Securing Linux Servers - A Survey

Nigel Edwards
Hewlett-Packard, Internet Security Solutions Lab

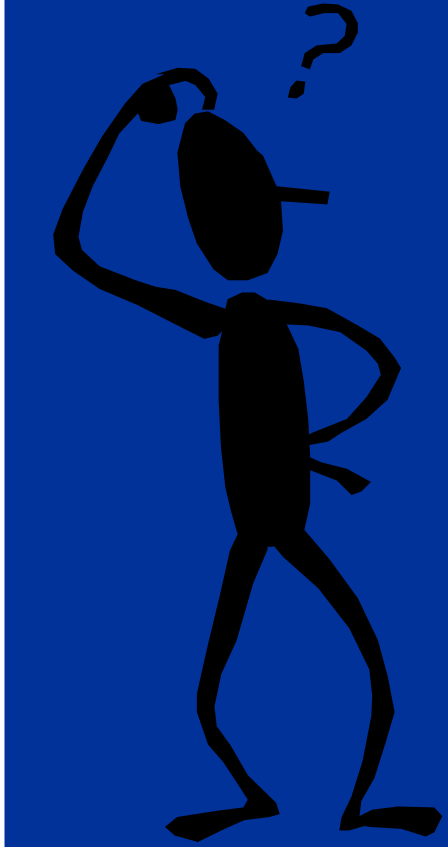
Agenda

- Patching
- Tools and utilities
- Kernel strengthening
  - Some background & concepts
  - Some examples

Out of scope for this talk

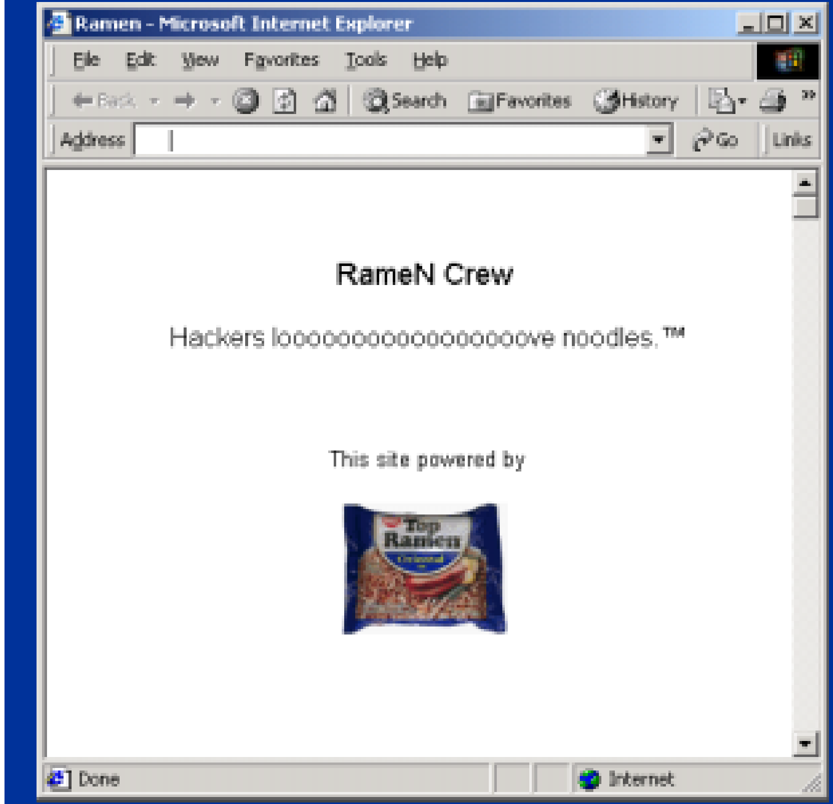
- Defences against attacks involving physical access to the machine
    - Encrypted file systems, biometrics, smartcards…
- Backup and crash recovery
    - Amanda, BRU/CRU,…
- Programmer tools
    - pscan, perlnecklace, …
- Password strengthening and authentication technology
    - Smartcards, PKI, password crackers,…
- Network access control and firewalls
    - TCP Wrappers, ipchains, iptables

What’s the nature of the problem?



- Bugs are the major source of vulnerabilities
  - Application bugs account for about 80%
- CERT issued 37 security advisories in 2001
  - http://www.cert.org/advisories/
  - 30 concerned bugs in applications
  - 1 or 2 did not concern bugs

The anatomy of the Ramen worm (January 2001)


Ramen - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address Go Links
Done Internet

RameN Crew

Hackers looooooooooooooooooove noodles.™

This site powered by

- Exploited known bugs in services:
  - rpc.statd, WU-FTP, LPRng
  - Buffer overflow allows the attacker to gain control of the process executing the service
  - Code is downloaded to overwrite some system executables
  - Root access is gained
  - “index.html” files overwritten
  - The network is probed looking for other vulnerable hosts

Rootkits

- Hide the attackers presence
- Allow repeated use of the system by attackers
- Two variant
  - Updated system binaries
  - Loadable kernel module

Patching

- Security Alerts
  - Vendor security bulletins
  - Bugtraq, CERT, etc, …
  - Managed services e.g. Security Focus
- Automated services
  - Aduva - http://www.aduva.com/
  - Red Hat - https://rhn.redhat.com/
- Problems with patching
  - Not all problems are known – you may be victim before the patch is available or before you can apply the patch
  - The attackers are reading the same information sources
    - – who is going to win…..?

Tools and Utilities

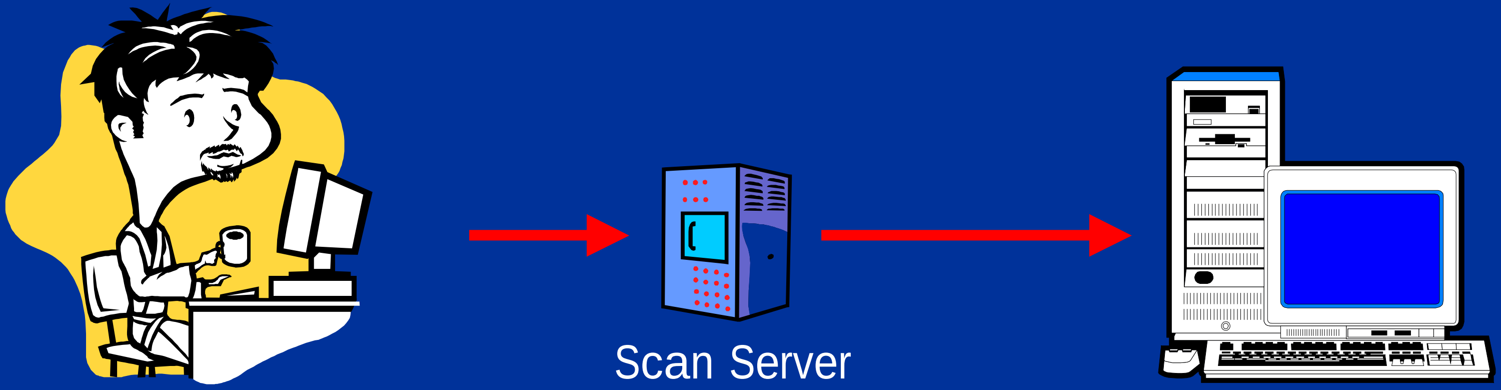
- Bastille
- System scanners (e.g. Nessus, Tiger)
- Intrusion detection (Snort)
- Audit (Snare)
- Psionic PortSentry
- Psionic HostSentry
- Psionic LogCheck
- Tripwire

Bastille

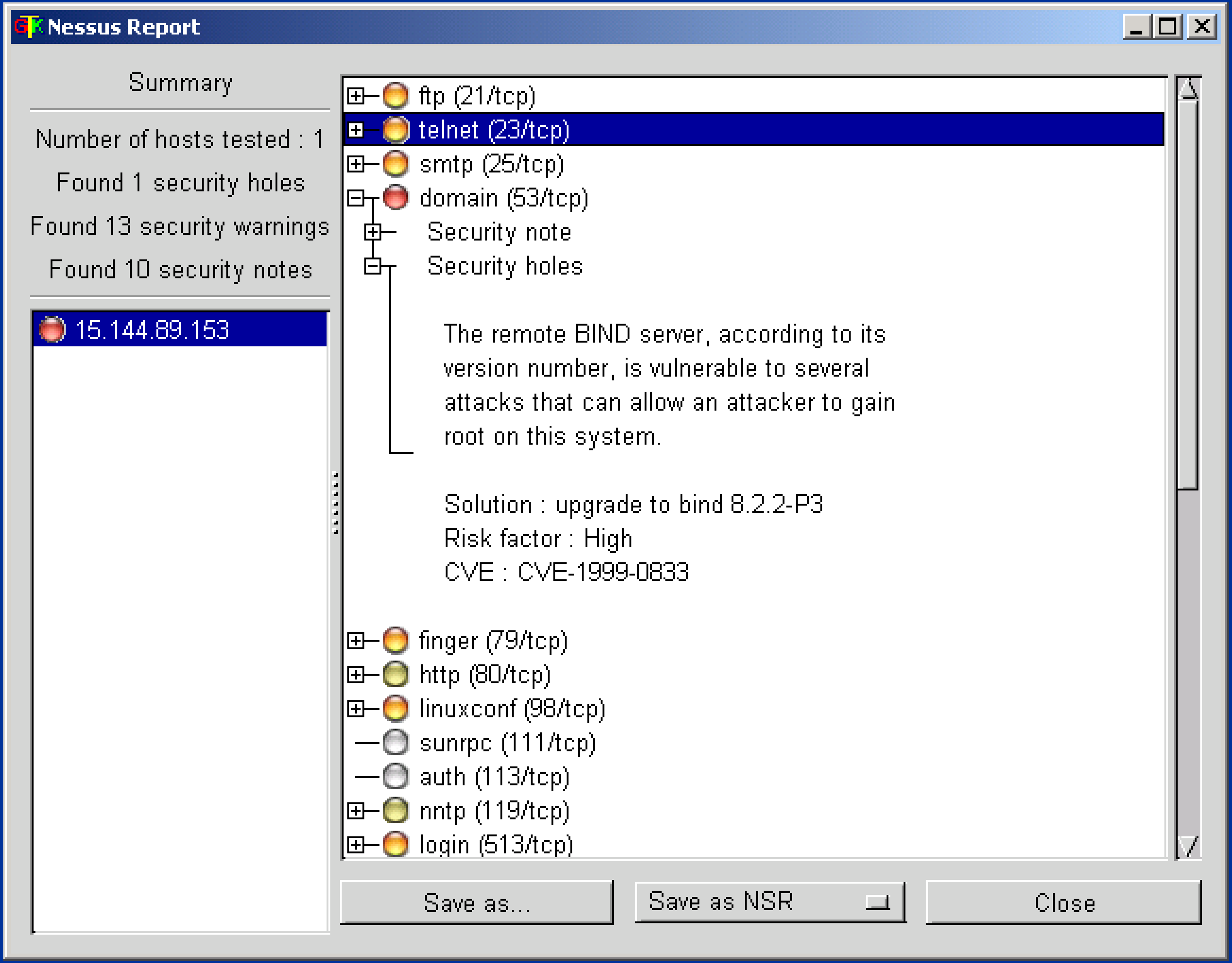
- Secure configuration scripts for Linux
  - Mandrake and Red Hat
- Tightens permissions (e.g remove SUID)
- Account security (e.g. password aging)
- Disables dangerous protocols and services
- Secure configurations for:
  - DNS, sendmail, apache, ftp,…
- Enhances system logging
- Automatic patch downloader
- http://www.bastille-linux.org/
- See also Center for Internet Security
  - http://www.cisecurity.org/bench.html

(Remote) System Scanners

- Nessus
  - http://www.nessus.org/
- Internet Security Systems System Scanner
  - http://www.iss.net/
- WebTrends Security Analyser
  - http://www.webtrends.com
- And many more



Nessus



Summary

Number of hosts tested : 1

Found 1 security holes

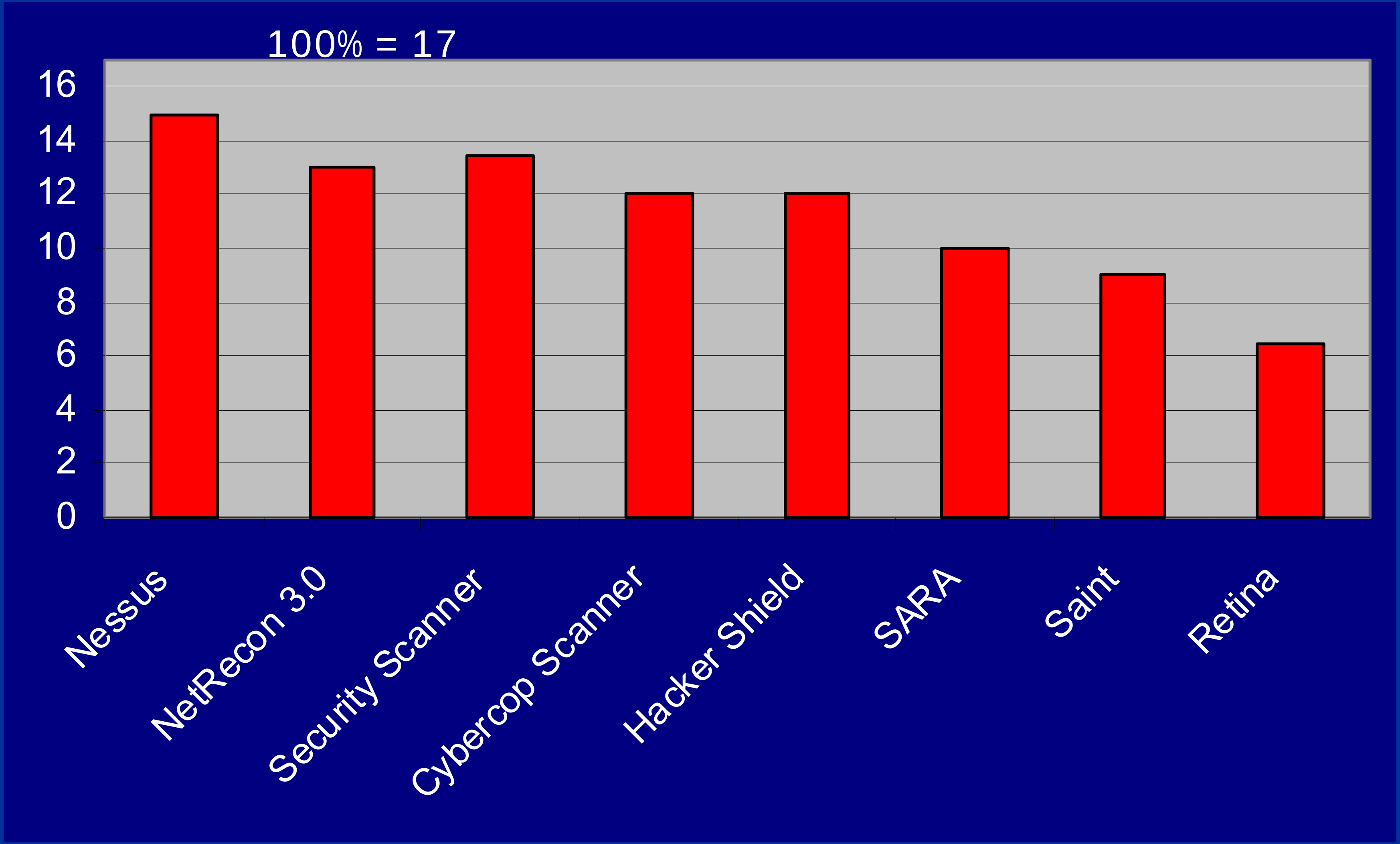
Found 13 security warnings

Found 10 security notes

15.144.89.153

```
⊞ ftp (21/tcp)
⊞ telnet (23/tcp)
⊞ smtp (25/tcp)
⊟ domain (53/tcp)
   ⊞ Security note
   ⊟ Security holes

       The remote BIND server, according to its
       version number, is vulnerable to several
       attacks that can allow an attacker to gain
       root on this system.

       Solution : upgrade to bind 8.2.2-P3
       Risk factor : High
       CVE : CVE-1999-0833

⊞ finger (79/tcp)
⊞ http (80/tcp)
⊞ linuxconf (98/tcp)
— sunrpc (111/tcp)
— auth (113/tcp)
⊞ nntp (119/tcp)
⊞ login (513/tcp)
```

Scanner Effectiveness



Source: Network Computing January, 2001
http://www.networkcomputing.com/

(Local) System Scanners – Tiger

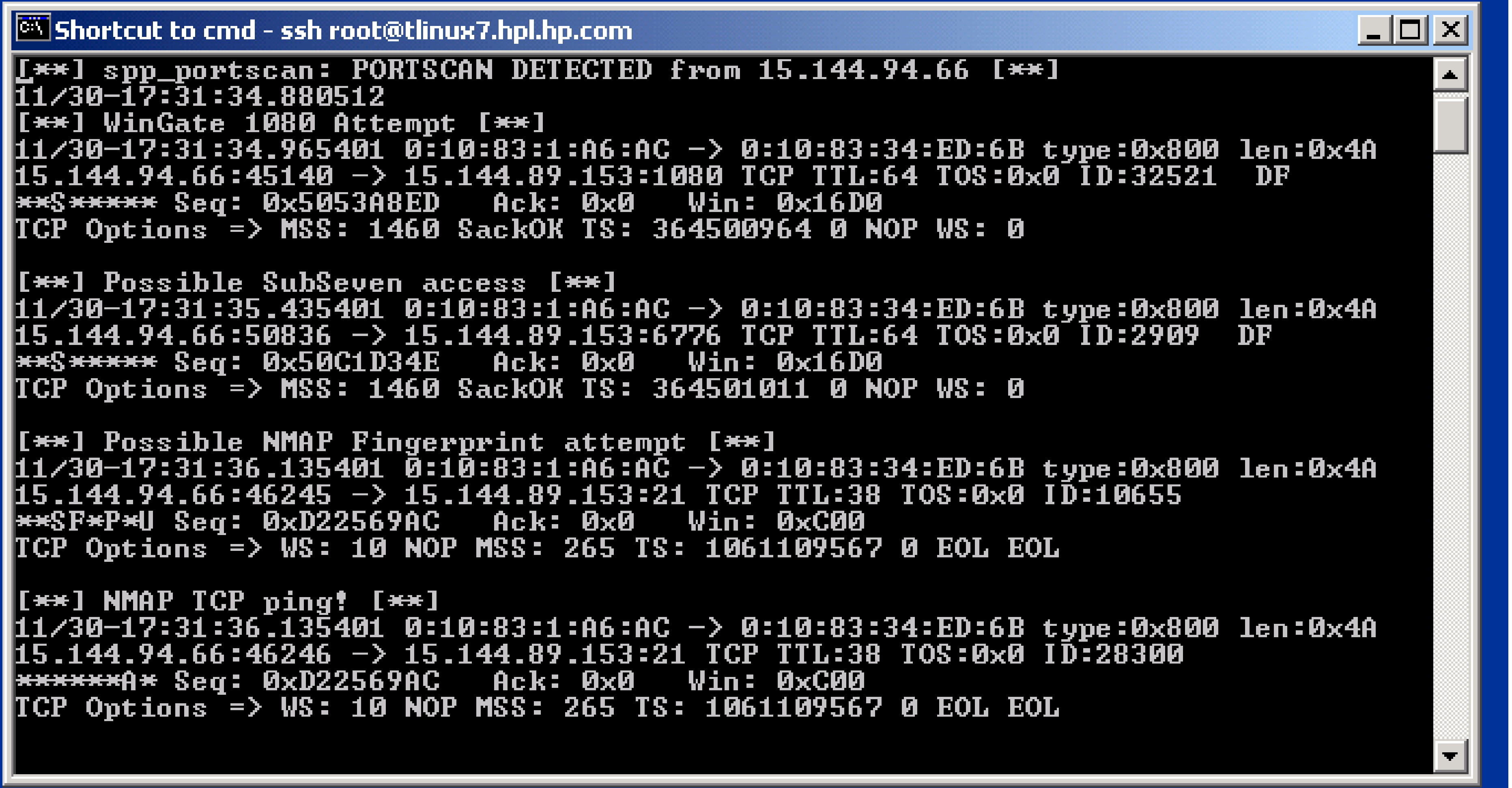
- Local host security scanner
- Looks for local configuration problems
  - PATH problems
  - .rhost files
  - Checks file permission
  - Runs password cracker
- http://www.net.tamu.edu/network/tools/tiger.html
- Some overlap with “lock-down” tools

Snort intrusion detection system

- Packet sniffer and logger
- Potentially can detect various attacks including:
  - Port scans
  - Buffer overflows…
- http://www.snort.org/
- See also tcpdump

Securing Linux Servers

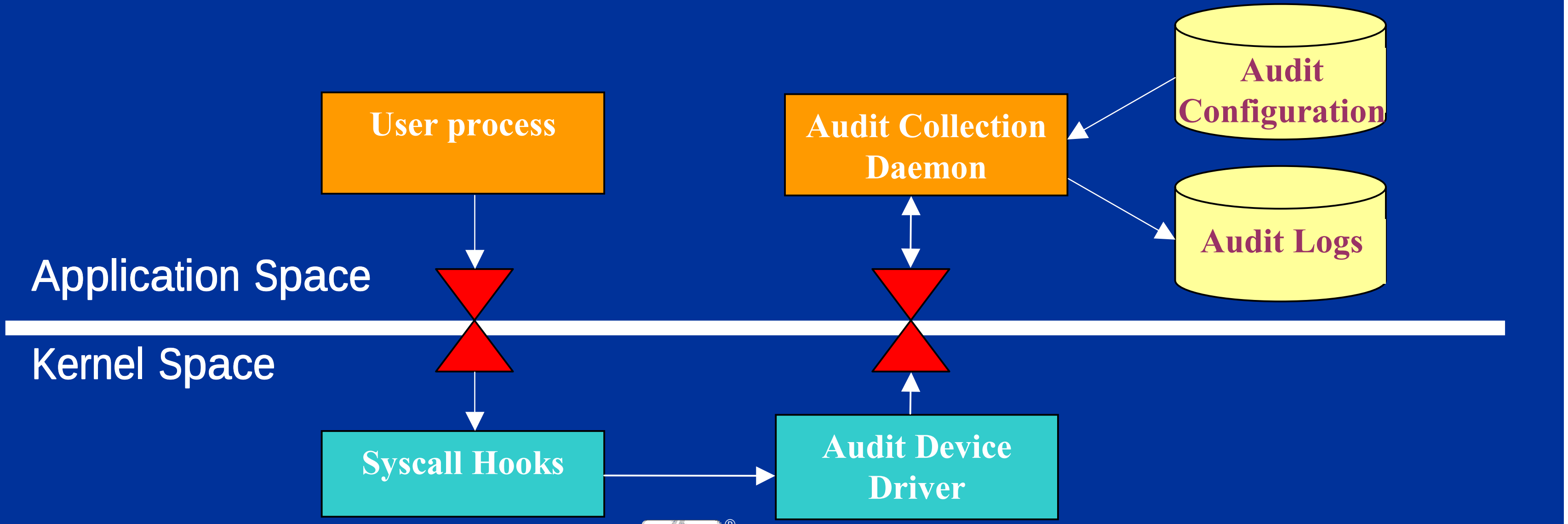
Snort – example output



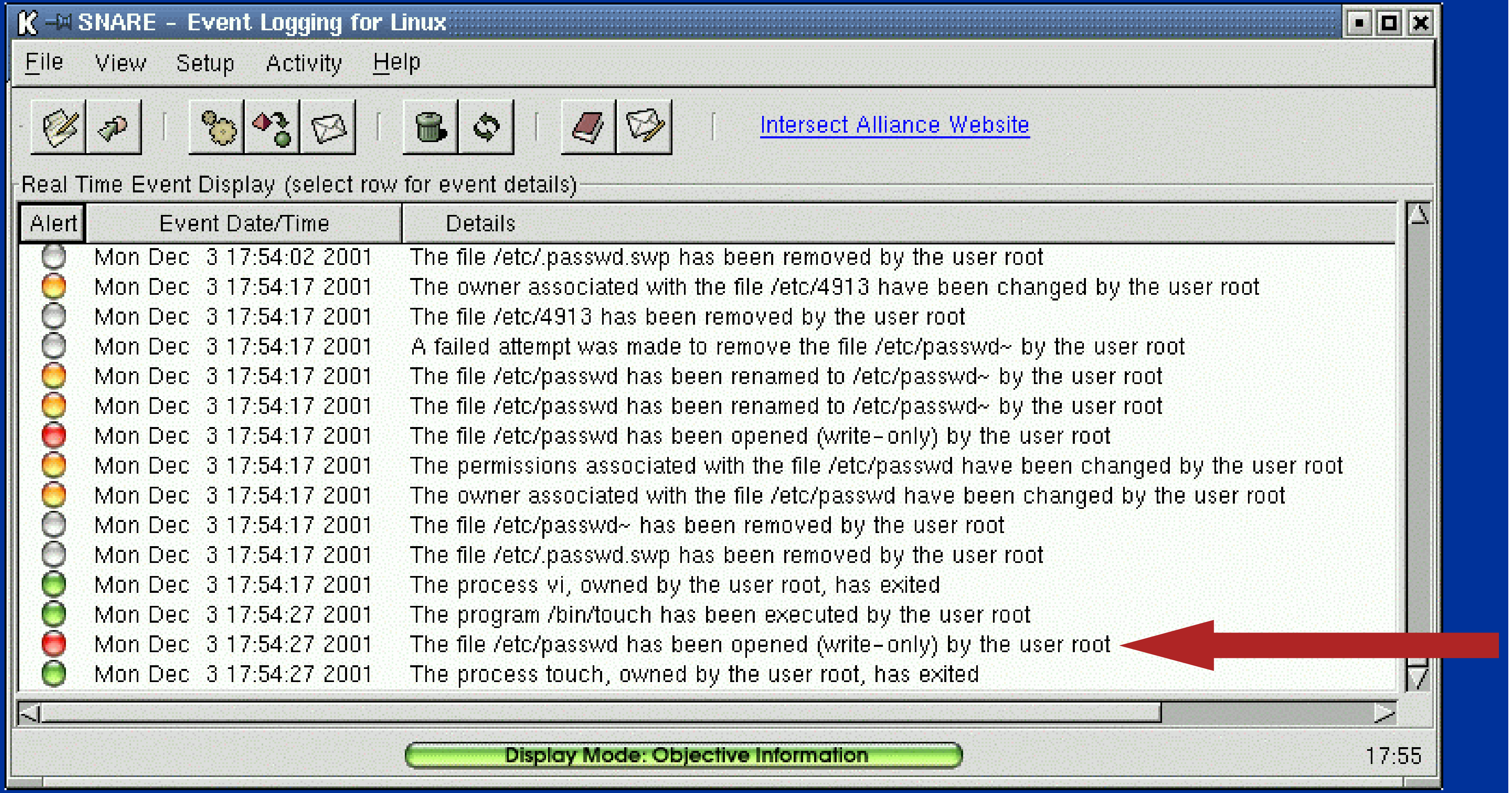
```
[**] spp_portscan: PORTSCAN DETECTED from 15.144.94.66 [**]
11/30-17:31:34.880512
[**] WinGate 1080 Attempt [**]
11/30-17:31:34.965401 0:10:83:1:A6:AC -> 0:10:83:34:ED:6B type:0x800 len:0x4A
15.144.94.66:45140 -> 15.144.89.153:1080 TCP TTL:64 TOS:0x0 ID:32521  DF
**S***** Seq: 0x5053A8ED   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 364500964 0 NOP WS: 0

[**] Possible SubSeven access [**]
11/30-17:31:35.435401 0:10:83:1:A6:AC -> 0:10:83:34:ED:6B type:0x800 len:0x4A
15.144.94.66:50836 -> 15.144.89.153:6776 TCP TTL:64 TOS:0x0 ID:2909  DF
**S***** Seq: 0x50C1D34E   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 364501011 0 NOP WS: 0

[**] Possible NMAP Fingerprint attempt [**]
11/30-17:31:36.135401 0:10:83:1:A6:AC -> 0:10:83:34:ED:6B type:0x800 len:0x4A
15.144.94.66:46245 -> 15.144.89.153:21 TCP TTL:38 TOS:0x0 ID:10655
**SF*P*U Seq: 0xD22569AC   Ack: 0x0   Win: 0xC00
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL

[**] NMAP TCP ping! [**]
11/30-17:31:36.135401 0:10:83:1:A6:AC -> 0:10:83:34:ED:6B type:0x800 len:0x4A
15.144.94.66:46246 -> 15.144.89.153:21 TCP TTL:38 TOS:0x0 ID:28300
******A* Seq: 0xD22569AC   Ack: 0x0   Win: 0xC00
TCP Options => WS: 10 NOP MSS: 265 TS: 1061109567 0 EOL EOL
```

Snare – audit

- Audit collection and analysis tools
  - All system calls can be audited
  - High level objectives can also be defined
- Deciding on a “good” set of audit events is not trivial
- Available free from:
  - http://www.intersectalliance.com/projects/Snare/index.html

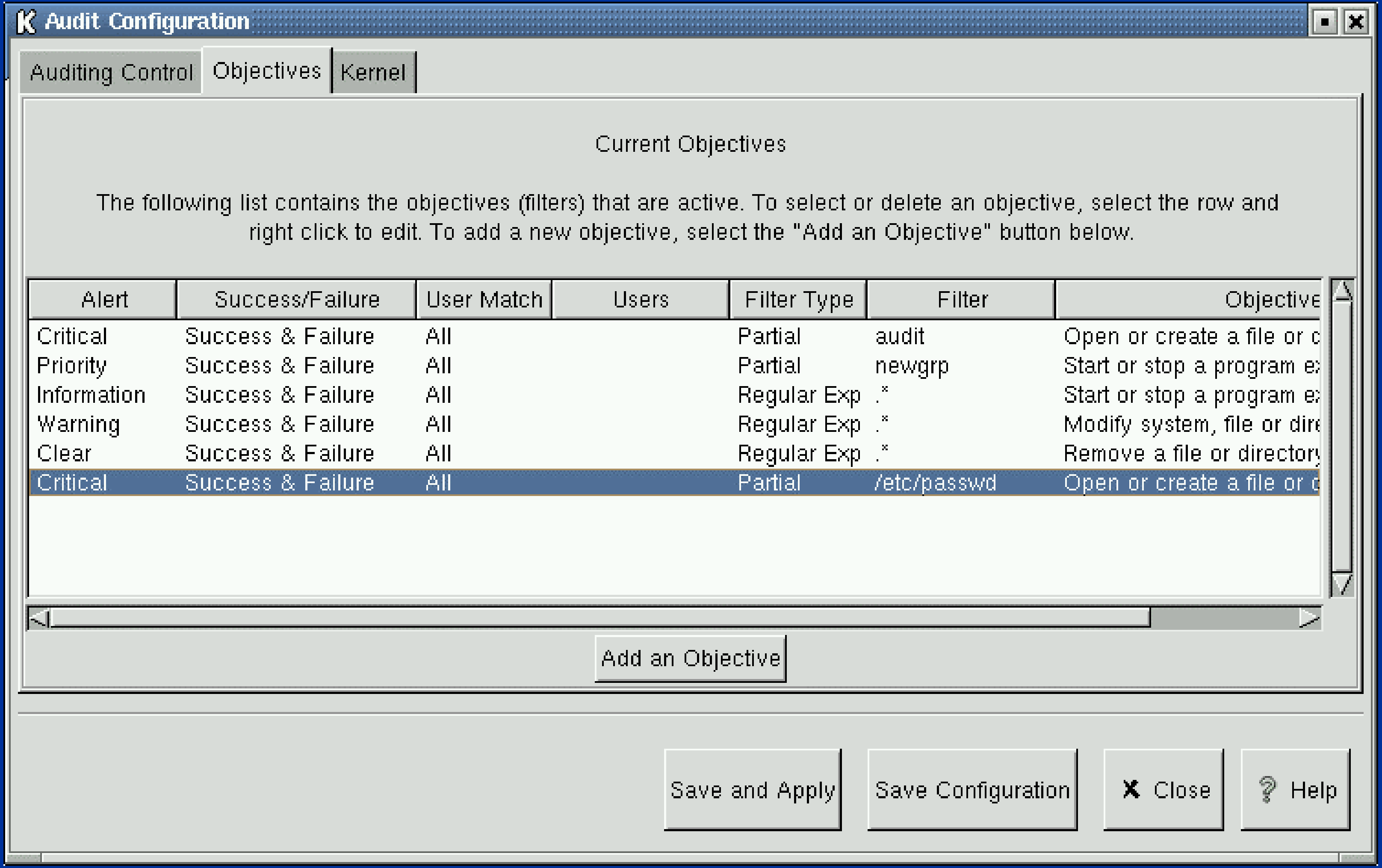


Snare – audit log display


SNARE - Event Logging for Linux
File
View
Setup
Activity
Help
Intersect Alliance Website
Real Time Event Display (select row for event details)
Display Mode: Objective Information
17:55

Alert	Event Date/Time	Details
⚪	Mon Dec 3 17:54:02 2001	The file /etc/.passwd.swp has been removed by the user root
🟠	Mon Dec 3 17:54:17 2001	The owner associated with the file /etc/4913 have been changed by the user root
⚪	Mon Dec 3 17:54:17 2001	The file /etc/4913 has been removed by the user root
⚪	Mon Dec 3 17:54:17 2001	A failed attempt was made to remove the file /etc/passwd~ by the user root
🟠	Mon Dec 3 17:54:17 2001	The file /etc/passwd has been renamed to /etc/passwd~ by the user root
🟠	Mon Dec 3 17:54:17 2001	The file /etc/passwd has been renamed to /etc/passwd~ by the user root
🔴	Mon Dec 3 17:54:17 2001	The file /etc/passwd has been opened (write-only) by the user root
🟠	Mon Dec 3 17:54:17 2001	The permissions associated with the file /etc/passwd have been changed by the user root
🟠	Mon Dec 3 17:54:17 2001	The owner associated with the file /etc/passwd have been changed by the user root
⚪	Mon Dec 3 17:54:17 2001	The file /etc/passwd~ has been removed by the user root
⚪	Mon Dec 3 17:54:17 2001	The file /etc/.passwd.swp has been removed by the user root
🟢	Mon Dec 3 17:54:17 2001	The process vi, owned by the user root, has exited
🟢	Mon Dec 3 17:54:27 2001	The program /bin/touch has been executed by the user root
🔴	Mon Dec 3 17:54:27 2001	The file /etc/passwd has been opened (write-only) by the user root
🟢	Mon Dec 3 17:54:27 2001	The process touch, owned by the user root, has exited

Snare configuration (1/2)

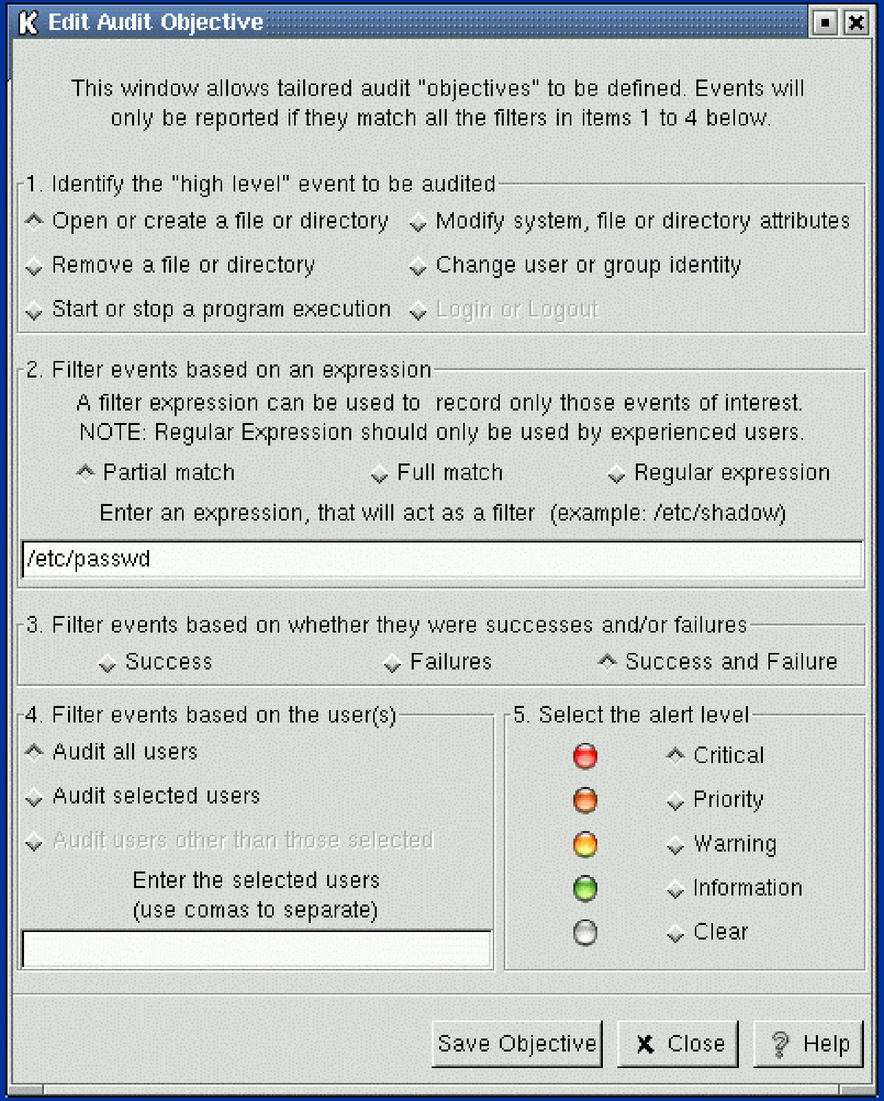


Current Objectives

The following list contains the objectives (filters) that are active. To select or delete an objective, select the row and right click to edit. To add a new objective, select the "Add an Objective" button below.

Alert	Success/Failure	User Match	Users	Filter Type	Filter	Objective
Critical	Success & Failure	All		Partial	audit	Open or create a file or c
Priority	Success & Failure	All		Partial	newgrp	Start or stop a program e:
Information	Success & Failure	All		Regular Exp	.*	Start or stop a program e:
Warning	Success & Failure	All		Regular Exp	.*	Modify system, file or dire
Clear	Success & Failure	All		Regular Exp	.*	Remove a file or directory
Critical	Success & Failure	All		Partial	/etc/passwd	Open or create a file or c

Snare configuration (2/2)



Edit Audit Objective

This window allows tailored audit "objectives" to be defined. Events will only be reported if they match all the filters in items 1 to 4 below.

1. Identify the "high level" event to be audited

◆ Open or create a file or directory ◇ Modify system, file or directory attributes

◇ Remove a file or directory ◇ Change user or group identity

◇ Start or stop a program execution ◇ Login or Logout

2. Filter events based on an expression

A filter expression can be used to record only those events of interest.
NOTE: Regular Expression should only be used by experienced users.

◆ Partial match ◇ Full match ◇ Regular expression

Enter an expression, that will act as a filter (example: /etc/shadow)

/etc/passwd

3. Filter events based on whether they were successes and/or failures

◇ Success ◇ Failures ◆ Success and Failure

4. Filter events based on the user(s)

◆ Audit all users

◇ Audit selected users

◇ Audit users other than those selected

Enter the selected users
(use comas to separate)

5. Select the alert level

🔴 ◆ Critical

🟠 ◇ Priority

🟡 ◇ Warning

🟢 ◇ Information

⚪ ◇ Clear

Psionic PortSentry, LogCheck and HostSentry

- Psionic PortSentry
  - Detects and stops port scan
- Psionic LogCheck
  - Scans system logs for security violations
- Psionic HostSentry
  - Detects login anomalies
- Available free: http://www.psionic.com/

Tripwire (& friends)

- Intrusion detection
- Periodically (e.g. daily) scans files to detect changes
- Email notification to administrator
  - Update tripwire database or…
  - Manually revert file
- Included in some distributions (e.g. Red Hat 7.1)
  - http://www.tripwire.org/
- ViperDB – an alternative to Tripwire
  - http://www.resentment.org/projects/viperdb/
- Chkrootkit - http://www.chkrootkit.org/

Limitations of patching and layered security utilities

- Not all vulnerabilities and bugs are known
  - A patch may not be available
  - You may not apply the patch in time
  - The security utilities are generally ineffective against unknown vulnerabilities
- Security utilities do not generally detect all known vulnerabilities
- Limited protection against accidental misconfiguration of applications
- Kernel root kits are extremely difficult to detect

Agenda

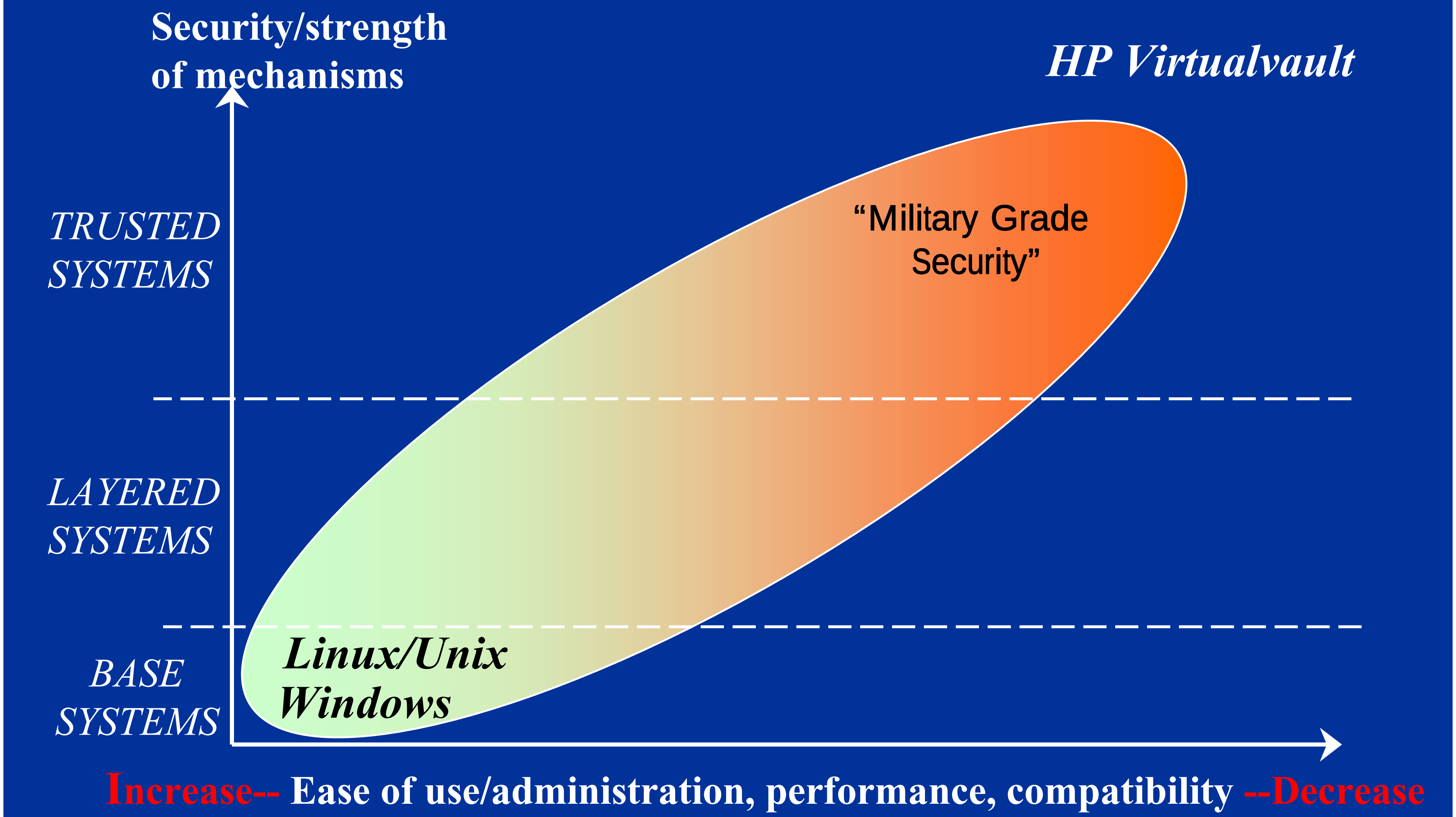
- Patching
- Tools and utilities
- **Kernel strengthening**
  - **Some background & concepts**
  - **Some examples**

Kernel strengthening

- Philosophy
  - Bugs are inevitable
    - You cannot not know what a program will do until it runs
  - Misconfiguration and administration errors are inevitable
  - Attempt to contain the damage
    - “A sandbox” limits access to system and network resources
- Disadvantage
  - Can require extensive integration work to “port” an application or service
  - Administration complexity
  - Kernel code changes

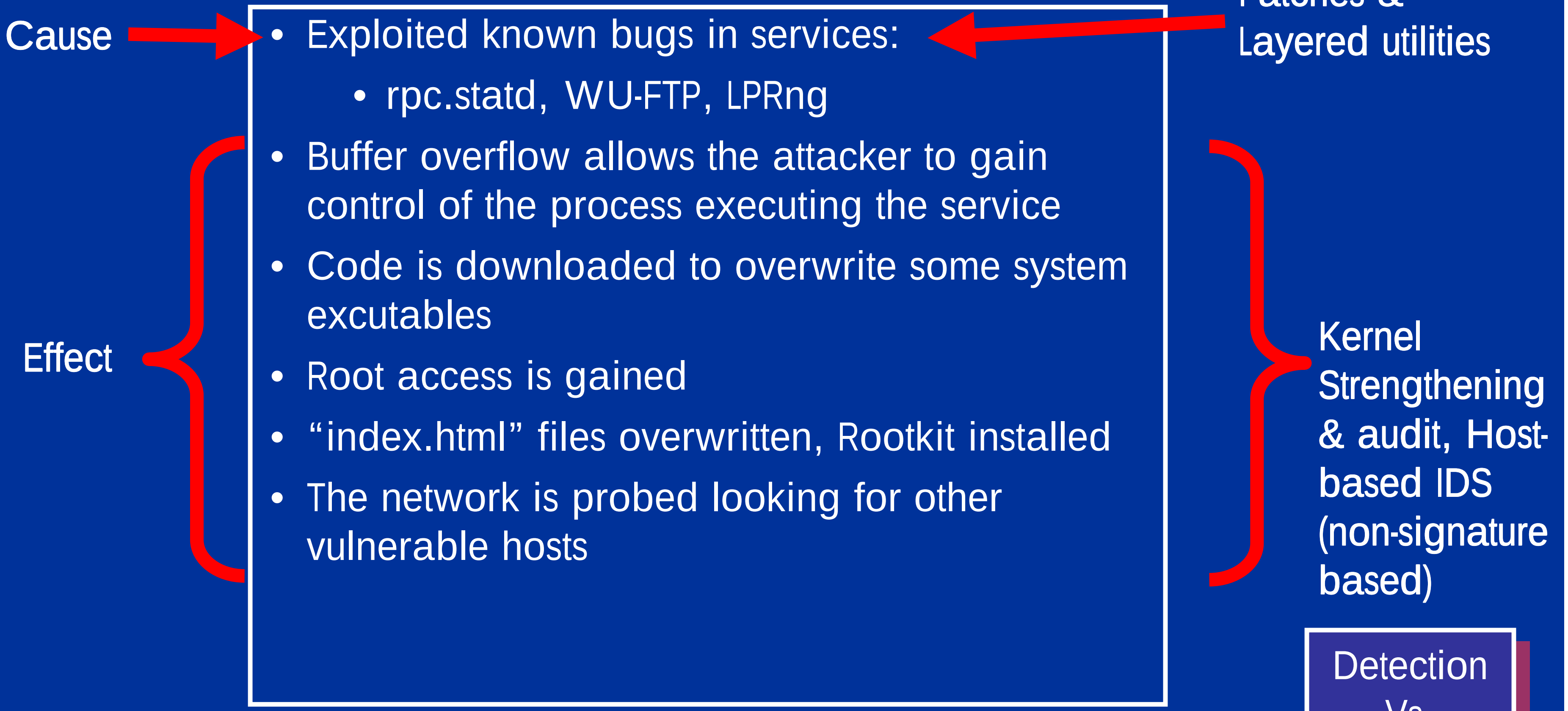
Securing Linux Servers

The Security and OS Landscape


Security/strength of mechanisms
HP Virtualvault
TRUSTED SYSTEMS
"Military Grade Security"
LAYERED SYSTEMS
BASE SYSTEMS
Linux/Unix
Windows
Increase-- Ease of use/administration, performance, compatibility --Decrease

Kernel strengthening versus layered utilities and patching

Attack Pathology



Cause

Patches & Layered utilities

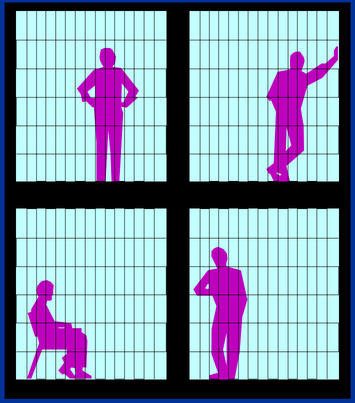
- Exploited known bugs in services:
  - rpc.statd, WU-FTP, LPRng
- Buffer overflow allows the attacker to gain control of the process executing the service
- Code is downloaded to overwrite some system excutables
- Root access is gained
- “index.html” files overwritten, Rootkit installed
- The network is probed looking for other vulnerable hosts

Effect

Kernel Strengthening & audit, Host-based IDS (non-signature based)

Detection Vs. Prevention

Containment – a key concept



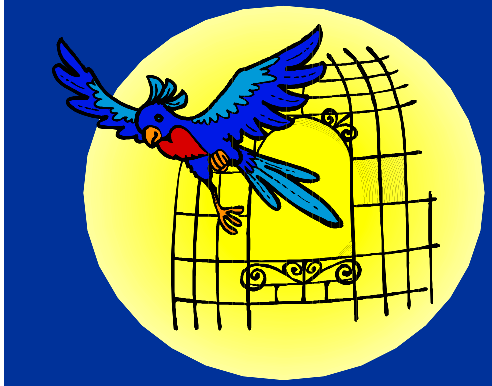
CONTAINMENT

- Contain a process to a known part of the system
  - Define and fix the resources available to it
    - system and network
  - Define and fix the privileges available to it
    - principle of "Least-Privilege"
- Boundaries are defined around the known "correct" behavior

Discretionary access control (DAC)

```
-rw-r--r--   1 nje        users      3083399 Nov 18  1998 BellLaPadula.pdf
-rw-r--r--   1 nje        users      8082702 Nov  1  2000 ande72.pdf
-rw-r--r--   1 nje        users      1580903 Nov  1  2000 ande80.pdf
-rw-r--r--   1 nje        users       433265 Nov  1  2000 dod85.pdf
```

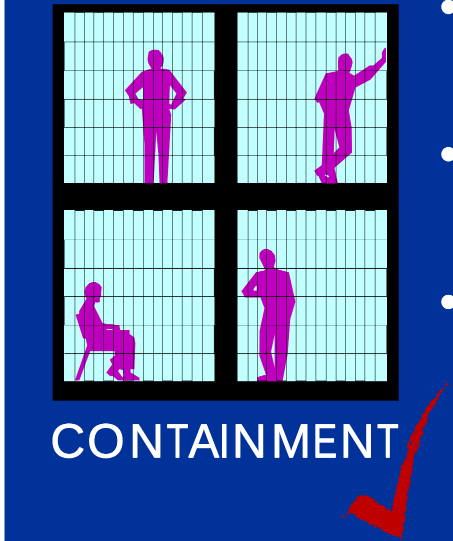
- File owner can grant access to anybody
- DAC does not give good containment properties
  - Users can change who gets access to different parts of the system
  - Users can be tricked into updating files which they own to introduce “Trojan Horses”



CONTAINMENT ✗

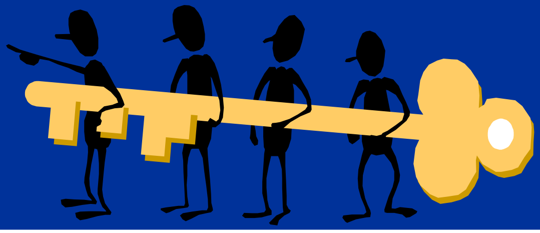
Mandatory Access Control

- Mandatory Access Control
  - Access control beyond the discretion of the owner
  - Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, December, 1985
- Important for protecting sensitive files
  - Web pages, executables, …
- Important for protecting other system resources
  - E.G. communication channels
- Important for constraining the user/process to a known part of the system - CONTAINMENT



LIDS – Linux Intrusion Detection System (1/3)

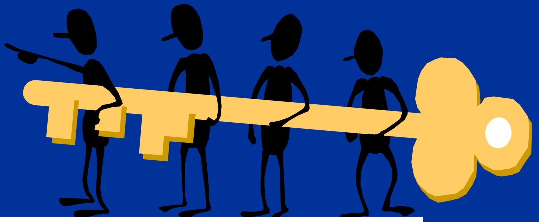
- Novel mandatory access control and least privilege model built into Linux
- Port scan detector built into the kernel
- Seals the kernel to prevent new kernel modules being loaded
- ACLS on files and directories



```
    lfs# lidsadm -A -o /etc -j READ
    lfs# lidsadm -A -o /etc/motd -j WRITE
    lfs# lidsadm -A -o /etc/shadow -j DENY
    lfs# lidsadm -A -s /usr/sbin/sshd -o \
    /etc/shadow -j READ
```

LIDS – Linux Intrusion Detection System (2/3)

- LIDS uses and extends Linux capabilities to implement “Least Privilege”
- Capabilities can be removed and added to the kernel bounding set without rebooting
  - e.g. CAP_KILL can be removed/added
- New capabilities introduced
  - e.g. CAP_INIT_KILL and CAP_HIDDEN
- Capabilities can be granted to specific executables



```
lfs# lidsadm -A -s /usr/sbin/httpd \
-o CAP_BIND_NET_SERVICE 80-80,443-443 -j GRANT
```

LIDS – Linux Intrusion Detection System (3/3)

- LIDS does not explicitly control networking or inter-process communication
- Installation – download patch and rebuild your kernel
- For more information:

An Overview of LIDS:
http://www.securityfocus.com/cgi-bin/infocus.pl?id=1496
also:
http://www.lids.org/

Immunix (1/3)

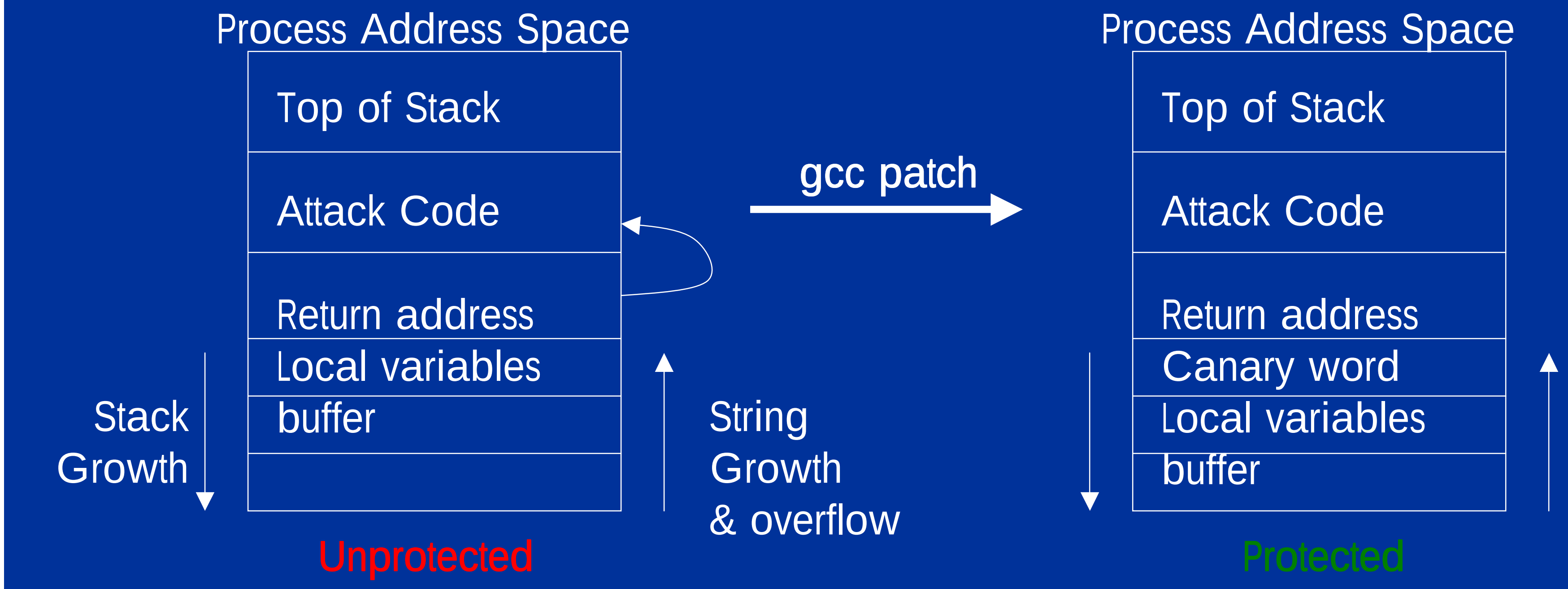
- Mandatory Access Control & explicit protection against certain common attacks
- SubDomain confines programs to explicitly declared files

```
foo {
        /etc/readme r ,
        /etc/writeme w ,
        /usr/bin/bar x +{/etc/otherwrite w} ,
        /usr/bin/baz x -{/etc/writeme w} ,
}
```

- No constraints on network access

Immunix (2/3)

- StackGuard
  - Protects against most “stack smashing” attacks
  - Patch to gcc + recompilation of code

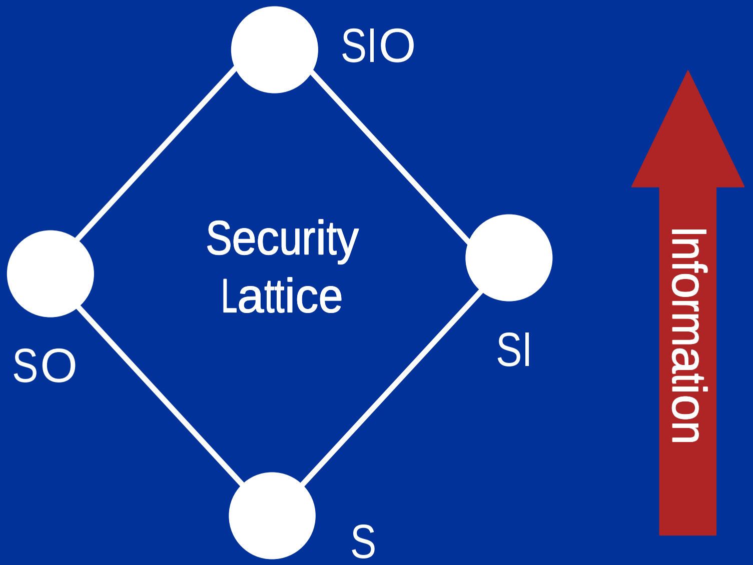


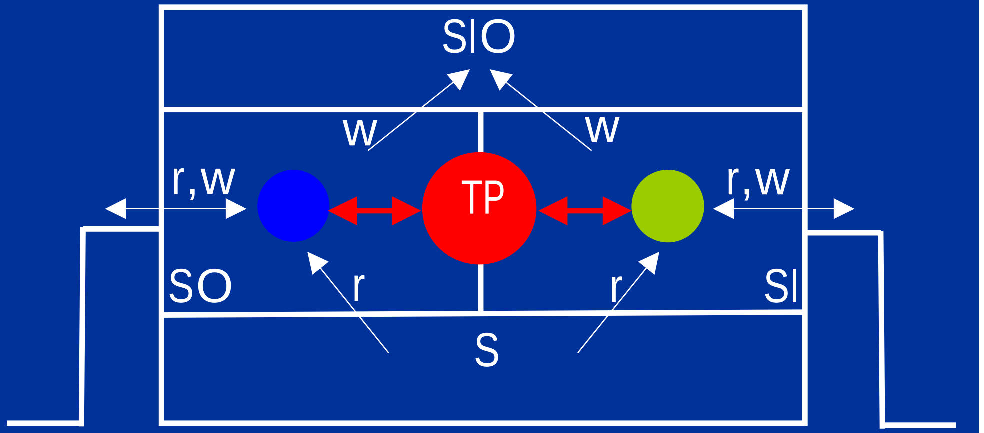
Immunix (3/3)

- FormatGuard
  - Protects against most format string attacks
  - Patch to glibc + recompilation of code
- RaceGuard
  - Protects against the symlink attack in /tmp
- Openwall kernel patch
- Availability
  - http://www.immunix.org/
  - Free for non commercial use

Some theory – Multi-Level Security and Bell – La Padula

- Developed for military use in 1973
- Information assigned hierarchical levels (Secret, Top Secret) and non-hierarchical categories (Project1, Project2..)
- In implementations "root" is replaced by multiple privileges





Some theory – Type Enforcement (1/2)

Domains

Types

	Type			
Domain	html	apache_c	srvlets	system_f
apache	r	r	n	n
tomcat	n	n	r	n
web_auth	rw	n	rw	n
system_p	n	n	n	rw

Domain Definition Table (DDT)

Some theory – Type Enforcement (2/2)

Domains

Domains

	Domain			
Domain	apache	tomcat	web_auth	system_p
apache	rw	rw	n	n
tomcat	rw	rw	n	n
web_auth	n	n	rw	n
system_p	n	n	n	rw

Domain Interaction Table

Type Enforcement in perspective

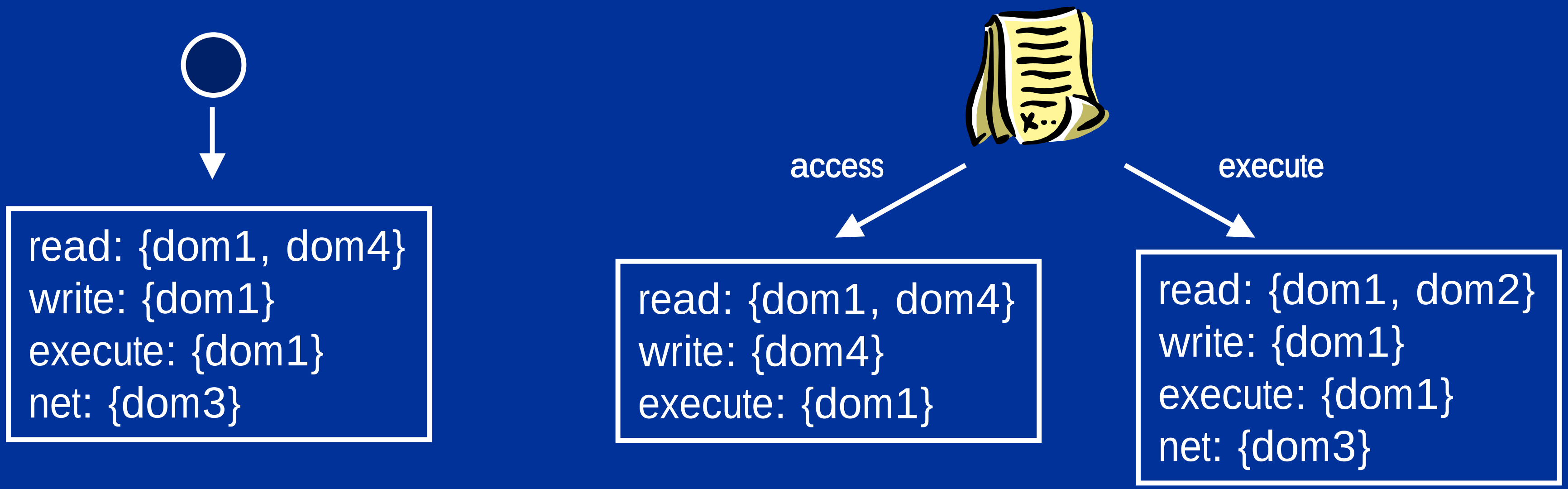
- Proposed in 1985 by Boebert and Kain
  - Information flows are expressed explicitly
  - Tables can get very complicated
- 1995 Badger et al. proposed abstract C-like language for defining policies (Domain and Type Enforcement)
  - Hosts (IP addresses mapped to domains)
- A superset of MLS
  - Flexibility can lead to complexity
- Limited expressiveness for domain-domain interaction
  - OS versus model abstraction mismatch
  - Not able to express limitations on ports, channels etc.
  - Is the indirection introduced by the tables necessary?

Security Enhanced Linux

- Implemented by the National Security Agency
- Fundamentally a Type Enforcement system
- Demonstrates the flexibility of TE for MAC
    - Support for Multi-Level Security
    - Support for Role Based Access Control
        - Roles are a set of domains
        - Can express hierarchies of roles
        - Role transitions can be defined
            - (system_r, login_exec_t) -> login_r
- http://www.nsa.gov/selinux/index.html

PitBull LX (1/2)

- Implements MAC via enhanced Type Enforcement system
  - Each entity assigned a set of zero or more domains



read: {dom1, dom4}
write: {dom1}
execute: {dom1}
net: {dom3}

read: {dom1, dom4}
write: {dom4}
execute: {dom1}

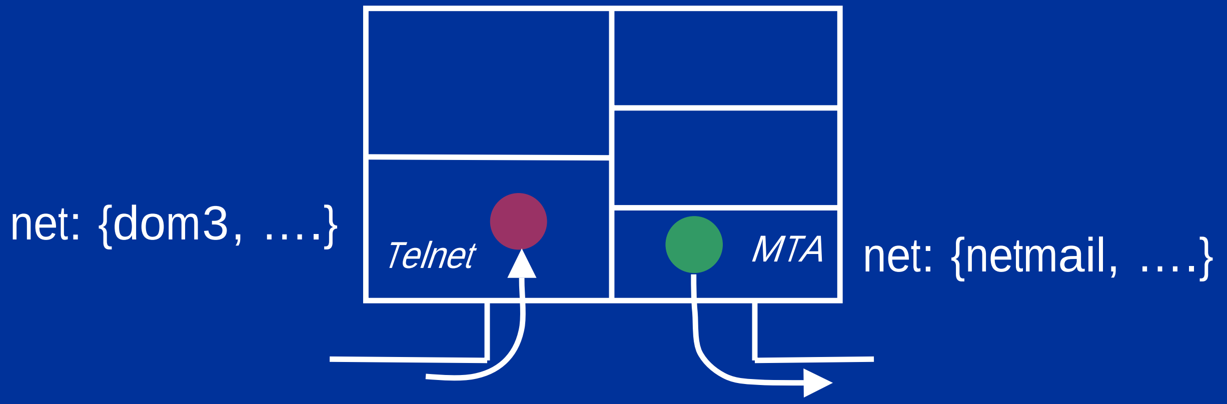
read: {dom1, dom2}
write: {dom1}
execute: {dom1}
net: {dom3}

- Access rule: ds(process) ⊇ access-ds(file)
- Execution inheritance rule: ds(child) = ds(parent) ∩ ds(executable)
  or ds(child) = ds(parent) ∪ ds(executable)

Securing Linux Servers

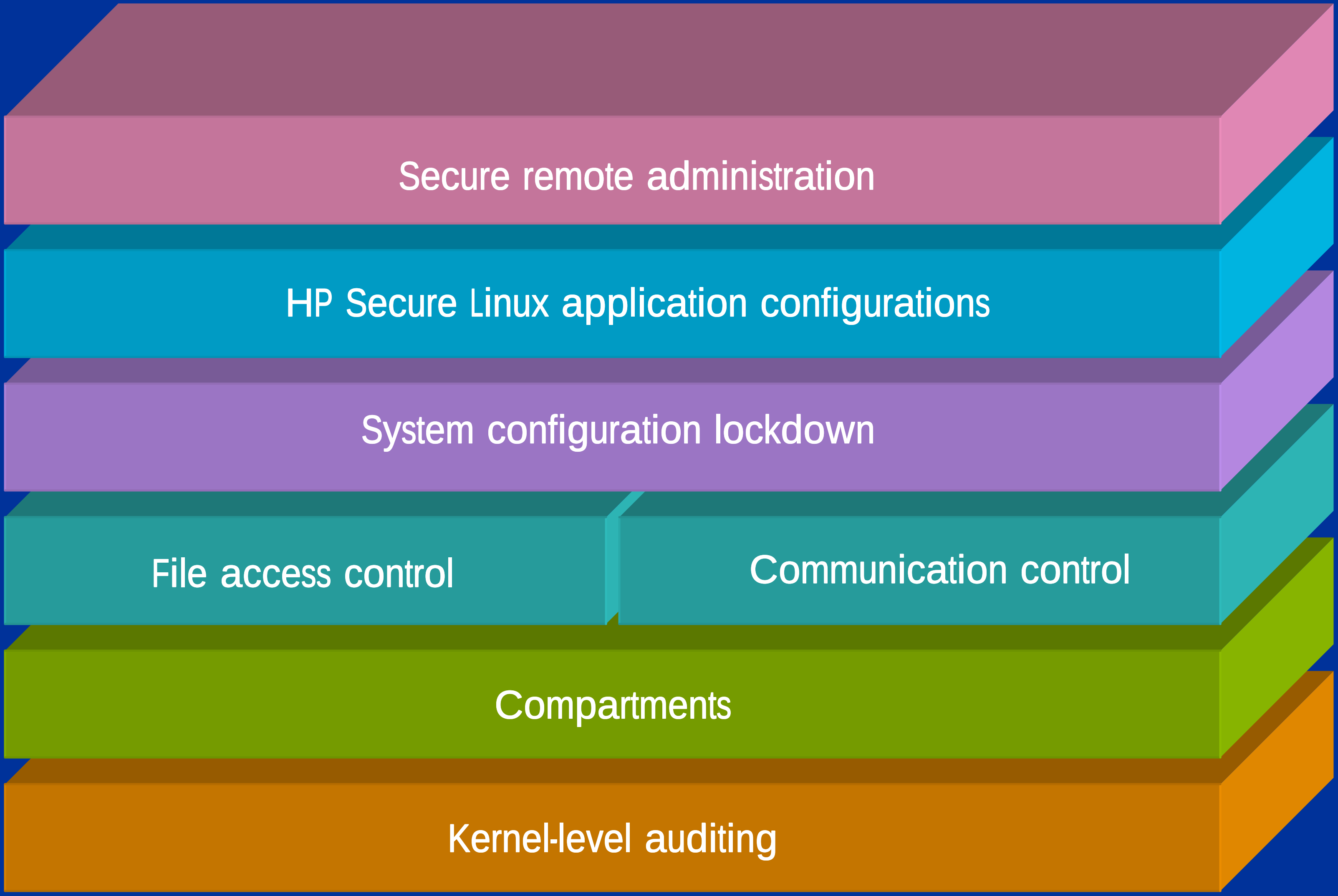
PitBull LX (2/2)

- Domain assignment to network endpoints
  - proto:tcp daddr:localserv dport:23 domain:dom3
  - proto:tcp daddr:remoteserv dport:25 domain:netmail
- Access rule (ds-process) $\cap$ ds(endpoint) $\neq \varnothing$



- More details: http://www.argus-systems.com/

HP Secure OS Software for Linux



Secure remote administration

HP Secure Linux application configurations

System configuration lockdown

File access control

Communication control

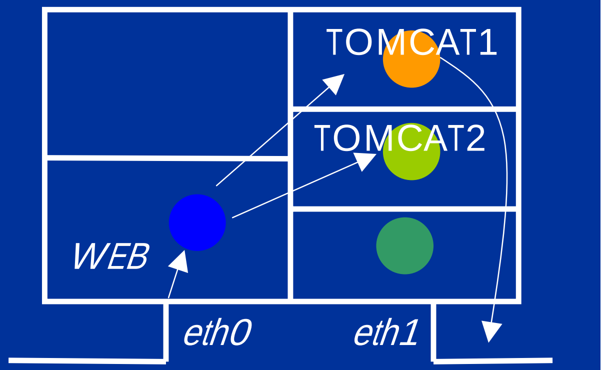
Compartments

Kernel-level auditing

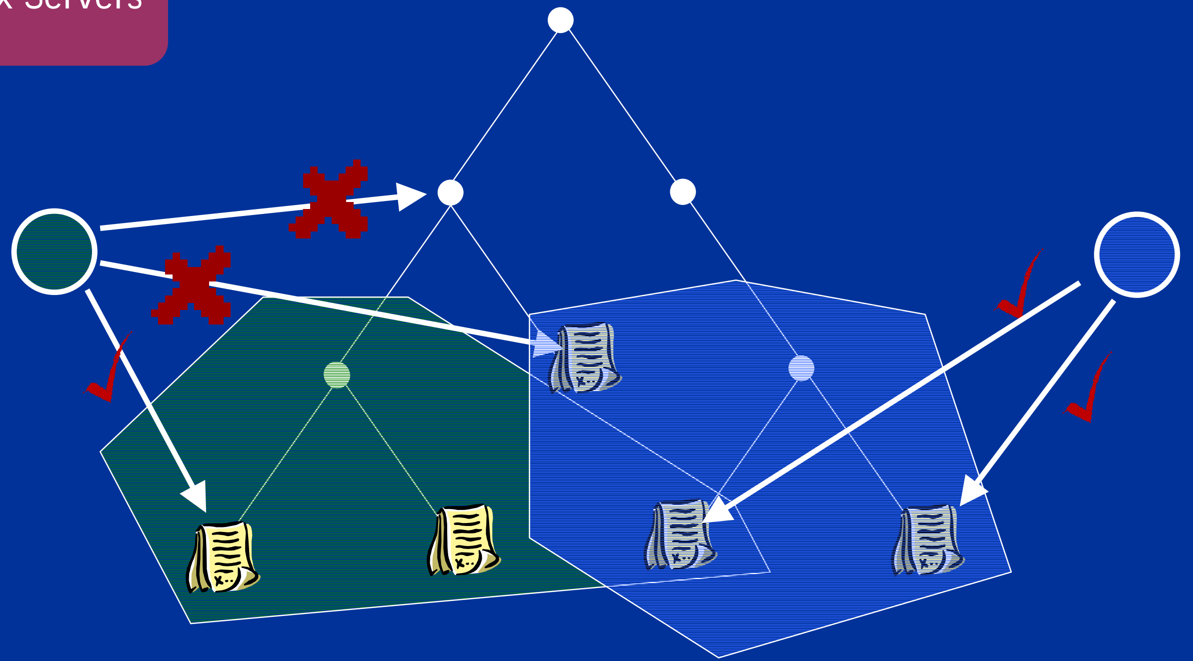
HP Secure Linux Compartment Communication Rules

```
HOST:* -> COMPARTMENT:WEB
          METHOD TCP PORT 80 NETDEV eth0

COMPARTMENT:WEB -> COMPARTMENT:TOMCAT1
          METHOD TCP PORT 8007 NETDEV lo

COMPARTMENT:WEB -> COMPARTMENT:TOMCAT2
          METHOD TCP PORT 8008 NETDEV lo

COMPARTMENT:TOMCAT1 -> HOST:SERVER1
          METHOD TCP PORT 8080 NETDEV eth1
```

Explicit paths in hp secure Linux



HP Secure Linux: File system protection



- File Control Table specifies compartment access: read, write, append
  - Fine-grain control and coarse grain (per file, per directory)

```
web /compt/web/apache/logs read,write
web /compt/web/dev         read,write
web /compt/web/tmp         read,write
web /compt/web             read
```

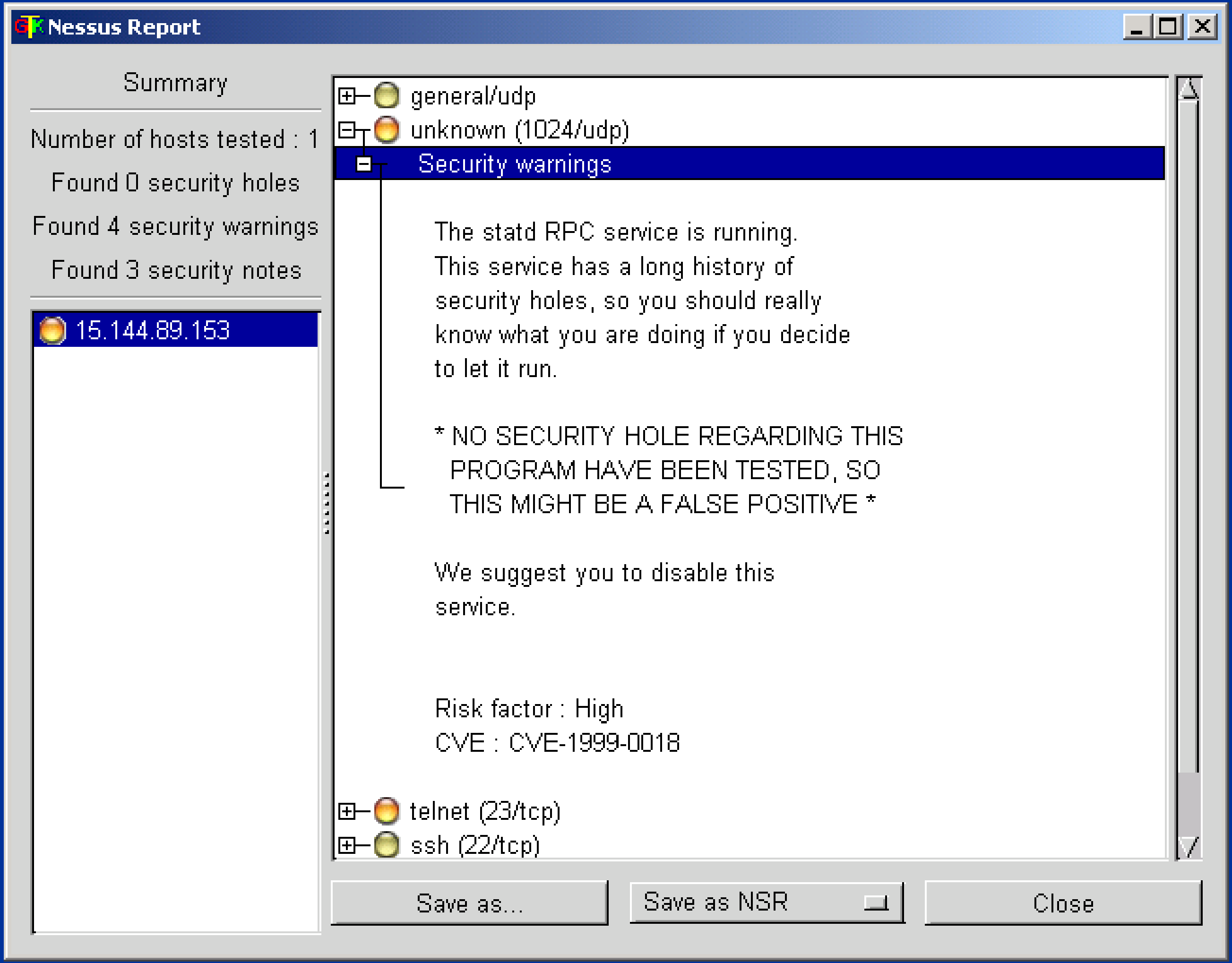
- Tripwire – Integrity protection
- More details: http://www.hp.com/security/products/linux/

Conclusion

- Each approach has pros and cons
- The most secure approach combines all three
  - Patching
  - Layered Security products
  - Kernel strengthening system
- The best protection against “unknown” attacks is to strengthen the operating system
  - Containment
    - MAC for all resource access

Backup Slides follow

Nessus



Summary

Number of hosts tested : 1

Found 0 security holes

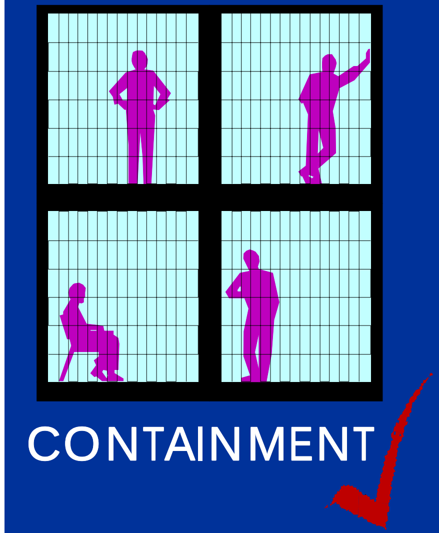
Found 4 security warnings

Found 3 security notes

15.144.89.153

```
The statd RPC service is running.
This service has a long history of
security holes, so you should really
know what you are doing if you decide
to let it run.

* NO SECURITY HOLE REGARDING THIS
  PROGRAM HAVE BEEN TESTED, SO
  THIS MIGHT BE A FALSE POSITIVE *

We suggest you to disable this
service.


Risk factor : High
CVE : CVE-1999-0018
```

Principle of "Least Privilege"



- A subject (process) only gets the minimum privilege it needs to perform its intended function

- Constrains the actions a process can perform
  - Example Linux Capability model
  - CAP_KILL, CAP_DAC_OVERRIDE, CAP_SYS_ADMIN,…

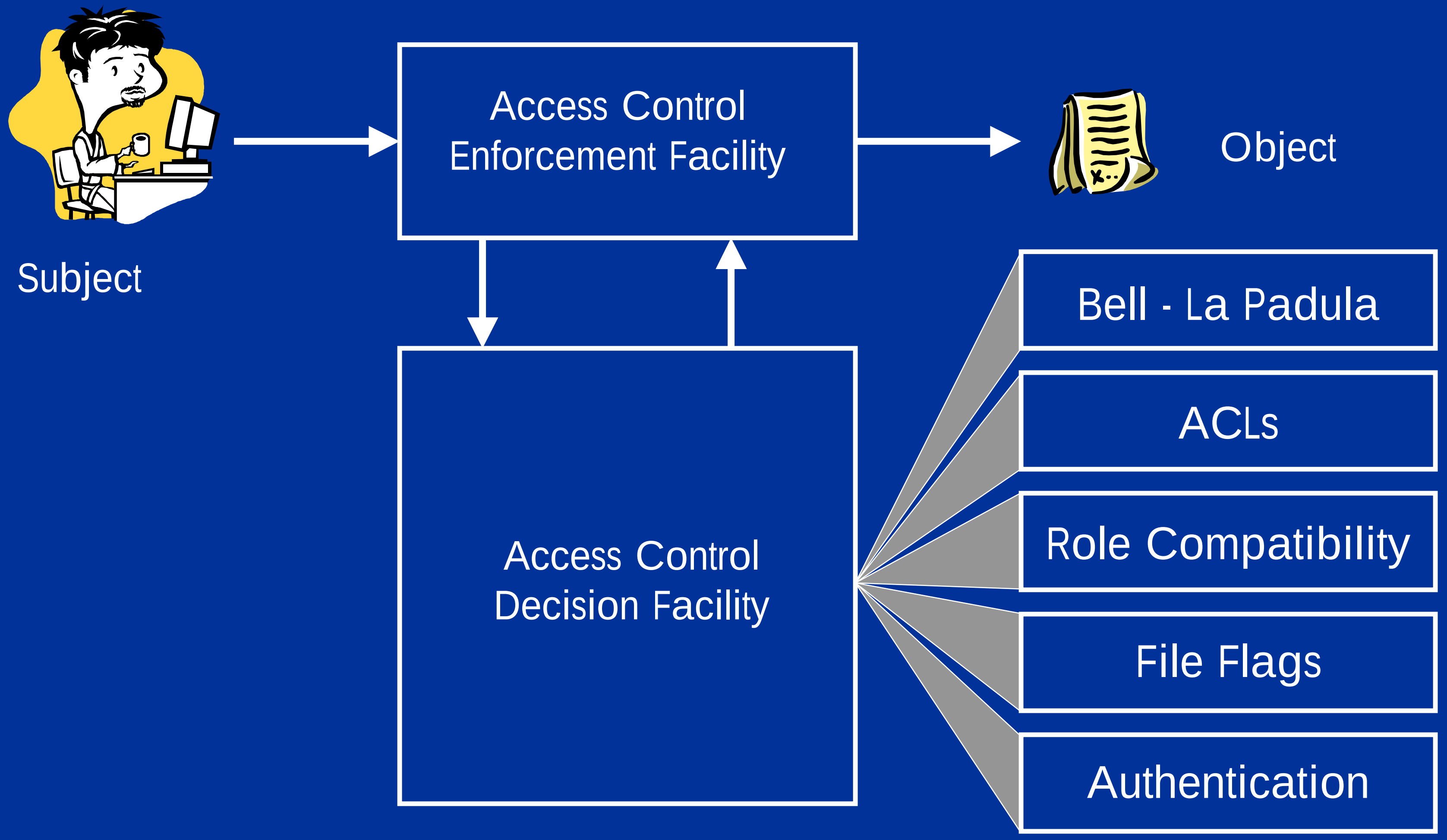
- Constrains the "roles" available to a process
  - Control the ID with which a process can execute
  - Prevent "root" access

- Important flexibility versus usability trade-offs need to be made

The Openwall kernel patch

- Non executable stack
  - This may stop some things working
- Restrictions on links in /tmp
- Restricted /proc
- Special handling of file descriptors 0, 1 and 2
- ….
- Available free from: http://www.openwall.com/linux/

Rule Set Based Access Control (RSBAC)



- 9 installable security modules

The RSBAC “Role Compatibility Model”

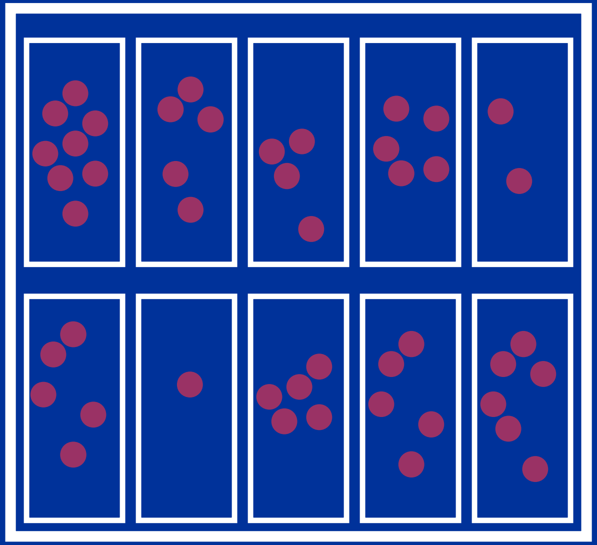
role

	Access Type			
role	a1	a2	a3	a4
r1	y	n	n	y
r2	n	n	n	n
r3	y	y	y	y
r4	n	n	n	n

RSBAC Security model

- The nine supported models include some novel features
- File Flags model (for Files and Directories) defines MAC read_only, execute_only, …
  - Prevents critical files being overwritten
- Authentication model restricts the ID a process or executable can run under
  - Restricts “root exploits”
- All nine models can be loaded simultaneously
- http://www.rsbac.org/

Virtual private servers



- Use chroot to isolate vservers into a virtual file system
- Processes in other vservers are hidden
- Separate IP addresses for each vserver
- Capabilities of “root” in each vserver is constrained
- Vserver 1 & 0 are special – can see and manipluate other vservers
- Available free: http://www.solucorp.qc.ca/miscprj/s_context.hc

Others

- Trustix - http://www.trustix.net/
- Owl - http://www.openwall.com/Owl/
- EnGuarde - http://www.engardelinux.org/
  - Includes LIDS, Snort and Tripwire
- Blue Linux - http://bluelinux.org/
  - Aims to include patch server
- Castle - http://castle.altlinux.ru/
  - Includes RSBAC
- Kaladix - http://www.kaladix.org/
  - Includes RSBAC, Snort and Tripwire