RATS & Flawfinder
- IDAPro (Very Advanced)

Application Security

Find It

- Examine the communication protocols in use
  - Database Traffic
  - Proprietary Traffic
- Advanced students may wish to create a “protocol fuzzer” to see what happens when sending unexpected input over the network

Application Security

Fix It

- The major flaws with this application were:
  - Database userid & password exposed by application
  - Database userid has way too many privileges on the system
  - Database is running as a privileged user
  - Proprietary protocol encoder/decoder flaws (server crashes)

Application Security

Fix It

- This is the perfect place for well defined policy. Applications introduced into the DMZ potentially create a means for hackers to gain unauthorized access to a system. Ensure that your policy checks for the following before any application is developed or installed:
  - Code reviews for good/secure programming practices
  - Implementing “least privileges” policy for database users & the database service
  - Encrypt link(s) between the SCADA & the DMZ
  - Implement smart Network Intrusion Detection
  - Implement monitoring/control to prevent man in the middle attacks
  - Outbound firewall rules will prevent the rootkit from calling home

Least Privileges

"The principle of least privilege requires that a user be given no more privilege than necessary to perform a job. Ensuring least privilege requires identifying what the user's job is, determining the minimum set of privileges required to perform that job, & restricting the user to a domain with those privileges & nothing more. By denying to subjects transactions that are not necessary for the performance of their duties, those denied privileges cannot be used to circumvent the organizational security policy.[1] "

1. Integrity in Automated Information Systems. National Computer Security, Center, September 1991.

Least privileges may not be possible due to technology limitations
User may be a computer

Least Privileges

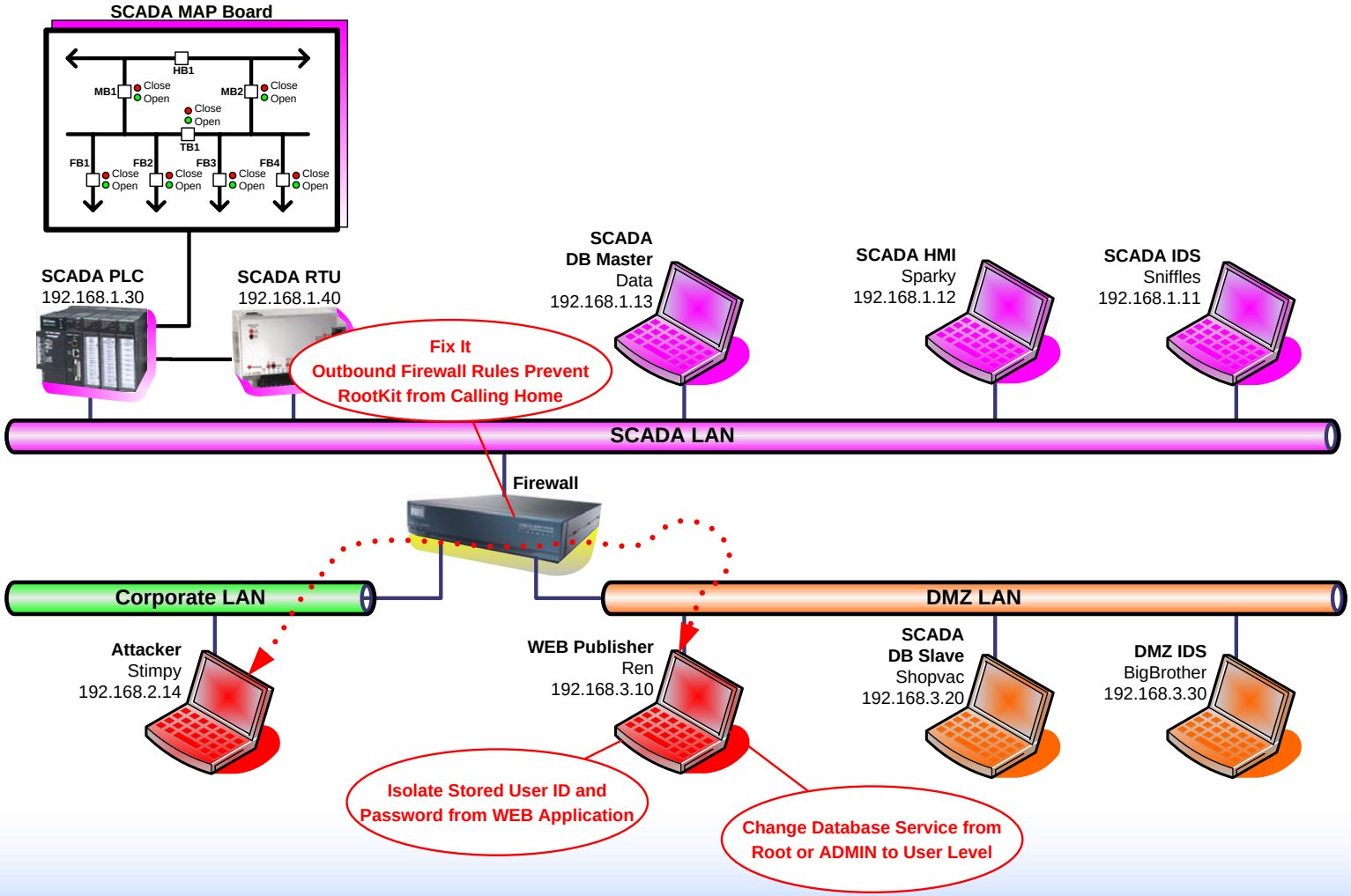
An Important Note with Respect to Least Privileges:

This methodology does not remove vulnerabilities from a system. It only prevents exploitation from obtaining immediate superuser access.

Administrators still need to care for their systems to prevent ***escalation of privileges*** when unauthorized access is gained.

Application Security

Fix It



Idaho National Laboratory

Application Security

Relate It

- **Excerpts from NERC's Top 10 Vulnerability List**
  - **Software used in control systems is not adequately scrutinized, & newer systems include extraneous vulnerable software.**
  - **Installation of inappropriate applications on critical systems.**
  - **Poorly designed control system networks that fail to employ sufficient defense-in-depth mechanisms.**
  - **Policies, procedures & culture governing control system security are inadequate & lead to lack of executive management buy in. In addition, personnel routinely ignore or lack training in policies & procedures to protect the control systems.**

Application Security

Test It

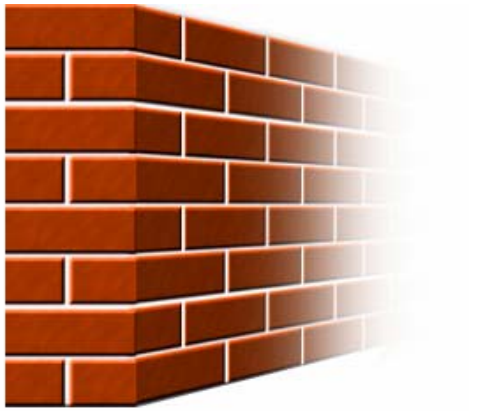
- Let's test our newly implemented mitigation techniques to see if our system is more secure.
  - Is the database still functional? How about the replication?
  - Can Snort rules be created to monitor this activity?
  - Is Arpwatch able to detect the MITM on the DMZ?
  - Does the addition of outbound filtering on the firewall prevent normal operations?

Break

SQL Injection Identification & Remediation (DMZ Network)

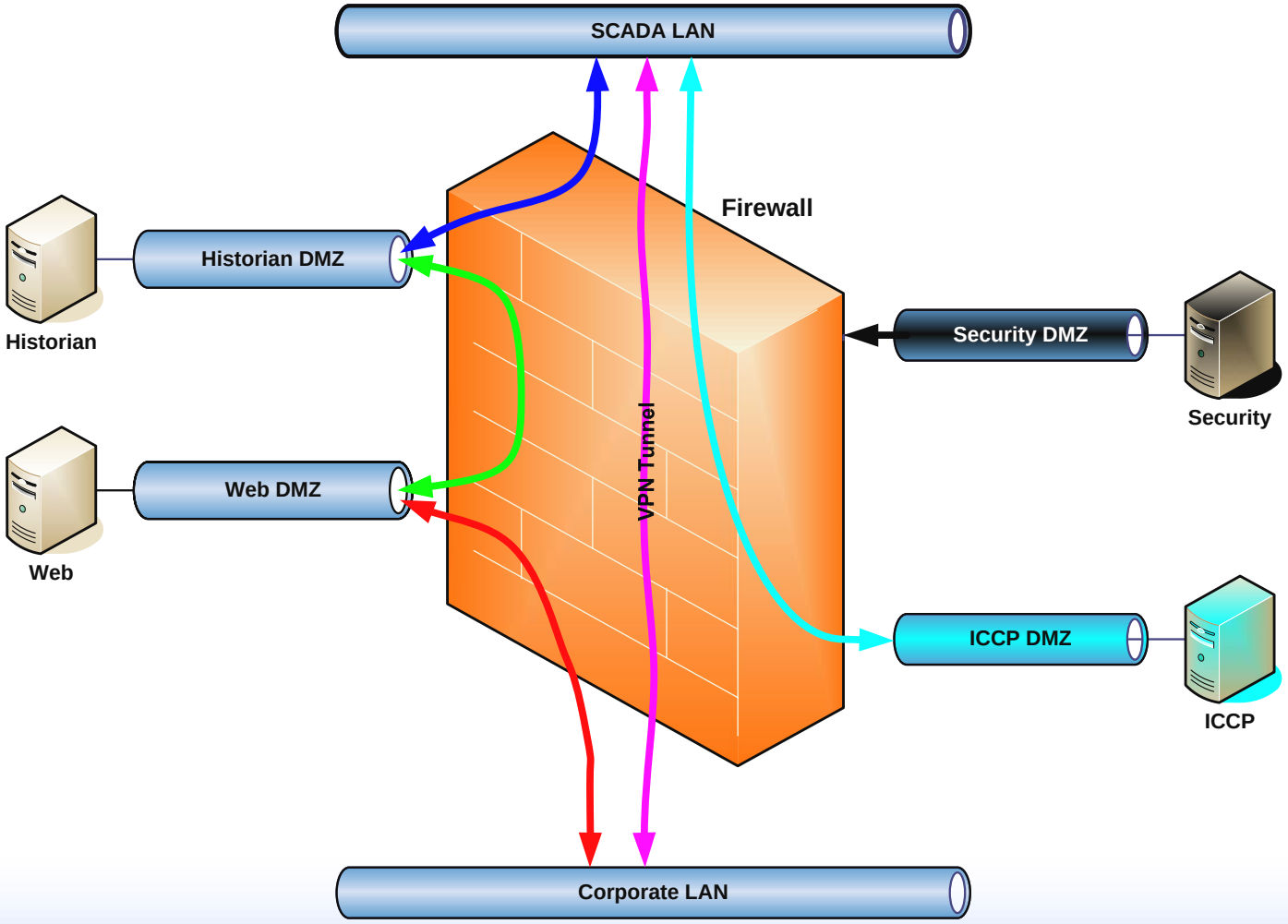
First Step

Understand the Firewall

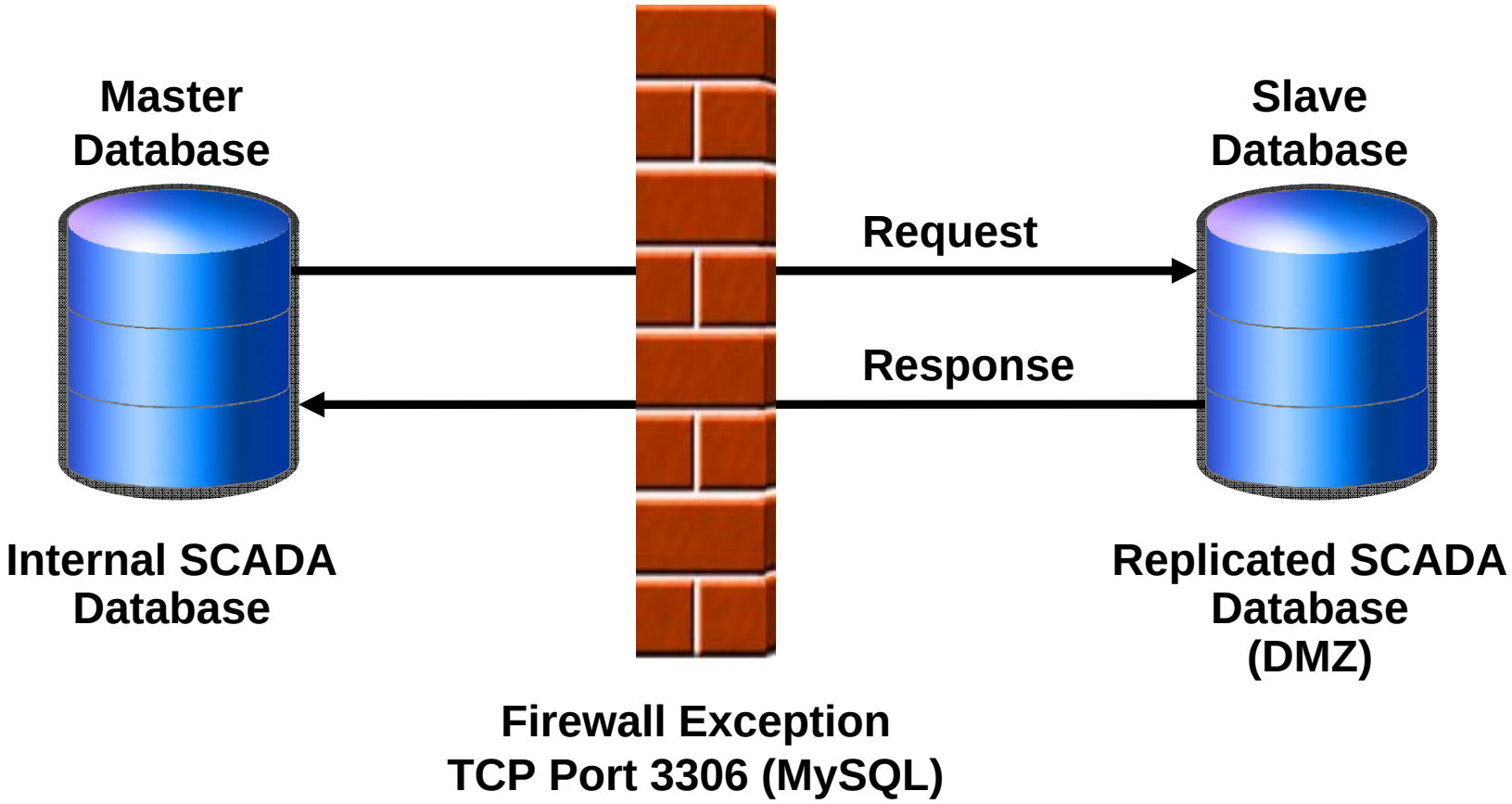


- **To fix the problems on our DMZ we need to first understand how the firewall is configured**
- **The exploit used to move from the DMZ to the SCADA network involved exploiting a database exception in the firewall**
- **Use the tools we've demonstrated thus far to watch the communications into the SCADA network**

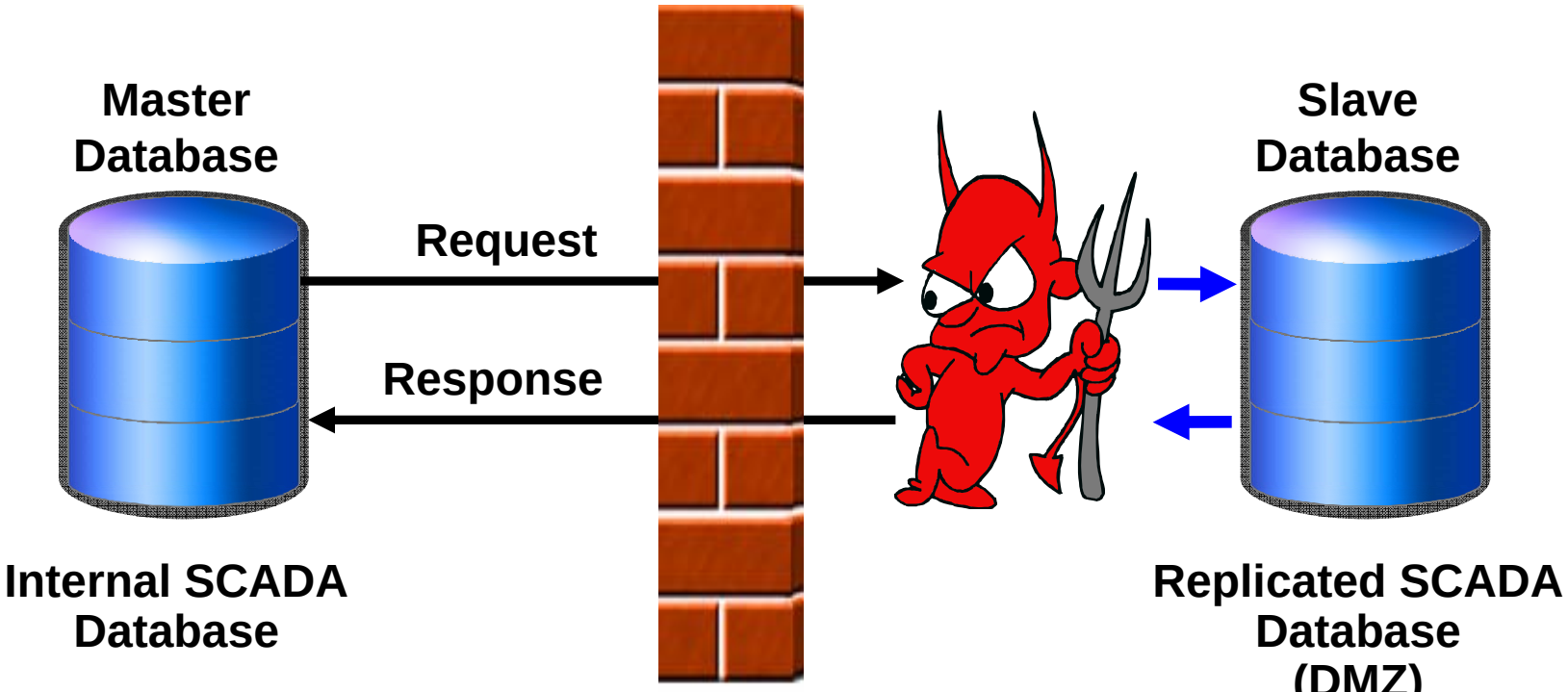
Segmented Firewall



Firewall Exception – Database

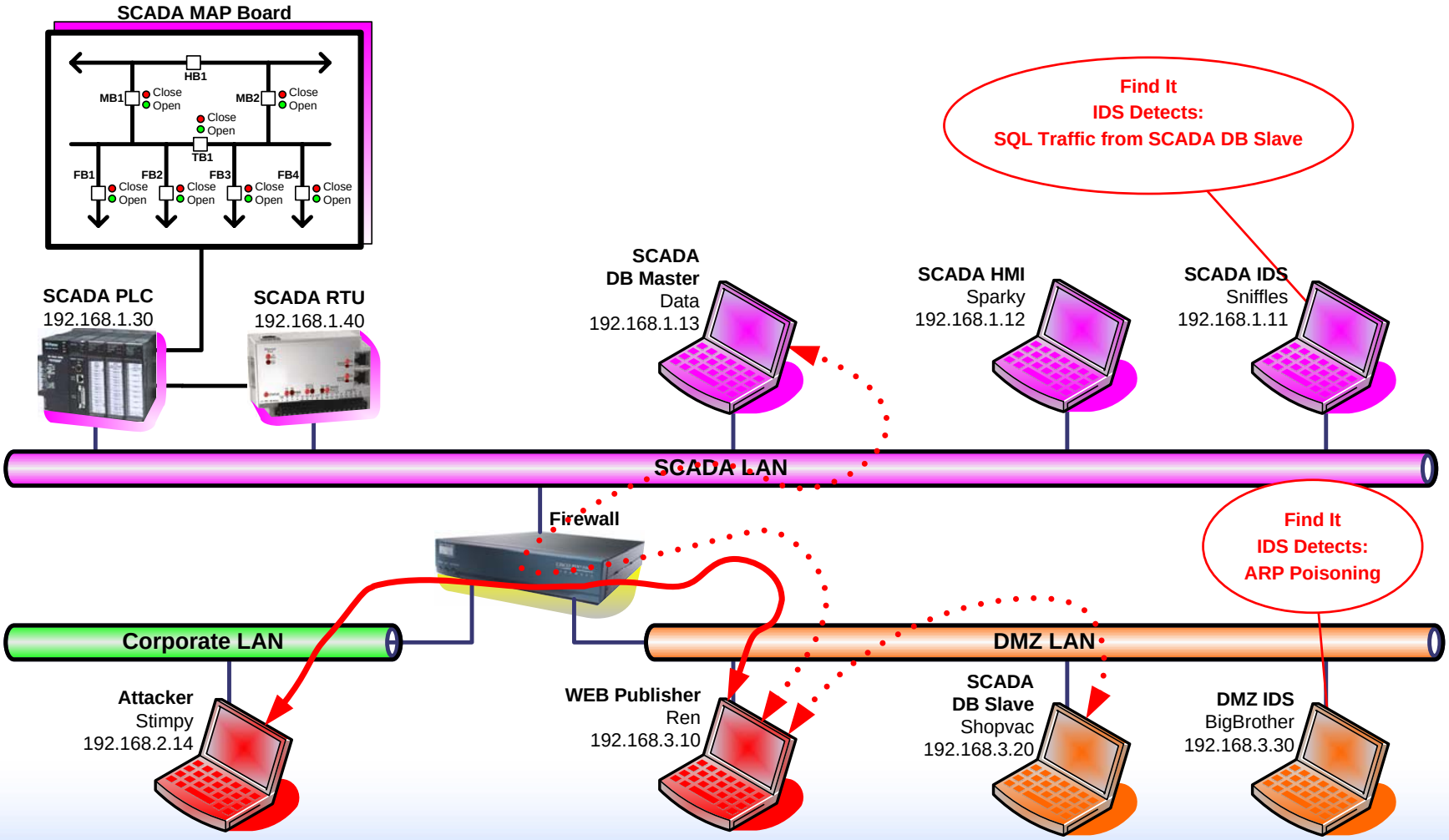


Firewall Exception – Database



SQL Injection

Find It



Idaho National Laboratory

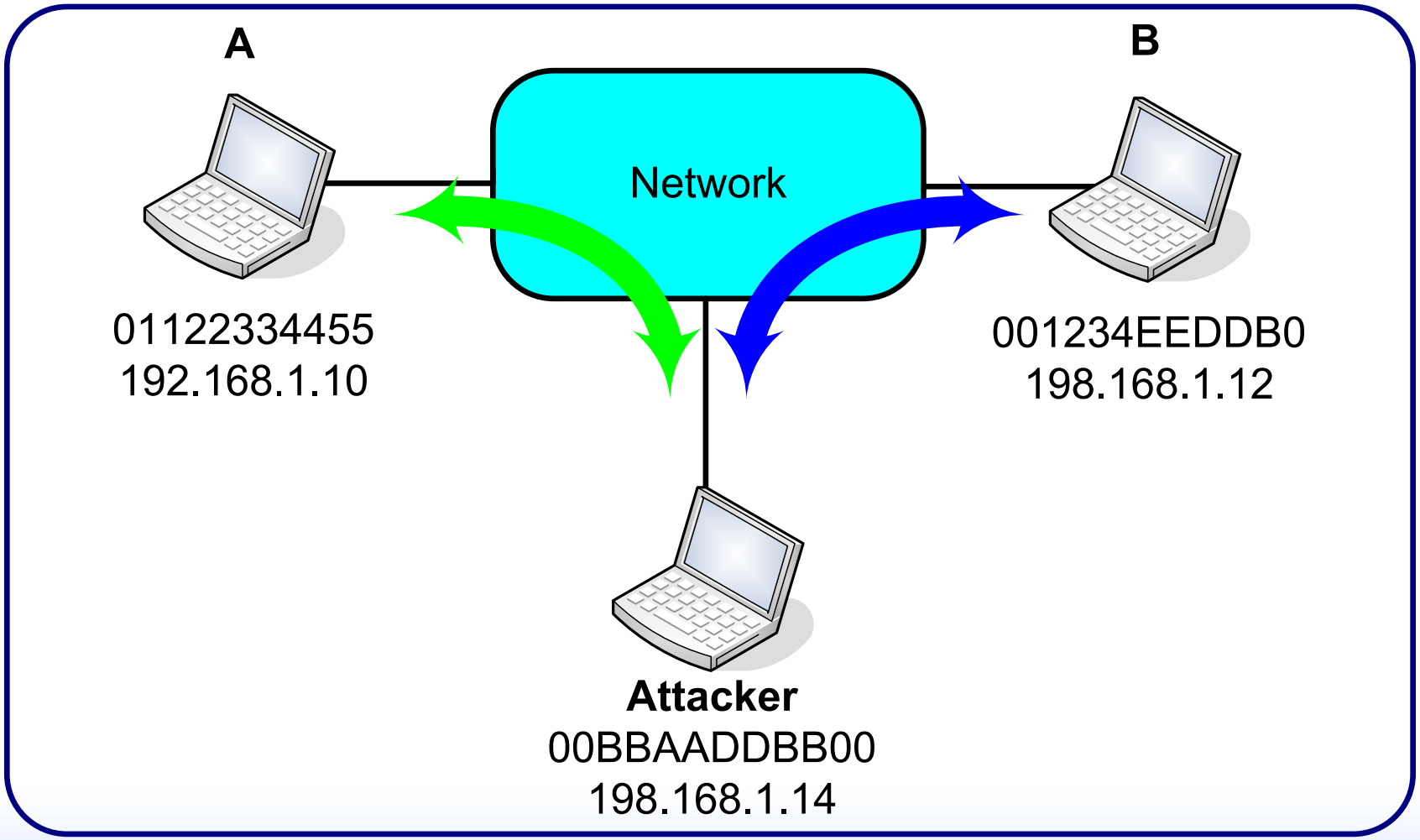
Second Step

Monitor Exceptions

- **Exceptions should be closely monitored using syslog & IDS, but another weakness is spoofing or MITM**
- **"Arpwatch keeps track of ethernet/ip address pairings. It syslogs activity & reports certain changes via email. Arpwatch uses pcap to listen for arp packets on a local ethernet interface."** **Linux man page**

This Is a Good Simple Solution for Monitoring the Network for Potential ARP MITM Attacks WITHOUT Having to Hardcode ARP Addresses

ARP Poisoning



Arpwatch

- Arpwatch is typically configured to send alerts to syslog & local email. Some common configuration options include:
- -m
  - Specify the email address of where to send alerts
- -u
  - userid & groupid for Arpwatch (least privileges)

Third Step

Prevent Data Overload

Swatch Perl script utility is designed to monitor system activity. In order for Swatch to be useful, it requires a configuration file which contains pattern(s) to look for & action(s) to perform when each pattern is found.

Swatch Is Extremely Useful for Harvesting Important Information from Log Files As New Data Is Entered by Applications Such As Snort & Arpwatch.

Swatch Can Also Be Used to Create Application Log Intrusion Detection

Swatch

```
*** swatch version 3.1 (pid:7619) started at Tue Feb 21 08:07:13 MST 2006

Tue Feb 21 08:32:07 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**]
Tue Feb 21 08:33:47 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 26 times)
Tue Feb 21 08:34:56 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 352 times)
Tue Feb 21 08:38:24 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 28 times)
Tue Feb 21 08:39:53 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 32 times)
Tue Feb 21 08:40:51 MST 2006
[**] [1:1000000:1] Not HMI talking to PLC [**]
Tue Feb 21 08:40:54 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 451 times)
Tue Feb 21 08:42:08 MST 2006
[**] [1:1000003:1] Not slave db talking to master [**] (seen 92 times)
Tue Feb 21 08:42:23 MST 2006
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
Tue Feb 21 08:42:40 MST 2006
[**] [1:1000000:1] Not HMI talking to PLC [**] (seen 146 times)
Tue Feb 21 08:43:20 MST 2006
```

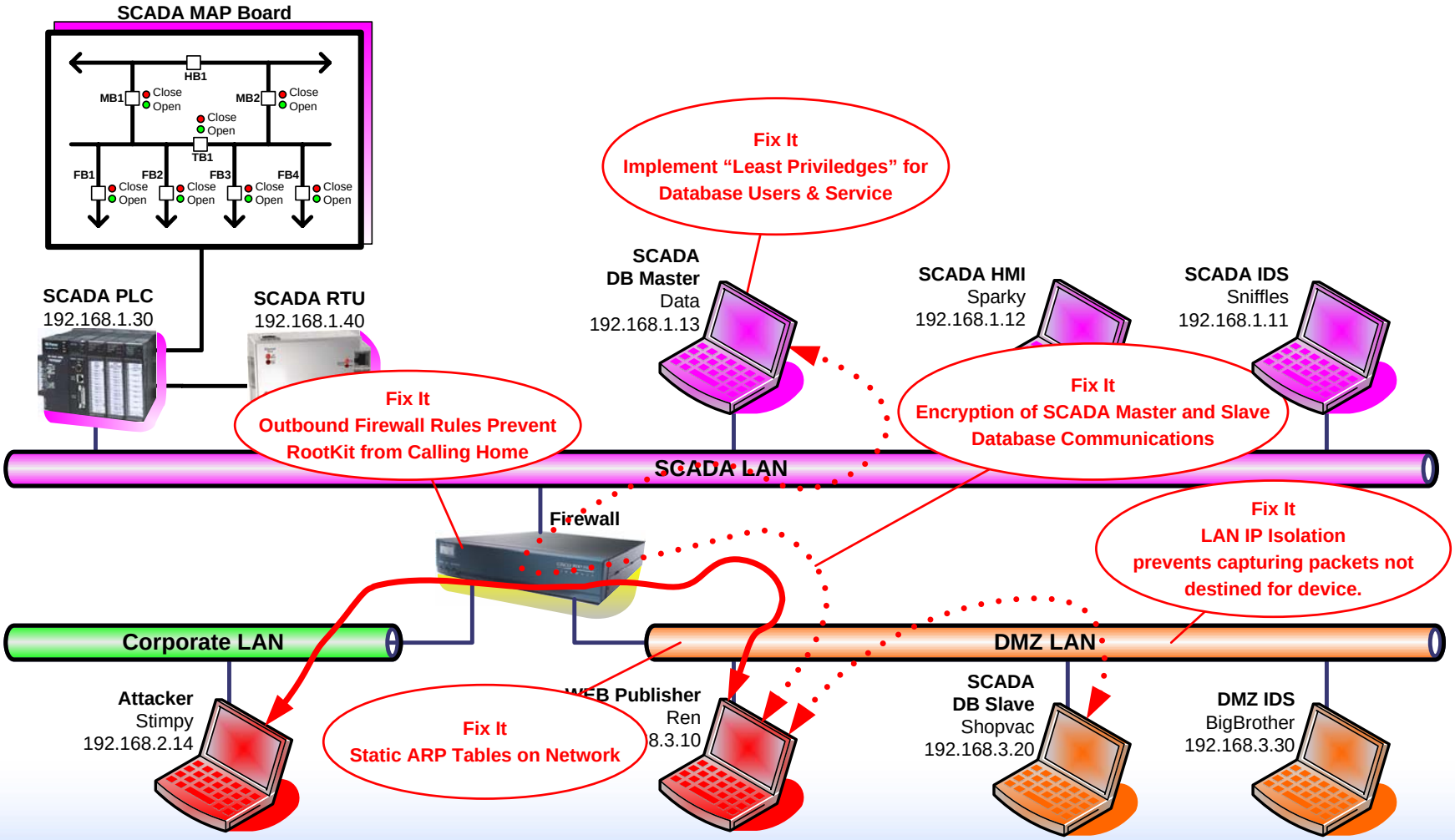
Swatch is easily configured to monitor multiple data sources in real-time.

- **-c Specify the configuration file to use**
- **-t Specify the log file to monitor**

Swatch uses simple regular expressions to monitor lines of text as they are added to log files.

SQL Injection

Fix It



Idaho National Laboratory

SQL Injection

Relate It

Excerpts from NERC's Top 10 Vulnerability List

- **Control systems data sent in clear text.**
- **Lack of quick & easy tools to detect & report on anomalous or inappropriate activity. Non existent forensic & audit methods.**
- **Poorly designed control system networks that fail to employ sufficient defense-in-depth mechanisms.**
- **Policies, procedures & culture governing control system security are inadequate & lead to lack of executive management buy in. In addition, personnel routinely ignore or lack training in policies & procedures to protect the control systems.**

SQL Injection

Test It

- Let's test our newly implemented mitigation techniques
  - Is the database still functional?
  - How about the replication?
  - Does the additional outbound filtering on the firewall prevent normal operations?
  - Can IDS rules be created to monitor this activity?
  - Is the IDS able to detect the MITM on the DMZ?

Break

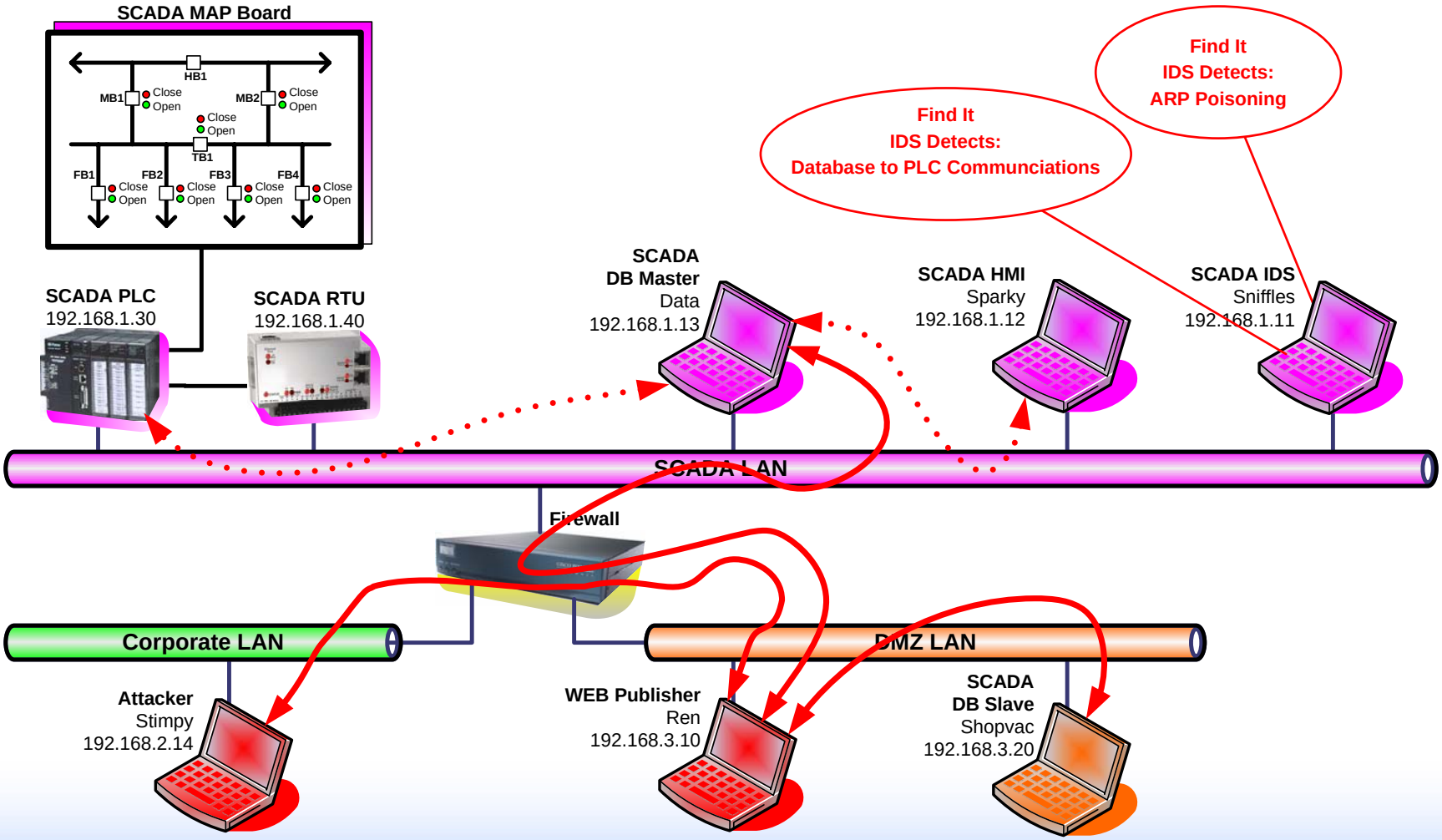
Unauthorized Control Identification & Remediation (Control Network)

First Step – Identify Reality

- Using the tools and skills developed thus far, discover the process and protocols currently in use
- Develop a “reality diagram” of the control system network

Unauthorized Control

Find It



Idaho National Laboratory

Second Step – Monitor Communications

"Snort is an open source network intrusion prevention system (IPS) capable of performing real-time traffic analysis and packet-logging on IP networks. It can perform protocol analysis, content searching & matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts and more.

Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that uses a modular plug-in architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user-specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump, a packet logger (useful for network traffic debugging and so), or as a full-blown network intrusion prevention system."

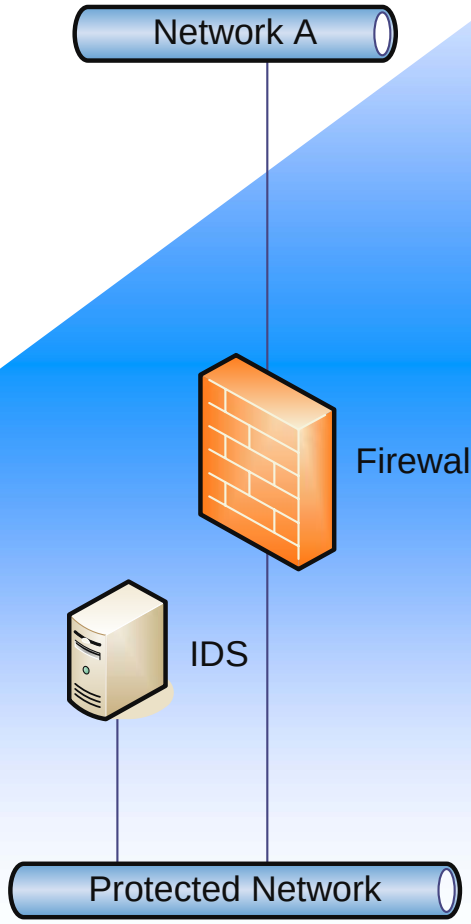
www.webopedia.com

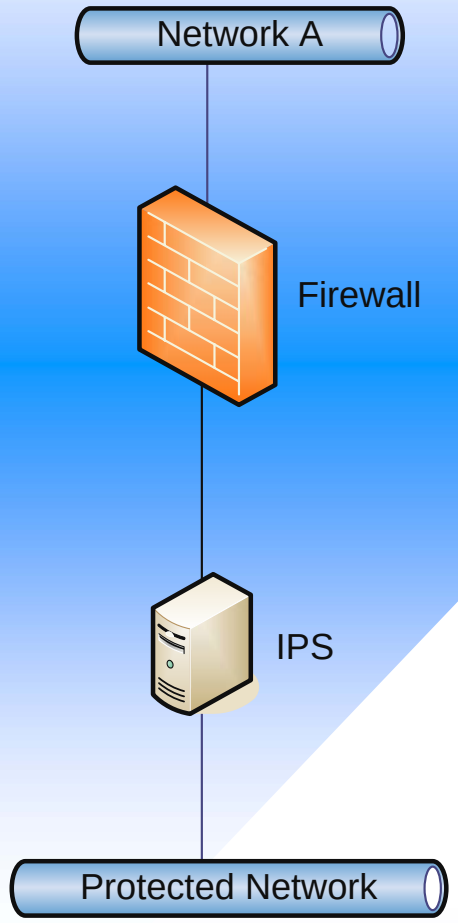
Network Intrusion Detection Is a Great Way of Monitoring What Communication You KNOW Should Be ALLOWED on Your Network.

Snort Rule Expectations

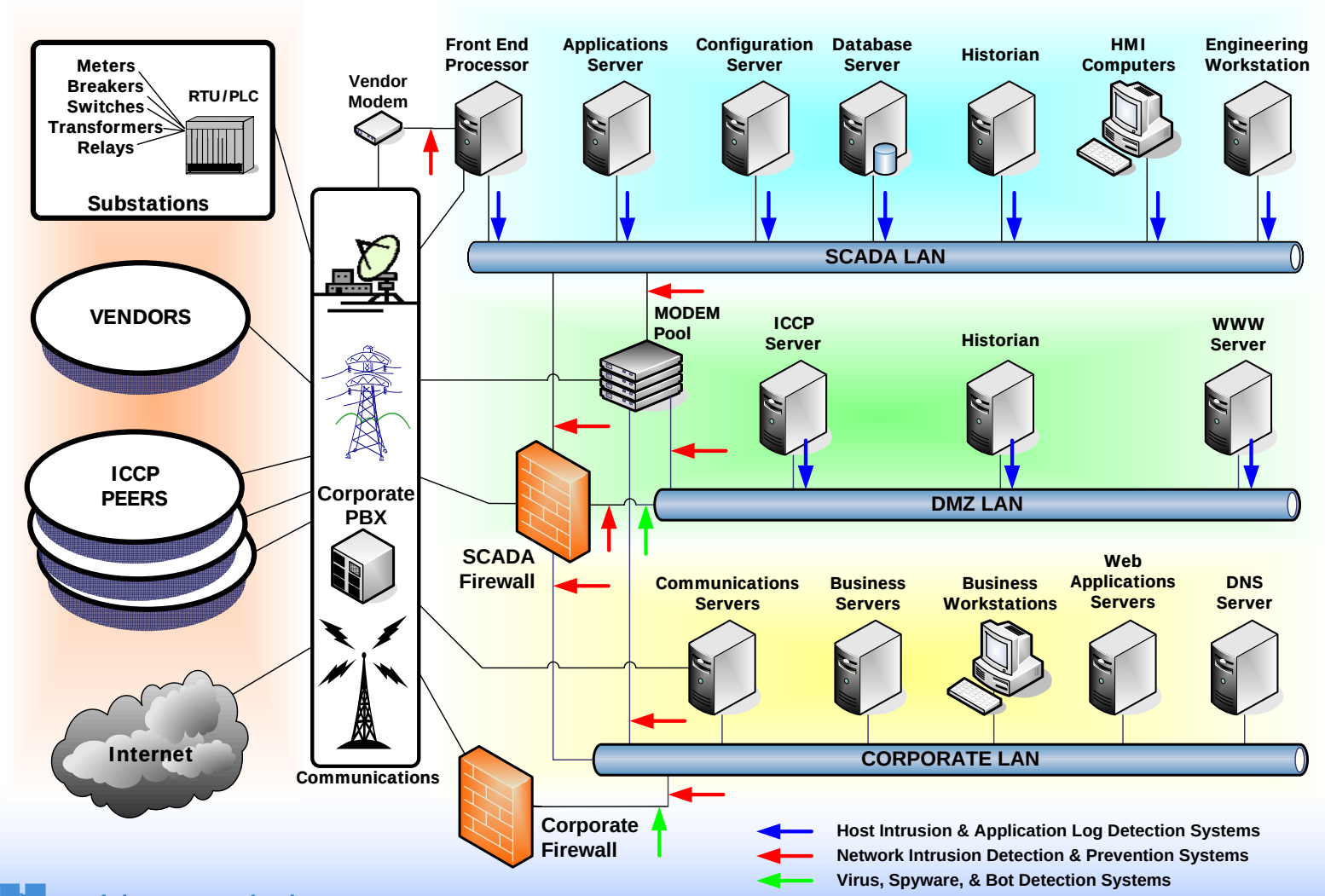
- What they can do for you
  - Tell you what it knows about, specific information
  - Tell you there might be a problem
  - Tell you that you are being picked on
- What they can't do
  - Tell you if the system was exploited
  - Tell you what happened on the system console
  - Do analysis

IDS vs. IPS Placement





IDS Placement Overview

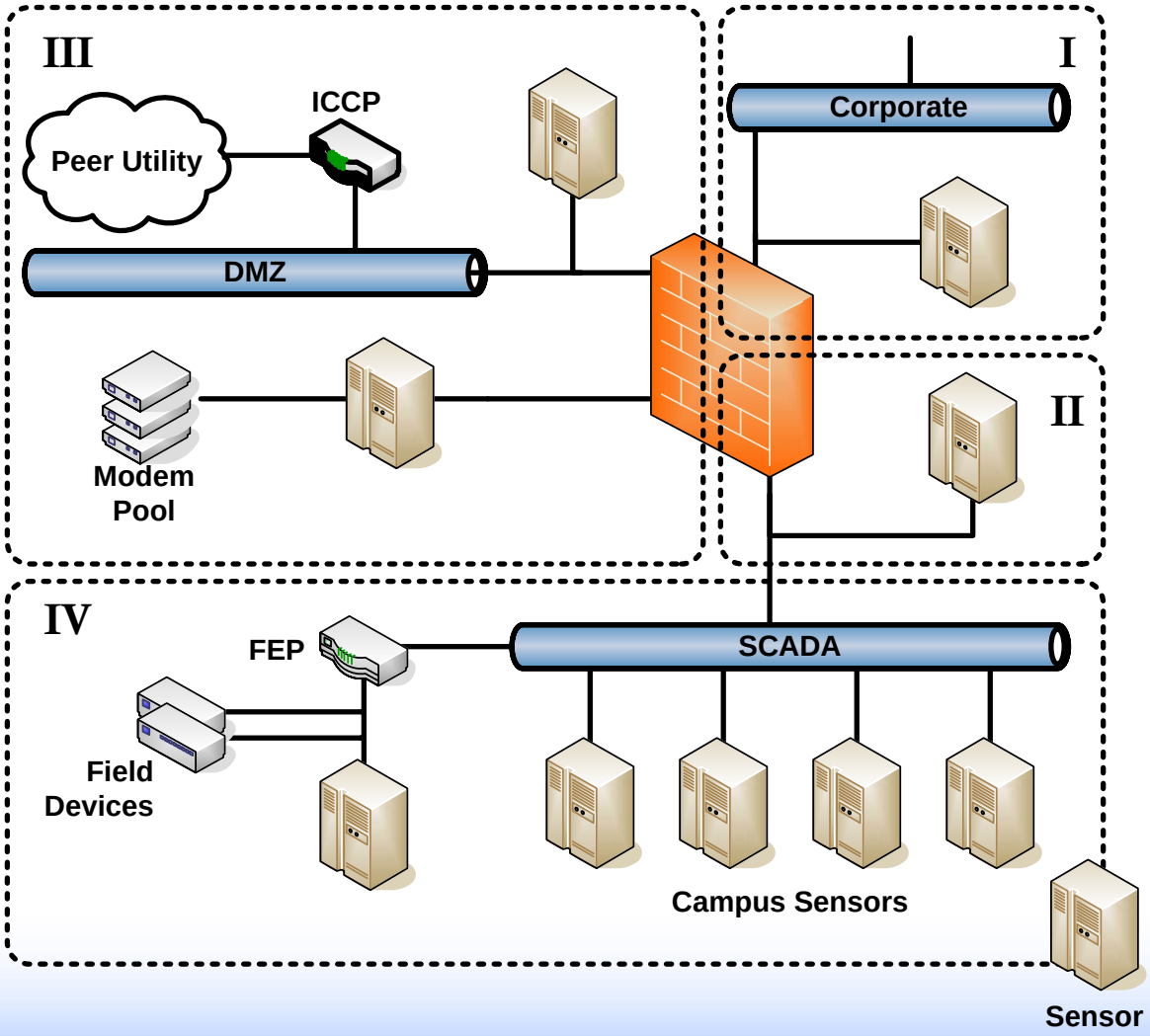


Idaho National Laboratory

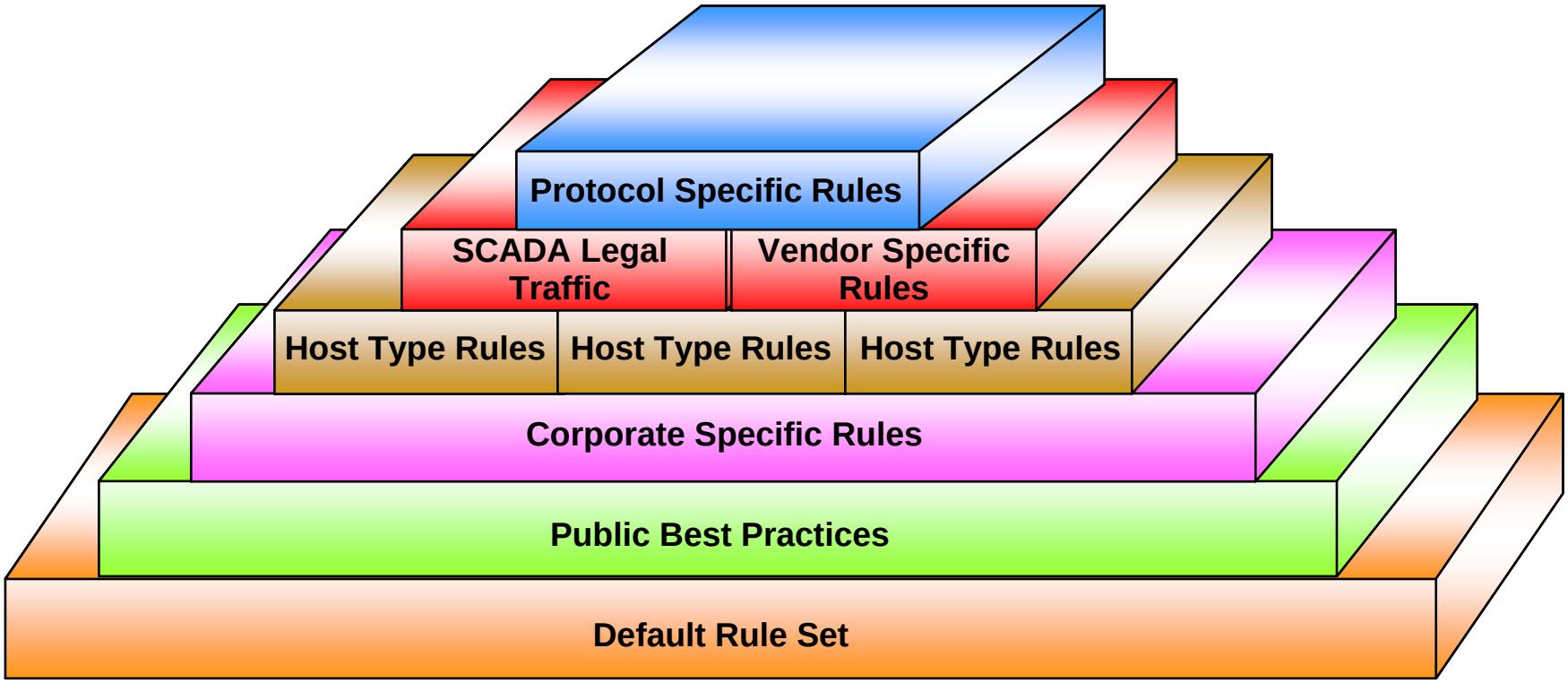
IDS Rules

- Actions taken:
  - None (PASS),
  - Notification (LOG),
  - Alerting (ALERT)
- Writing rules
- Data collection
- Monitoring IDS's

Rule Strategy



Rule Strategy



SCADA Rule Set Should Build Upon Existing Rules

Snort Preprocessors

- **Modular plugins that extend Snort functionality**
- **Process packets after decoding & before the detection engine**
  - **Allows preprocessor to add additional/modify information useful to the detection engine**
- **Preprocessors can also generate alerts**

Snort Intrusion Detection

- Snort has a main configuration file named snort.conf
- This file should be extensively modified to meet the needs of the enclave in which your Snort sensor is deployed.
- Key configuration options to look for:
  - Pre-processing Options
  - Snort Rules
  - Output processing options

Snort Rule Details

- Two main components in Snort rules
- Rule Header
  - Action, Protocol, IP Address, Port Number, direction, IP Address, Port Number
  - alert tcp 10.1.1.1 any -> 10.1.1.3 80
- Rule Options
  - content, messages, references, sid, uricontent, flow, etc
  - alert tcp 10.1.1.1 any -> 10.1.1.3 80 (msg:”cmd.exe web request”; uricontent:”cmd.exe”;)

Snort Output Plugins

- **Flexible system for logging & alerting Snort events**
- **Attached to the alert or log output chain**
- **Plugins on the same chain are run in series**

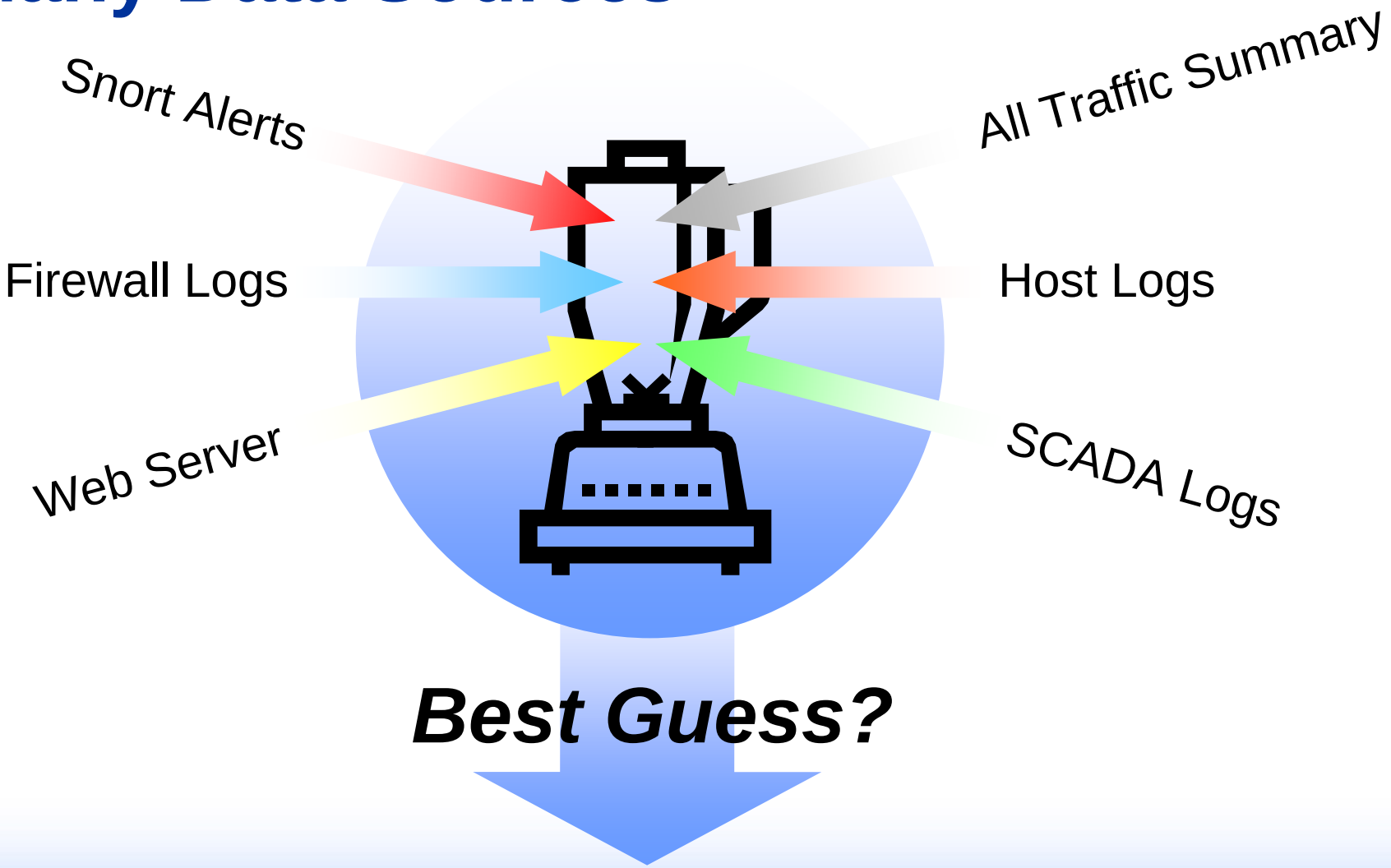
SCADA Application Logs

Data Mining

Process Control Engineer

Security Analyst

Many Data Sources



Log Correlation

- Applications logs
  - Access: who, when, where
  - Events: what, when, where
  - Status: states, when
  - Traffic: source, destination, when, volume
- Traditionally used by control engineers
  - Timing
  - Traffic flow
  - Logic
  - Debug

Log Correlation

- **Data of Security interest**
  - **Access: Authorized?**
  - **Event correlation: I saw events A & B**
  - **Status: Compromised?**
  - **Traffic: Legitimate?**
  - **Timing: DOS?**
  - **Traffic flow: Anomalies?**

Third Step – Customize Network

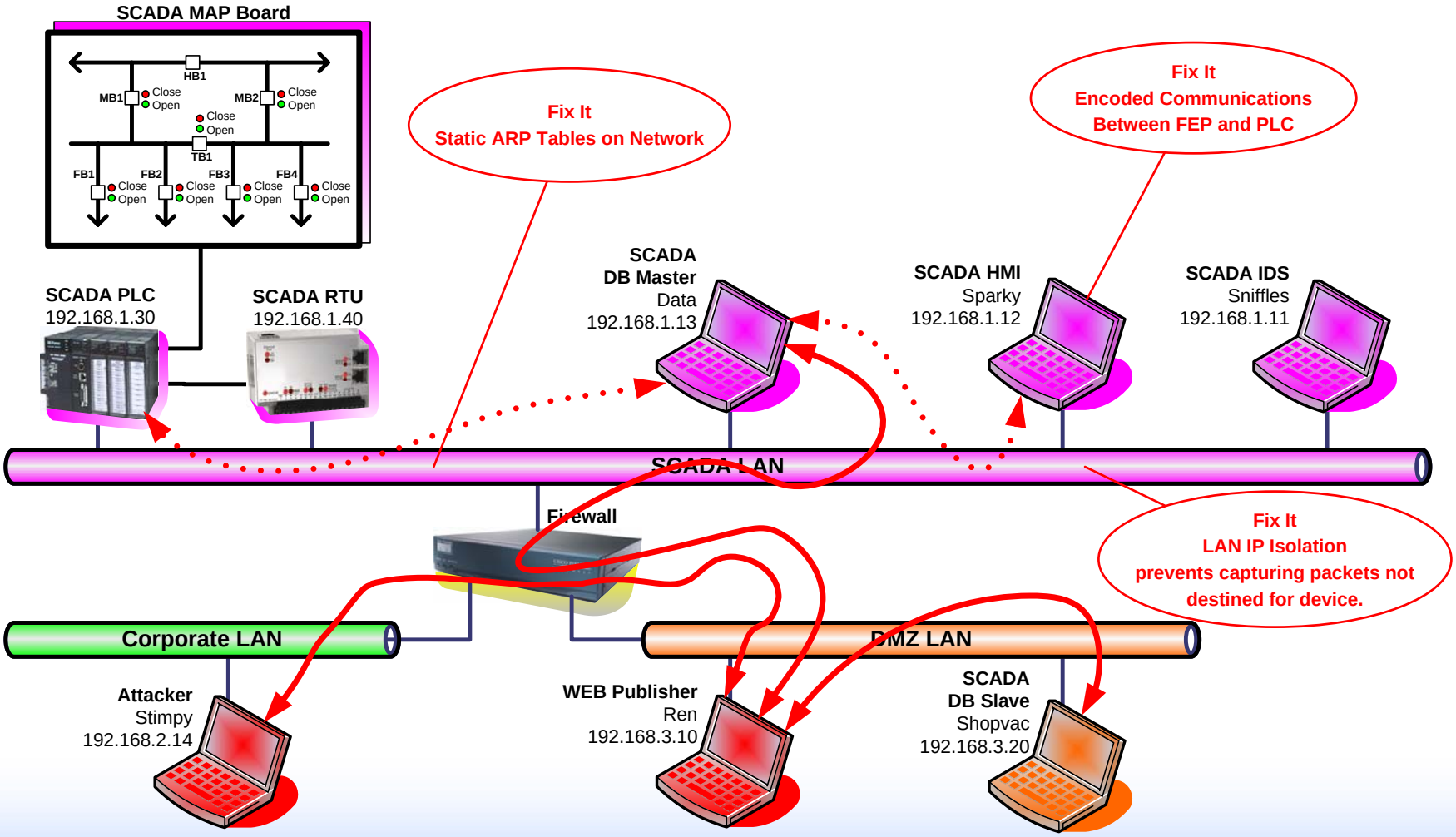
A Good Question that Needs to Be Addressed is:

"Should the Implicit Outbound Rule on the Firewall Be Allowed on the SCADA Network?"

- **Should Hosts Be Able to Access Networks Other than Their Own?**
- **Do the SCADA Hosts Need Default Gateways?**
- **If So, Are These Networks Outside the Firewall? Why?**
- **Outbound Exceptions Should Be Created Just Like Inbound Exceptions**

Unauthorized Control

Fix It



Idaho National Laboratory

Unauthorized Control

Relate It

Excerpts from NERC's Top 10 Vulnerability List

- **Control systems data sent in clear text.**
- **Lack of quick & easy tools to detect & report on anomalous or inappropriate activity. Non existent forensic & audit methods.**
- **Poorly designed control system networks that fail to employ sufficient defense-in-depth mechanisms.**

Unauthorized Control

Test It

- Let's test our newly implemented mitigation techniques to see if our detection is more reliable & effective
- How do the IDS alerts compare to the original demonstration?
- Is the IDS able to detect the MITM?

Summary

- What was wrong?
- How did we find it?
- How did we fix it?
- What did we learn from it?
- When does it end?
- Q&A session

Interactive Test Discussion

Thank You

Supplemental Slides

Protocols

Process/ Application	Hypertext Transfer	File Transfer	Electronic Mail	Terminal Emulation	Domain Names	File Transfer	Client/ Server	Network Management	Application
	Hypertext Transfer Protocol (HTTP) RFC 2068	File Transfer Protocol MIL-STD-1781 RFC 959	Simple Mail Transfer Protocol (SMTP) MIL-STD-1781 RFC 2821	TELNET Protocol MIL-STD-1782 RFC 854	Domain Name System (DNS) RFC 1034, 1035	Trivial File Transfer Protocol (TFTP) RFC 783	Sun Microsystems Network File System Protocol (NFS) RFCs 1014,1057, & 1094	Simple Network Management Protocol (SNMP) V1: RFC 1157 V2: RFC 1901-10 V3: RFC 2571-75	
Host-to-Host	Transmission Control Protocol (TCP) MI-STD-1788 RFC 793						User Datagram Protocol (UDP) RFC 768		Presentation Session
Internet	Address Resolution ARP RFC 826 RARP RFC 903		Internet Protocol (IP) MIL-STD-1777 RFC 791				Internet Control Message Protocol (ICMP) RFC 792		Transport Network
Network Interface	Network Interface Cards: Ethernet, Token Ring, ARCNET, MAN & WAN RFC 894, RFC 1042, RFC 1201 & others								Data Link
	Transmission Media: Twisted Pair, Coax, Fiber Optics, Wireless Media, etc.								Physical

Typical Attack Steps

Typical Attack Steps

- **Target Identification / Selection**
- **Reconnaissance**
- **System Exploits**
- **Keeping Access**
- **Covering the Tracks**

Target Identification / Selection

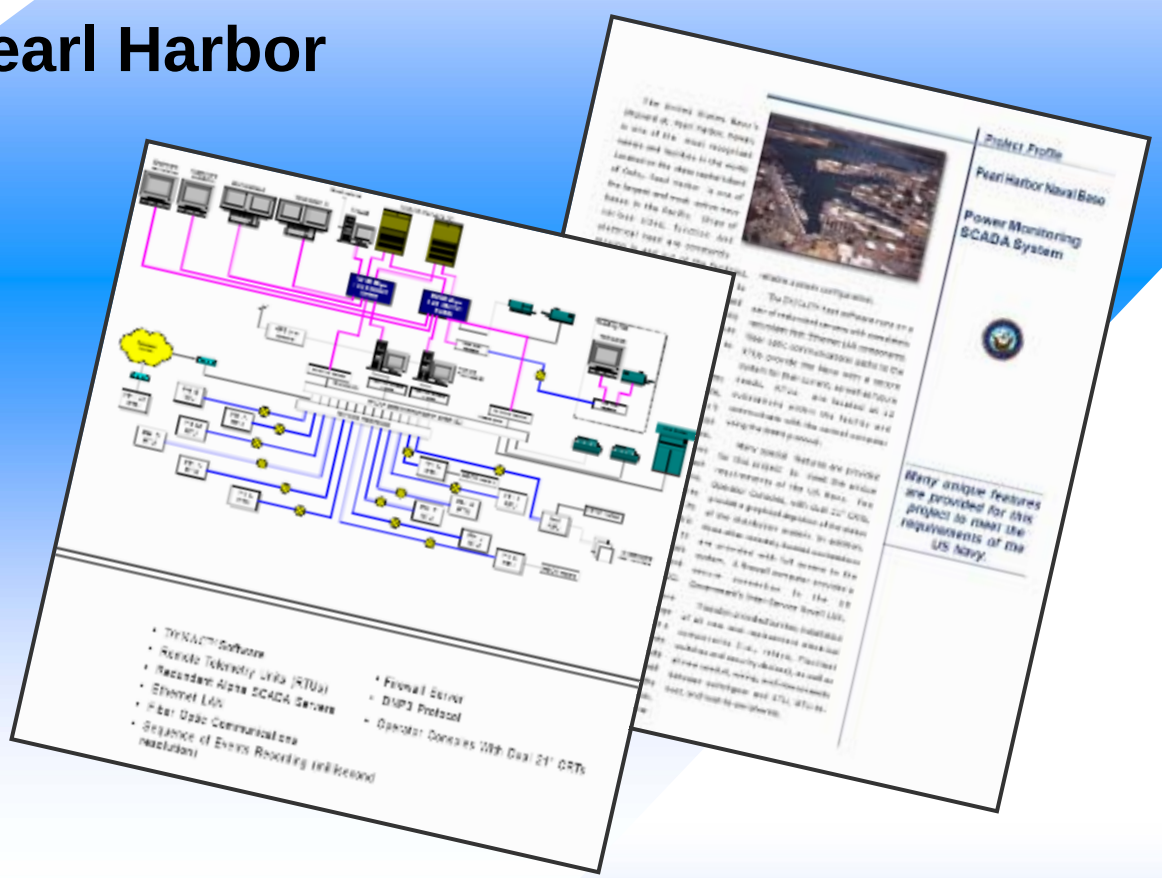
- **How visible is your company in the public?**
- **Is your company/utility desirable?**
- **How do your defenses compare to your neighbor?**

Reconnaissance

- Open Source Intelligence
  - External Web Site
  - Google (Internet) Searches
  - DNS Lookups
- Dumpster Diving
- Social Engineering
- War Dialing / War Driving
- Scanning
- Insider Threat

Reconnaissance Example

- **Picking on the U.S. Government**
  - **SCADA at Pearl Harbor**



System Exploits

- **Viruses & Worms**
- **Email**
- **Hostile Web Pages**
- **Direct Attacks**

Keeping Access

- **Attacker may/may not care**
- **Account creation**
- **Password cracking**
- **Backdoors / Trojan Horses**
- **Rootkits**

Covering the Tracks

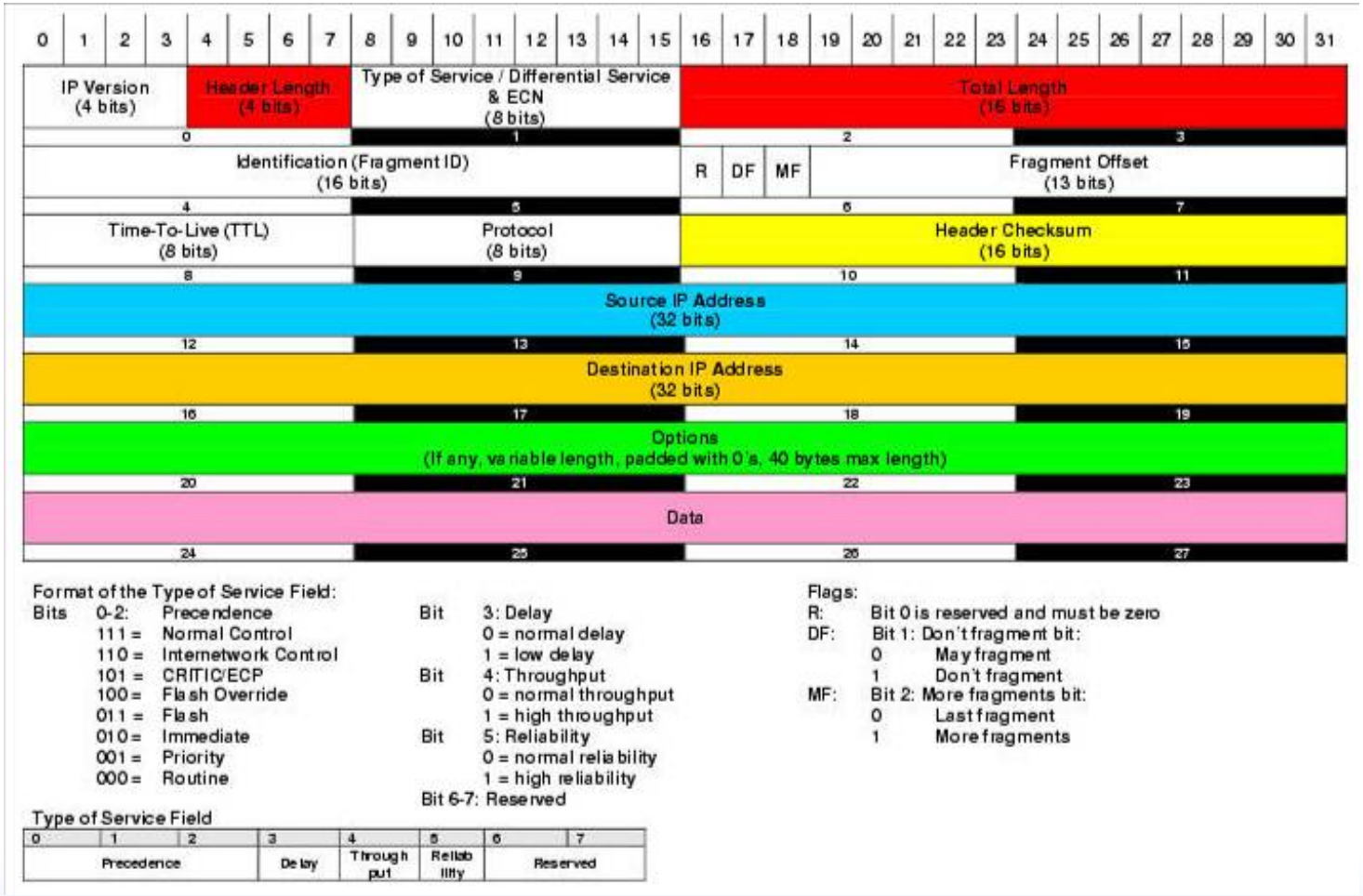
- **Physical damage**
- **Hiding files**
- **Log file modification / deletion**
- **Covert channels (loki, ncovert)**

TCP/IP Packet Headers

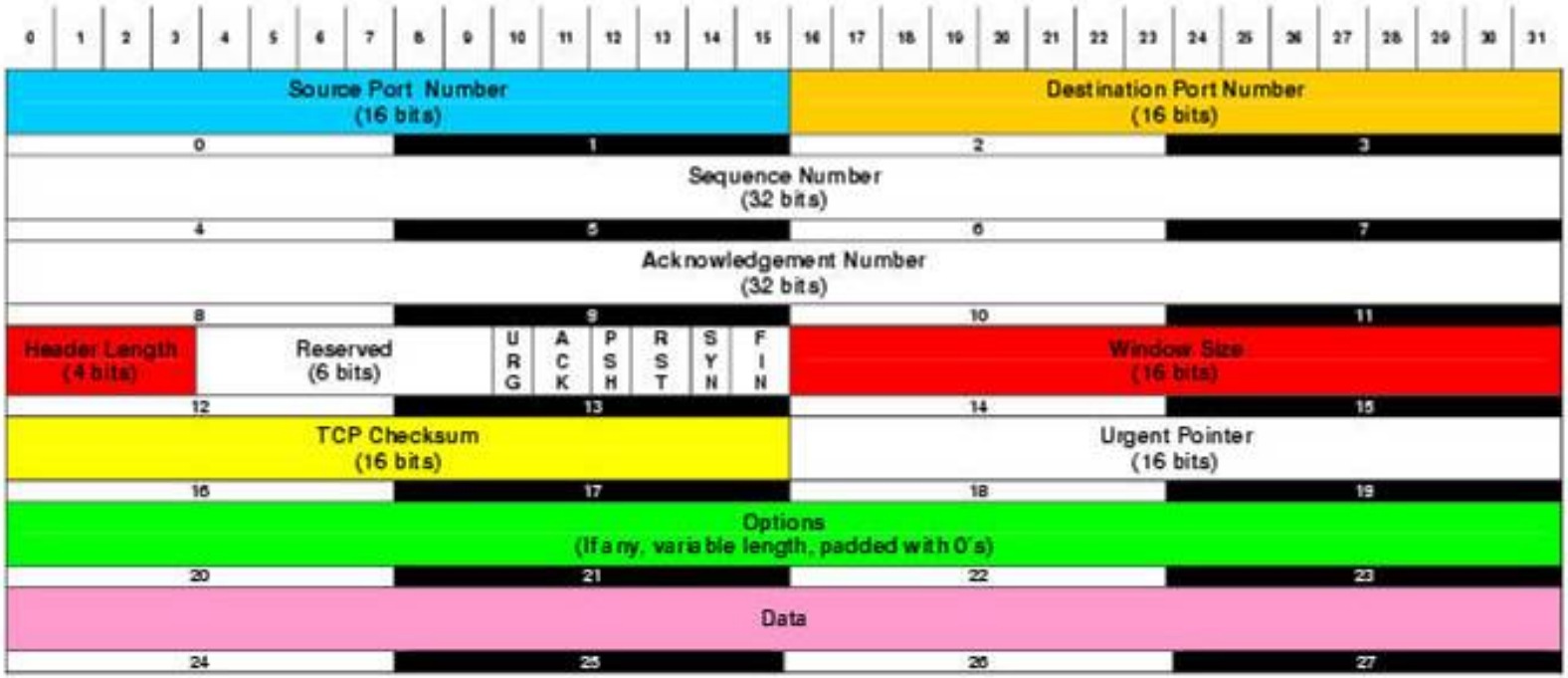
The Following Packet Header Slides Are Available on the Web From:

www.Securitywizardry.Com/Protpackets.Htm

IP Header


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
IP Version (4 bits)
Header Length (4 bits)
Type of Service / Differential Service & ECN (8 bits)
Total Length (16 bits)
0 1 2 3
Identification (Fragment ID) (16 bits)
R
DF
MF
Fragment Offset (13 bits)
4 5 6 7
Time-To-Live (TTL) (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
8 9 10 11
Source IP Address (32 bits)
12 13 14 15
Destination IP Address (32 bits)
16 17 18 19
Options (If any, variable length, padded with 0's, 40 bytes max length)
20 21 22 23
Data
24 25 26 27
Format of the Type of Service Field:
Bits 0-2: Precendence
111 = Normal Control
110 = Internetwork Control
101 = CRITIC/ECP
100 = Flash Override
011 = Flash
010 = Immediate
001 = Priority
000 = Routine
Bit 3: Delay
0 = normal delay
1 = low delay
Bit 4: Throughput
0 = normal throughput
1 = high throughput
Bit 5: Reliability
0 = normal reliability
1 = high reliability
Bit 6-7: Reserved
Flags:
R: Bit 0 is reserved and must be zero
DF: Bit 1: Don't fragment bit:
0 May fragment
1 Don't fragment
MF: Bit 2: More fragments bit:
0 Last fragment
1 More fragments
Type of Service Field
0 1 2 3 4 5 6 7
Precedence
Delay
Through put
Reliab ility
Reserved

TCP Header



Example TCP Applications

- TELNET
- FTP
- SMTP
- HTTP

UDP Header

Common UDP Well-Known Server Ports

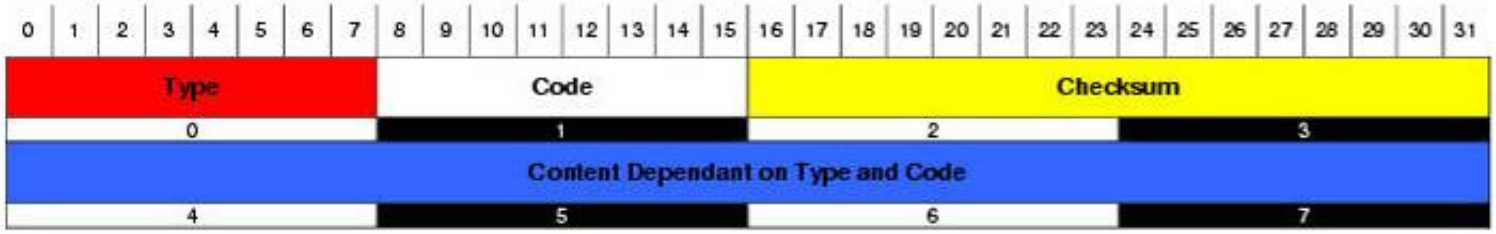
Port	Description
7	Echo
19	Chargen
37	Time
53	Domain
67	Bootps (DHCP)
68	Bootpc (DHCP)
69	Tftp
137	Netbios-ns

Port	Description
138	Netbios-dgm
161	Snmp
162	Snmp-trap
500	Isakmp
514	Syslog
520	Rip
33434	Traceroute

Length
The number of bytes in the entire datagram, including the header; minimum value = 8

Checksum
Covers pseudo-header and entire UDP datagram

ICMP Message Format

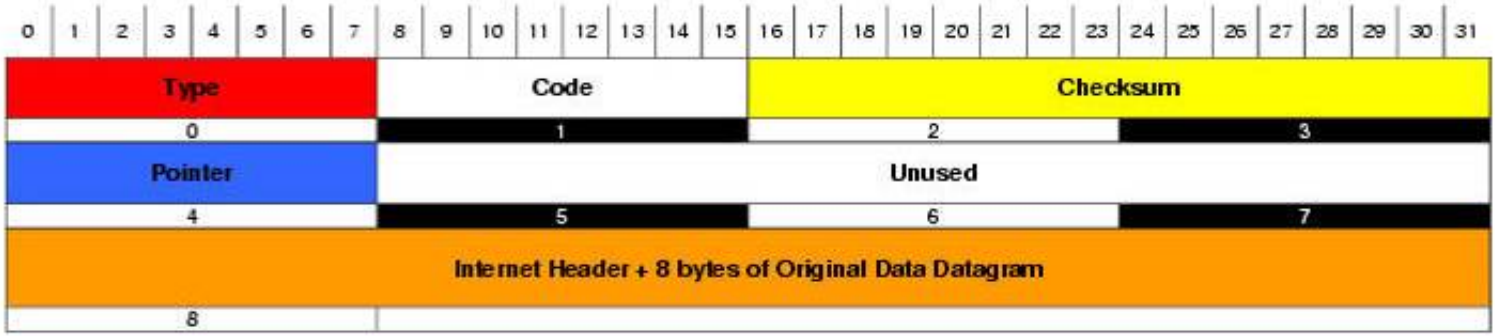


Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated
	9	Network admin prohibited
	10	Host admin prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
4	0	Source Quench (Slow down/Shut up)

Type	Code	Meaning
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & Network
	3	Redirect datagram for the TOS & Host
8	0	Echo
9	0	Router advertisement
10	0	Router selection
11	0	Time To Live exceeded in transit
	1	Fragment reassemble time exceeded
12	0	Pointer indicates the error (Parameter Problem)
	1	Missing a required option (Parameter Problem)
	2	Bad length (Parameter Problem)
13	0	Time Stamp
14	0	Time Stamp Reply
15	0	Information Request
16	0	Information Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (Tracert)

Idaho National Laboratory

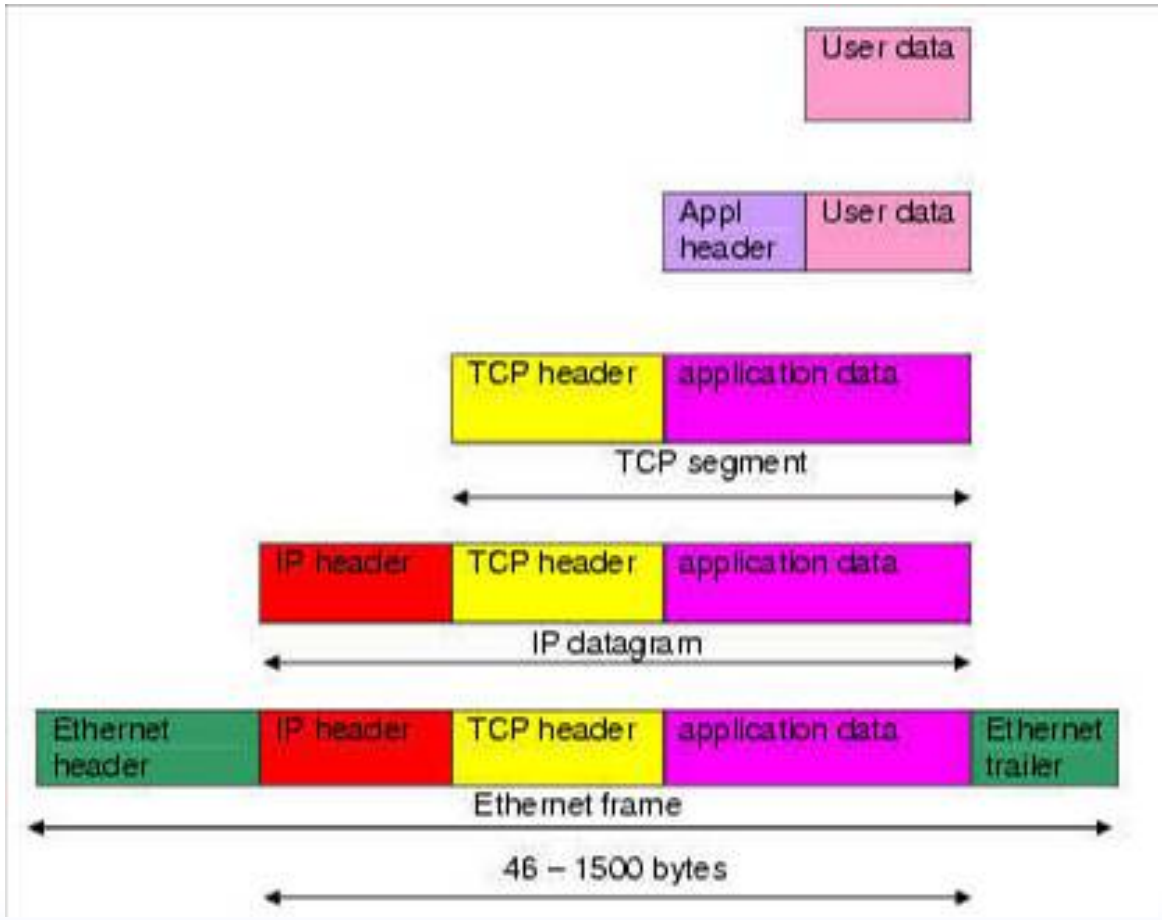
ICMP Parameter Message Format



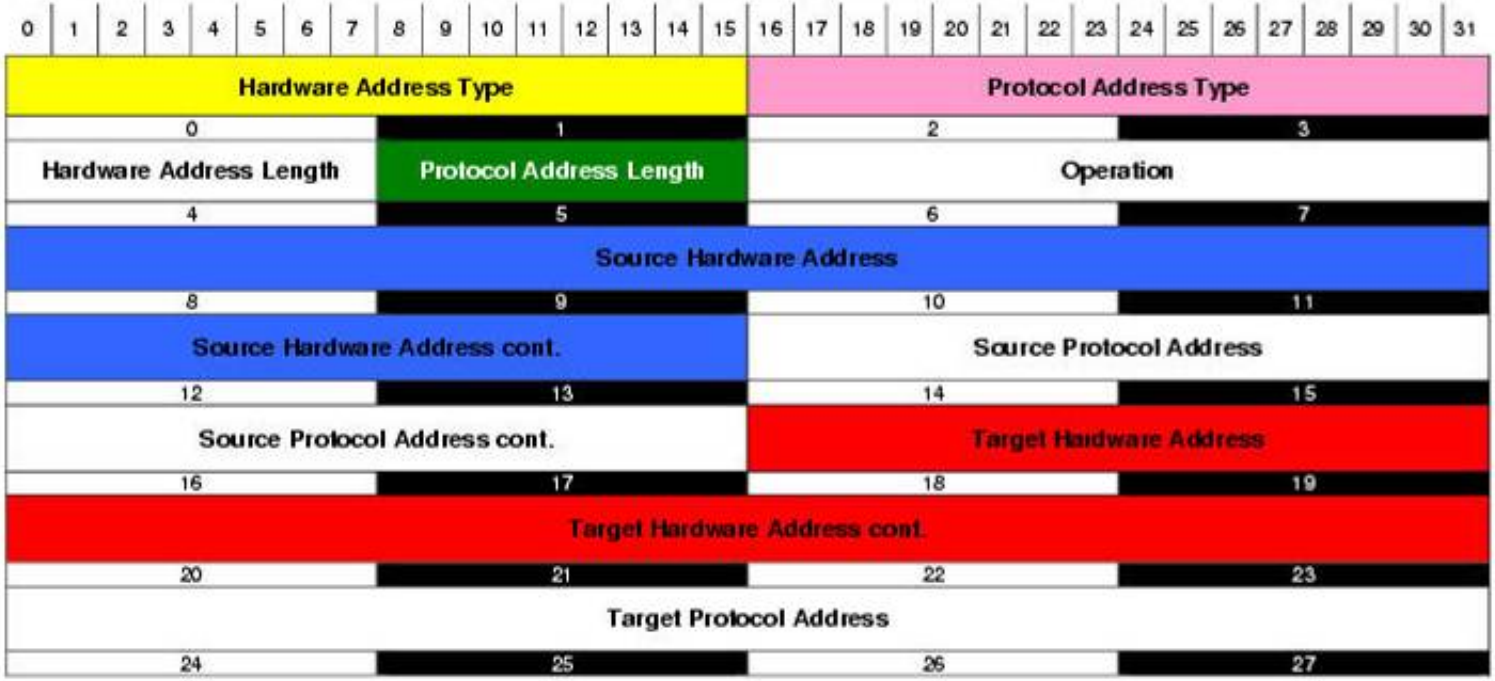
Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated
	9	Network admin prohibited
	10	Host admin prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
4	0	Source Quench (Slow down/Shut up)

Type	Code	Meaning
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & Network
	3	Redirect datagram for the TOS & Host
8	0	Echo
9	0	Router advertisement
10	0	Router selection
11	0	Time To Live exceeded in transit
	1	Fragment reassemble time exceeded
12	0	Pointer indicates the error (Parameter Problem)
	1	Missing a required option (Parameter Problem)
	2	Bad length (Parameter Problem)
13	0	Time Stamp
14	0	Time Stamp Reply
15	0	Information Request
16	0	Informaiton Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (Tracert)

Encapsulation of Data



ARP



ARP Parameters for Ethernet and IPv4

Hardware Address Type		
	1	Ethernet
	6	IEEE 802 LAN
Protocol Address Type		
	2048	IPv4 (0x0800)
Hardware Address Length		
	6	for Ethernet / IEEE 802

Protocol Address Length		
	4	For IPv4
Operation		
	1	Request
	2	Reply

DNS

DNS Parameters

Query / Response

0	Query
1	Response

Opcode

0	Standard Query (QUERY)
1	Inverse Query (IQUERY)
2	Server Status Request (STATUS)

AA	1 = Authoritative Answer
TC	1 = TrunCation
RD	1 = Recursion Desired
RA	1 = Recursion Available
Z	Reserved; set to 0 (The DNS Evil bit)

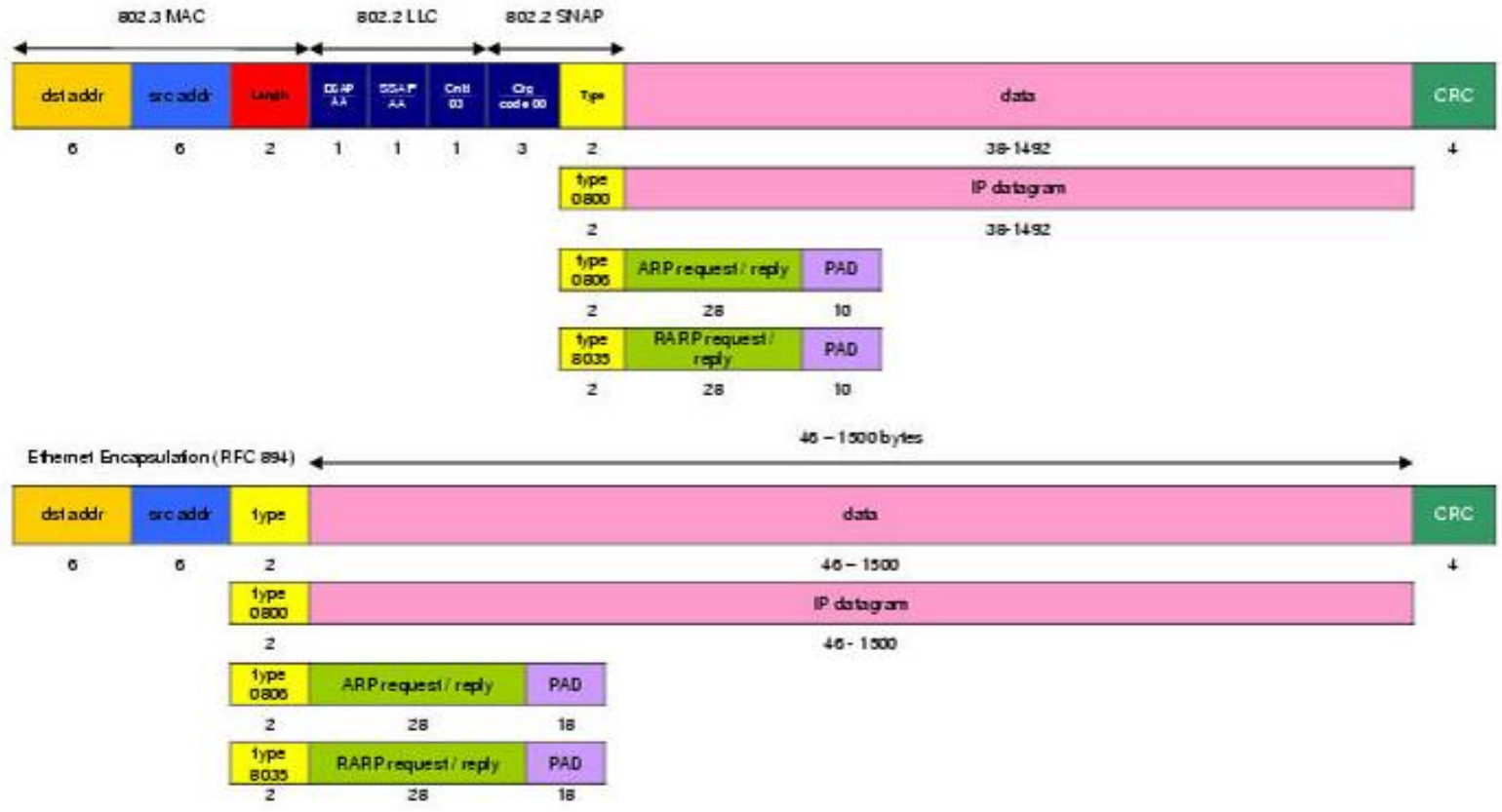
Response Code

0	No error
1	Format error
2	Server failure
3	Non-existent domain (NXDOMAIN)
4	Query type not implemented
5	Query refused

QDCOUNT	No. of entries in the Question Section
ANCOUNT	No. of resource records in Answer Section
NSCOUNT	No. of name server resource records in Authority Section
ARCOUNT	No. of resource records in Additional Information Section.

Link Layer Headers

IEEE 802.2 / 802.3 Encapsulation (RFC 1042)



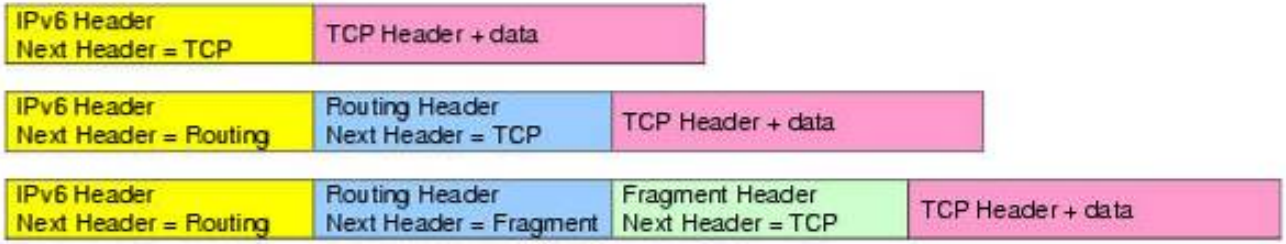
IPv6 Header

Note: the standard IPv6 is 40 bytes long as opposed to 20 bytes for a v4 header (without options) also the v6 header is a lot simpler than is older brother and thus easier to understand. There are however, IPv6 extension headers that fit between the IPv6 header and next header, these include routing headers and fragment headers.

Idaho National Laboratory

IPv6 Extension Headers

(RFC2460 IPv6 Spec) An IPv6 Packet Can Carry As Many Extension Headers As It Needs (Obviously within Reason)



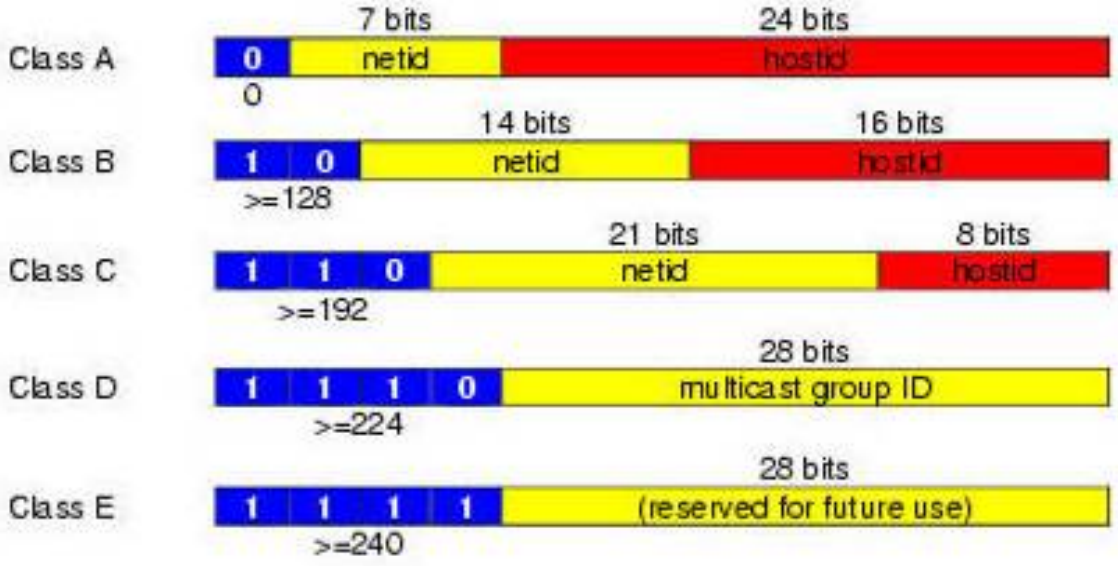
Extension Header Order

The extension headers are not checked until it the destination identified. The following order is recommended in RFC 2460 and that they feature no more than once except for the destination options which can occur at most twice (once before a Routing header and once before the upper-layer header):

IPv6 header, Hop-by-Hop Options header, Destination Options header (note 1), Routing header, Fragment header
then
Authentication header (note 2), Encapsulating Security Payload header (note 2), Destination Options header (note 3) and then upper-layer header

note 1: for options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header.	**note 2**: additional recommendations regarding the relative order of the Authentication and Encapsulating Security Payload headers are given in [RFC-2406].	**note 3**: for options to be processed only by the final destination of the packet.

Network Classes



Class	Range
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 255.255.255.255


SNORT™

Snort Rules Overview

- Snort Rules Expectations
- Snort Configuration files
- Rule Details
- Common Rule Options
- Custom Rules
- Rule Order
- Rule Tidbits

Snort Rule Expectations

- What they can do for you
  - Tell you what it knows about, specific information
  - Tell you there might be a problem
  - Tell you that you are being picked on
- What they can't do
  - Tell you if the system was exploited
  - Tell you what happened on the system console
  - Do analysis

Config Files:
Snort Configuration File

- Defines common variables
- Defines System settings
  - Defines Runtime configuration options
  - Specifies Rules file to be included

Config File: Common Variables

- var: <variable name> <value>
  - HOME_NET, EXTERNAL_NET, SMTP_SERVER, etc
- IP Lists
  - var HOME_NET [10.1.2.0/24,192.168.0.0/24]
  - var DNS_SERVER 10.1.1.1
- PORTS
  - Single port number
    - var HTTP_PORT 80
  - Range of port number
    - var COMMON_SERVICES 20:1024
    - var HIGH_PORTS 3000:
    - var LOW_PORTS :1025
  - Snort currently does not support port lists

Config File: System Settings

- **Runtime configuration options**
  - **config order: alert pass log custom**
  - **config bpf_file: filename**
- **Specify rule files to be included**
  - **include $RULE_PATH/local.rules**

Rule Details

- Two main components in Snort rules
- Rule Header
  - Action, Protocol, IP Address, Port Number, direction, IP Address, Port Number
  - alert tcp 10.1.1.1 any -> 10.1.1.3 80
- Rule Options
  - content, messages, references, sid, uricontent, flow, etc
  - alert tcp 10.1.1.1 any -> 10.1.1.3 80 (msg:”cmd.exe web request”; uricontent:”cmd.exe”;)

Rule Headers

alert udp $EXTERNAL_NET any -> $HOME_NET 135

- **Rule Actions**
  - **alert, log, pass, activate, dynamic, custom rule type**
  - **Active & dynamic are being deprecated in favor of tagging**
- **Protocols**
  - **IP,TCP, UDP, ICMP**
- **IP Address**
  - **$HOME_NET, $EXTERNAL_NET, 10.10.10.1/32, any**
- **Port Numbers**
  - **$HTTP_PORT, any, 6001:6666, 8080**
- **Direction Operator -> <- <>**
- **Absolute Minimum for a rule**

Common Rule Options

- **Four Categories of Rule Options:**
  - **Meta-Data**
    - **Information about the rule that has no effect on detection**
  - **Payload**
    - **Look for specified data in the packet payload**
  - **Non-Payload**
    - **Look for specified data in headers of packet**
  - **Post Detection**
    - **How subsequent packets are handled after the rule has been triggered**

Meta Data Options

- Message to Print Out When Rule Triggers
  - msg:”cmd.exe attempt”;
- Reference Information
  - ref: <id system>, <id>
  - List of available id systems is located in the reference.conf file
  - May add your own reference information
  - References are important!!!!
- Revision Information
  - rev:<number>;

Meta Data Options (cont.)

- **Snort Identification Number**
  - **sid:<number>**
  - **<100 reserved for future use**
  - **100-1,000,000 rules included in Snort distribution**
  - **>1,000,000 local custom rules**
  - **sid is also used for thresholding & suppression**

Meta Data Options (cont.)

- **Classtype categorizes rules into attack classtypes**
  - **classtype: <classification>;**
  - **Listing of default class types & priorities can be found in the classification.conf file**
- **Priority used to change default priority of classification type for a specific rule**
  - **priority:<number>;**

Payload Options

- **Content**
- **Uricontent**
- **PCRE - Perl Compatible Regular Expressions**

Payload Options – Content

- **Content – searches for specified data within the packet payload**
  - **Can be specified as ascii or hex**
  - **Multiple contents in one rule are searched for in the order they appear in the rule**
  - **content:”cmd.exe”; content:”|00 00 00 FA 00|”;**

Payload Options – Content Modifiers

- Content modifiers must be listed after the content they are to modify.
  - nocase – don't pay attention to case of content
    - content:"cmd.exe"; nocase;
  - rawbytes – ignores any decoding previously done & compares against the raw bytes of the packet
    - content:"|00 00 00 FA 00|"; rawbytes;
  - depth – tells the engine how deep into the packet to stop looking for the content, either from the beginning of the packet or from the previous content match
    - content:"cmd.exe"; depth 50;

Payload Options – Content Modifiers (cont.)

- Content Modifiers
  - within – search for content within x number of bytes from either the beginning of packet or previous content match (flexible)
    - content:”cmd.exe”; within 10;
  - offset – search for the content only after x number of bytes from either beginning of packet or previous content match (static)
    - content:”cmd.exe”; offset 5;
  - distance - only search for next content, x number of bytes relative from previous content match
    - content:”windows”; content:”cmd.exe”; distance 1;

Payload Options – Uricontent

- Works in conjunction with the HTTP Inspect preprocessor
- Uricontent searches normalized uri field
  - uricontent:”/etc/passwd”;
- Uses the same content modifiers as seen previously

Payload Options – PCRE

- Perl Compatible Regular Expressions (PCRE)
  - www.pcre.org
  - Replaces regex
- pcre /regular expression/ modifiers [ismx AEG RUB]
  - Perl compatible modifiers
    - i – ignore case
    - s – include newlines as meta characters
    - m – have ^ & $ evaluated at all newlines within the buffer
    - x – ignore white space except when escaped or inside a character class

Payload Options – PCRE (cont.)

- pcre /regular expression/ modifiers [ismx AEG RUB]
  - PCRE compatible modifiers
    - A – pattern match only at start of buffer same as perl ^
    - E – Set $ to match only at end of subject string
    - G – Sets greediness of qualifiers
  - Snort specific modifiers
    - R – search relative to last pattern match
    - U – match decode uri content
    - B – don't use decoded buffers (similar to rawbytes)
- alert tcp any any -> any any (pcre:"/BLAH/i";)
  - Looks for BLAH while ignoring the case of blah

Non-Payload Options

- General Information
- IP Specific
- TCP Specific
- ICMP Specific
- Other

Non-Payload Options – General

- **Snort is able to not only search on packet payload info but also on packet header information**
- **Fields of the IP, ICMP & TCP headers can be included in the rules**

Non-Payload Options – IP Specific

- **fragoffset**
  - **fragoffset [<|>] value**
  - **Use to compare the fragoffset value in the IP header to a decimal value. Commonly used with the more fragments options.**
- **fragbits**
  - **fragbits: [+-!] M D R**
  - **Check for the more fragment bit, don't fragment bit, or the reserved bits are set**

Non-Payload Options – IP Specific

- ip_proto
  - ip_proto: [!] value or name
  - This enable the rule to look for other IP protocols like IGMP
- TTL
  - ttl: value
  - Compares the ttl value of the IP header to a decimal value. Mostly used for traceroutes.

Non-Payload Options – TCP

- **Flags**
  - **flags: [!|*|+] <FSRPAU120>, <FSRPAU120>;**
  - **Checks the tcp flag bits**
- **Sequence**
  - **seq: number**
  - **Checks the sequence number**
- **Acknowledge**
  - **ack: number**
  - **Checks the acknowledgement number**

Non-Payload Options – TCP (cont.)

- **Flow**
  - **Not associated with the flow preprocessor**
  - **Dependent on the stream4 preprocessor**
  - **Allows rules to selectively apply only to one side of the session**
  - **flow: [to_server | to_client | from_server | from_client | stateless | established | no_stream | only_stream]**

Non-Payload Options – TCP (cont.)

- **Flow Modifiers**
  - **to_client, to_server, from_server, from_client**
    - **Which side of the session to apply the rule to**
  - **established**
    - **Only evaluate rule once the stream has been fully established**
  - **stateless**
    - **Don't worry about state of stream**
  - **only_stream**
    - **Only trigger on rebuilt streams**
  - **no_stream**
    - **Don't trigger on streams (useful for dsize & rawbytes)**

Non-Payload Options – ICMP

- itype
  - itype: value
  - Compares to the ICMP type
  - Useful for detecting icmp based scans
- icode
  - icode: value
  - Compares to the ICMP code
  - Useful for detecting icmp based scans
- icmp_id
  - icmp_id: value
  - Compares to the ICMP id value
  - Useful for detecting covert channels
- icmp_seq
  - icmp_seq: value
  - Compares to the ICMP sequence number value
  - Useful for detecting covert channels

Non-Payload Data – Other

- sameip
  - Triggers on packets whose source IP & destination IP are the same
- rpc
  - Looks for rpc application, version, & procedure number in SUNRPC Call requests
  - Due to fast pattern matching actually slower than content matching.
  - rcp: <application number>, <[version|*]>,< [procedure number|*]>;