GENERAL DYNAMICS
Mission Systems

®

PitBull® Executive Training

Scope of Presentation

- Basics principles
  - multilevel security (MLS) and cross domain solution (CDS)
  - protection levels
  - operating system security
- PitBull
  - what PitBull is and how it fits into the cybersecurity landscape
  - what features are included in PitBull
  - how PitBull is used to protect and enable
  - how PitBull impacts software and system design

PitBull at a Glance

- PitBull is **Linux distribution** that …
  - is based on and **compatible with RHEL** (Red Hat Enterprise Linux)
    - currently RHEL 6, RHEL 7-based release in 2018
    - includes some significant kernel modifications
  - is **extremely hardened** to protect all aspects of the system
    - the system, software applications, and data
    - windowed environments and GUIs
    - user and administrator accounts
    - network access and usage
  - has **new security capabilities** that enable complex, secure architectures
    - full BLP MLS mandatory access control
    - Biba integrity, roles, privileges and other controls
  - is used as
    - a GUI-based desktop (thick client or thin client)
    - a network server for browser-based user access to services
    - a server for hosting files services, mail, chat, databases, etc.

Basic Principles and Terms

Basic Principles - CDS

A CDS is a controlled interface which operates between two security domains.
The Unified Cross Domain Services Management Office (UCDSMO) defines three cross domain solution categories:

TRANSFER

- ***Data Transfer Solutions***: These interconnect networks or information systems that operate in different security domains and transfer information between them.

ACCESS

- ***Access Solutions:*** These provide simultaneous visualization of information from multiple security domains via a single workstation without any data transfer between the various domains.

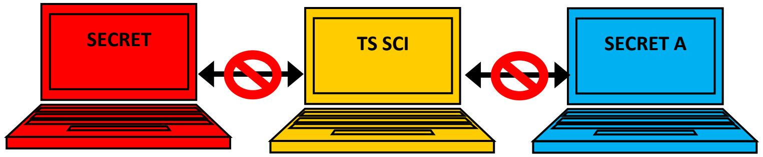
ENTERPRISE

- ***Multilevel Solutions***: These store and process information from different security domains of various security classifications and permit access and relabeling based on user clearances and authorizations

Multiple Security Domain Architectures

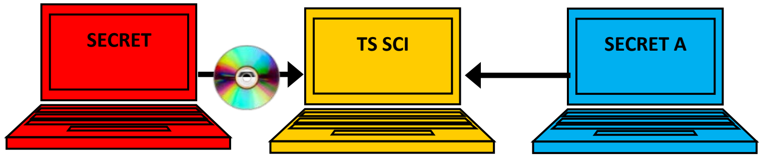
Multiple Security Levels

- Physical separation of systems & data
- Multiple computers, multiple networks
- No transfer of data between systems



System High

- Data Transfer Solution
- No separation of data
- All data treated becomes System High



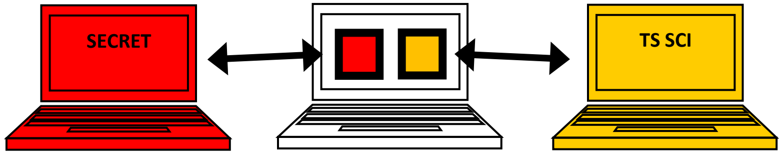
Personnel must all be cleared at highest level
Expensive point to point guards

MSL | SH

MILS | MLS

Multiple Independent Levels of Security

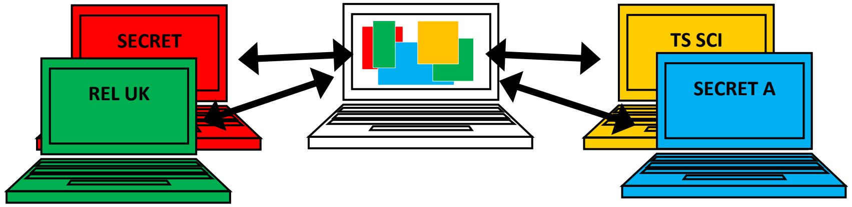
- Data Access Solution
- Can see data from different classifications
- Hypervisors, remote desktops
- Interact with remote systems / data



Cannot move data with guard

Multilevel Security

- Store multiple classification levels on one system
- Data retains original classification level
- Logical separation of multilevel data
- Not all users are cleared for all data
- Access only data at your clearance level or below



Basic Principles - MLS

- Multilevel security is a form of security that allows a system to do the following:
  - simultaneously import, export, process, and store data at different security classifications, compartments, and releasabilities
  - simultaneously be connected to multiple networks that have different security classifications, compartments, and releasabilities
  - allow multiple users with difference clearances to use the system simultaneously
  - allow users to access the system without being cleared for all the data being stored on or processed by the system

- *NOTE:*
  - *This is NOT done by isolating the different classifications, such as with virtual machines*
  - *Directories, desktops, network connections, and other resources are shared, not necessarily assigned a single classification or polyinstantiated*

Mandatory Access Control

- System defines and enforces a system-wide MLS policy
  - as set up by system administrators
  - file owner cannot change MAC settings without being authorized
  - file owner cannot grant access to other users
- System enforces MLS access
  - based on level of security, represented by a sensitivity label (SL)
    - every subject has an SL
    - every object has an SL
  - system compares subject SL with object SL to determine access
  - controls read, write, execute access
  - implements the Bell-LaPadula Model for information flow control

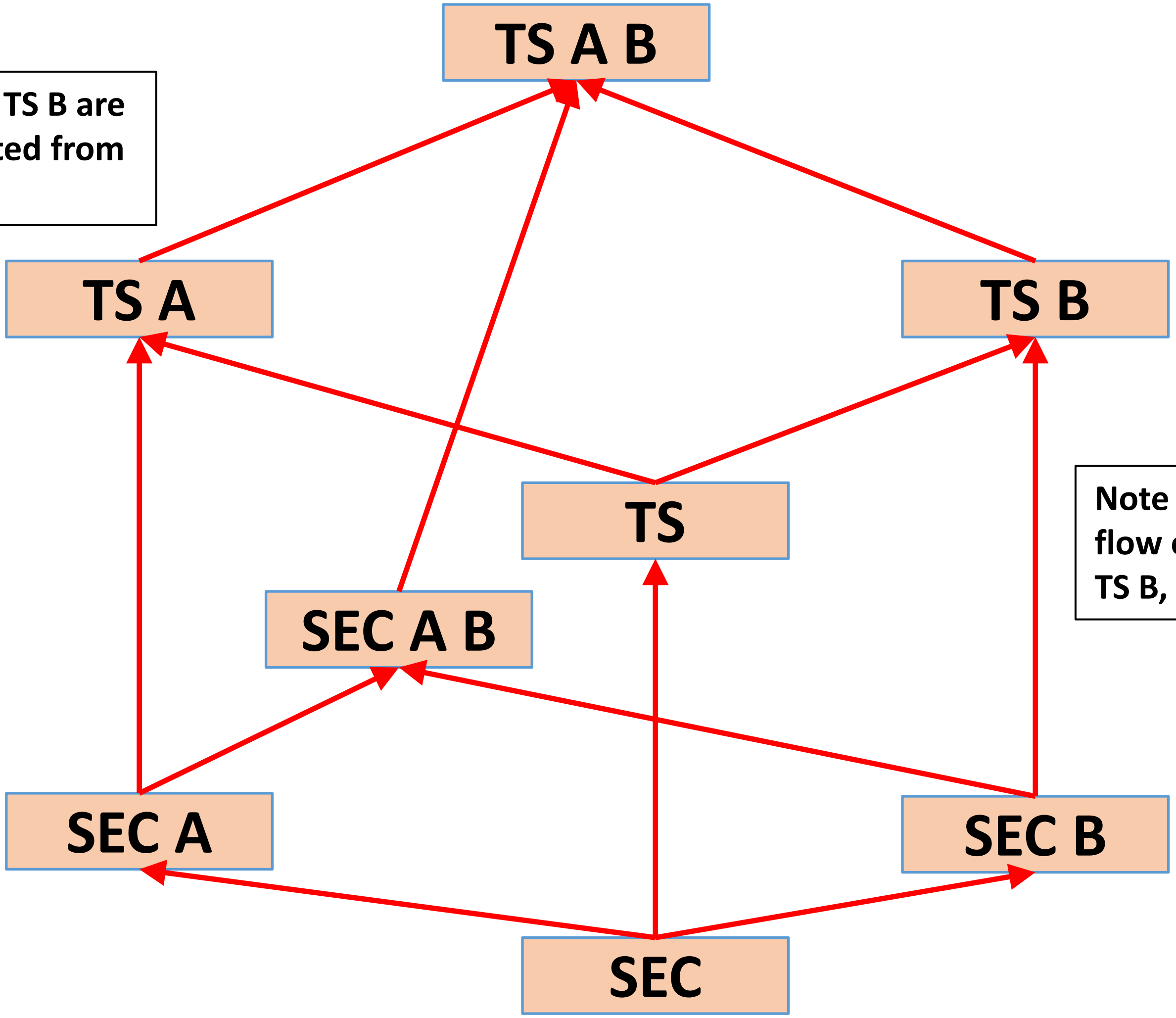
BLP Model Strong *-property

- “Read down, write equal” MAC policy
- Read examples:
  - SEC A B can read UNC, SEC, SEC A, SEC B, SEC A B
  - TS A can read UNC, SEC, SEC A, TS, TS A
  - TS A cannot read SEC A B, TS A B
- Write examples:
  - SEC A can only write to SEC A, not to SEC or TS
  - TS A B can only write to TS A B, not to TS or SEC A B
- Policies are transitive

$$X \rightarrow Y \quad \& \quad Y \rightarrow Z \quad \Rightarrow \quad X \rightarrow Z$$

- This results in both one-way data flows and isolation capabilities

Information Flow in the BLP Model



Note 1: TS A and TS B are completely isolated from each other

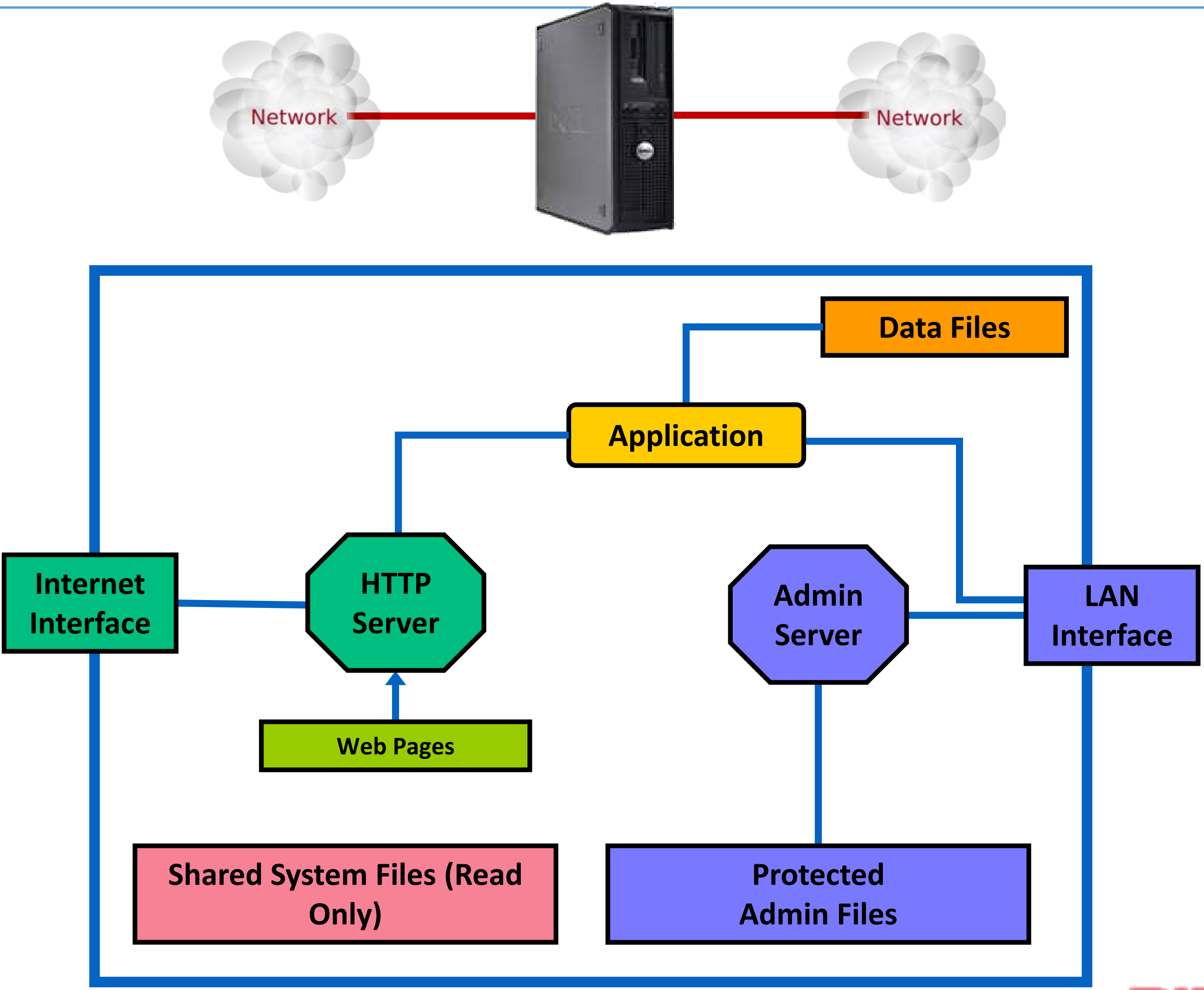
Note 2: Information can flow only from SEC B to TS B, not the other way

Multilevel vs. Single-level Applications and Processes

- A multilevel application
  - is aware of its own label
  - accesses data at multiple levels
  - may change its own label or the label of data
  - can override some MAC controls (has privilege)
  - must undergo significant testing and IA analysis

- A single-level application
  - thinks it is running on a non-MLS system
    - does not use any MLS-specific libraries
    - includes all COTS software
  - cannot override any MLS security
  - is subject to the MAC/BLP model rules
    - can read down, write equal
  - does not have any MAC override privileges
  - general requires little or no IA analysis

MLS/Compartmented Protection



PITBULL

Multiple Security Domain Architectures Comparison

CDS Features and Capabilities	Non CDS (MSL)	TRANSFER (System High)	ACCESS (MILS)	ENTERPRISE (MLS)
File transfer across security domains	X	●	X	●
Single terminal access to files on different security domains	X	X	●	●
Email transfer across security domains	X	●	X	●
Single terminal access to email on different security domains	X	X	●	●
Use local native mail client	●	X	●	●
Local multilevel applications accessing multiple security levels of data simultaneously	X	X	X	●
Remote desktop interface to multilevel data	X	X	●	●
Thin client interface to multilevel data	X	X	●	●
Web interface to multilevel data	X	●	X	●
Objects (file data, devices, network packets…) tagged/labeled at security level	X	X	X	●
Multiple security levels of data on local device, multilevel file system & multilevel database	X	X	X	●
Multilevel chat across security domains	X	●	X	●

Protection Levels

- PLs are defined in DCID 6/3
  - Director of Central Intelligence Directive 6/3: Protecting Sensitive Compartmented Information Within Information Systems (May 2000)

Protection Level	Lowest Clearance	Formal Access Approval	Need To Know
1	At Least Equal to Highest Data	All Users Have ALL	All Users Have ALL
2	At Least Equal to Highest Data	All Users Have ALL	NOT ALL Users Have ALL
3	At Least Equal to Highest Data	NOT ALL users have ALL	Not Contributing to Decision
4	Secret	Not Contributing to Decision	Not Contributing to Decision
5	Uncleared	Not Contributing to Decision	Not Contributing to Decision

Protection Levels: What They Really Mean

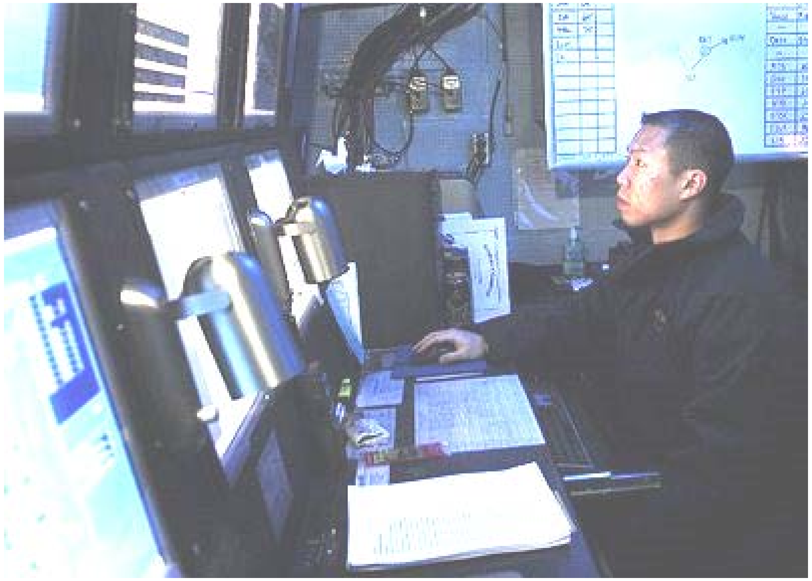
- Here is a description of what each PL means in practical terms and what kind of operating systems are needed for each level.

Protection Level	Clearance/Approval Status	Operating System
1	Everyone on the system is cleared and approved to see everything on the system	No operating system requirements for security enforcement except for login
2	Everyone on the system is cleared and approved to see everything on the system, but we want to limit access to some things by some users	Standard commercial operating systems with the ability to have users control access to their own files
3	Everyone on the system is cleared to see everything on the system, but not everyone is approved to see everything, so we must enforce a formal need-to-know policy	Standard commercial operating systems with strong configuration and extra tools to limit access to files based on a need-to-know policy
4	Not everyone is cleared to see everything on the system, but the range of clearances is typically SEC with releasabilities or TS with compartments, but sometimes UNC-SEC or SEC-TS might be approved	Operating systems that have a “mandatory access control” (MAC) capability built into them to restrict access by users to files, networks, and other objects based on the users’ clearances
5	Not everyone is cleared to see everything on the system and the system can support essentially any collection of user clearances on the system simultaneously	Operating systems that have MAC built into them plus are designed with high assurance separation and hardening

PitBull Overview

PitBull Summary

- PitBull Trusted Operating System is a commercial software product that has enhanced standard Red Hat Enterprise Linux (RHEL) into an operating system that is:
  - highly secure
  - multilevel
  - compatible
  - flexible
  - feature-rich



- PitBull is sold by General Dynamics as a separate, stand-alone product for the government and commercial markets
- PitBull has been bundled by third party companies as part of their product line

What PitBull Is Not

- PitBull isn't *encryption*
- PitBull isn't a *firewall*
  - but it does include advanced networking filtering
- PitBull isn't *intrusion detection*
  - but it *does* include tools to detect changes to files
- PitBull isn't *system access control*
  - but it *does* include enhancements to the login subsystem
- PitBull isn't *virus scanning*
- PitBull isn't *system hardening*
  - but is *does* significantly harden a system

What PitBull Is

- PitBull is a *multilevel security (MLS)* operating system *software* product
- PitBull modifies and enhances the *operating system* (kernel) so that the OS has more security features
- PitBull includes security functionality *outside the kernel* as well (e.g., enhanced login, password, integrity checking)
- PitBull includes *utilities* to configure and manage its security features

- PitBull…
  - Prevents any bug in any program from damaging the underlying system
  - Controls what network resources can be used by each program
  - Limits all user and administrator accounts
  - Enforces a security policy on a system of malicious software

PitBull History

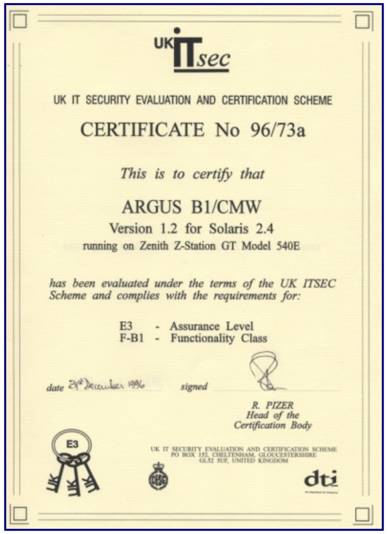
- Original technologies developed in 1988 (DIA/CMW)
  - Harris Corporation, Addamax, Argus, Innovative Security, General Dynamics
- Ported to 30+ operating systems from 1992 to 2012
- First commercial installation: Credit Suisse (1997)
- Over 50 commercial installations by 2005 in Europe, North America, Asia
- ITSEC evaluations (UK) in 1997 and 1999
- CC evaluations (Ger. BSI) in 2006 and 2007
- On Solaris 2.4 (1994) through Solaris 10
- AIX PitBull technology sold to IBM in 2005 (now Trusted AIX)
- Ported to Red Hat Enterprise Linux (RHEL) in 2012

PitBull Export Issues

- PitBull is subject to Department of Commerce EAR controls
  - Export Administration Regulation (EAR)
  - It is Export Control Classification Number (ECCN) 5D992.c mass market software
- PitBull can be exported essentially to any place that Red Hat's RHEL can be exported
  - subject to end-use and end-user as required by the EAR

Evaluations and Accreditations

- NCSC/DIA Evaluation B1/CMW
  - 1989-1993
  - completed through TRB phase
  - evaluated on SVR4 Unix
- Evaluated under ITSEC to F-B1/E3
  - 1996 and 1999; two certificates each
  - included networking and MLS GUI
  - evaluated on Sun Solaris
- Evaluated under Common Criteria
  - 2006 and 2007 LSPP/EAL4+
  - evaluated on IBM AIX
- Included in accreditations to PL4
  - base for NSA/DIA accreditations
  - four configurations of MLS desktop
  - accredited on Sun Solaris and RHEL (Linux)
  - both SABI and TSABI



UK ITsec

UK IT SECURITY EVALUATION AND CERTIFICATION SCHEME

CERTIFICATE No 96/73a

This is to certify that

ARGUS B1/CMW

Version 1.2 for Solaris 2.4

running on Zenith Z-Station GT Model 540E

has been evaluated under the terms of the UK ITSEC Scheme and complies with the requirements for:

E3 - Assurance Level
F-B1 - Functionality Class

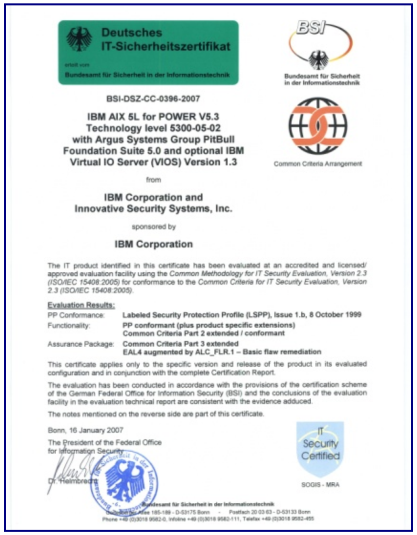
date 31st December 1996 *signed*

R. PIZER
Head of the
Certification Body

E3

UK IT SECURITY EVALUATION AND CERTIFICATION SCHEME
PO BOX 152, CHELTENHAM, GLOUCESTERSHIRE
GL52 5UF, UNITED KINGDOM

dti



Deutsches
IT-Sicherheitszertifikat

erteilt vom
Bundesamt für Sicherheit in der Informationstechnik

BSI
Bundesamt für Sicherheit
in der Informationstechnik

BSI-DSZ-CC-0396-2007

IBM AIX 5L for POWER V5.3
Technology level 5300-05-02
with Argus Systems Group PitBull
Foundation Suite 5.0 and optional IBM
Virtual IO Server (VIOS) Version 1.3

Common Criteria Arrangement

from

IBM Corporation and
Innovative Security Systems, Inc.

sponsored by

IBM Corporation

The IT product identified in this certificate has been evaluated at an accredited and licensed/ approved evaluation facility using the *Common Methodology for IT Security Evaluation, Version 2.3 (ISO/IEC 15408:2005)* for conformance to the *Common Criteria for IT Security Evaluation, Version 2.3 (ISO/IEC 15408:2005).*

Evaluation Results:

PP Conformance:	**Labeled Security Protection Profile (LSPP), Issue 1.b, 8 October 1999**
Functionality:	**PP conformant (plus product specific extensions)** **Common Criteria Part 2 extended / conformant**
Assurance Package:	**Common Criteria Part 3 extended** **EAL4 augmented by ALC_FLR.1 – Basic flaw remediation**

This certificate applies only to the specific version and release of the product in its evaluated configuration and in conjunction with the complete Certification Report.

The evaluation has been conducted in accordance with the provisions of the certification scheme of the German Federal Office for Information Security (BSI) and the conclusions of the evaluation facility in the evaluation technical report are consistent with the evidence adduced.

The notes mentioned on the reverse side are part of this certificate.

Bonn, 16 January 2007

The President of the Federal Office
for Information Security

Dr. Helmbrecht

Bundesamt für Sicherheit in der Informationstechnik

IT Security Certified

SOGIS - MRA

Bundesamt für Sicherheit in der Informationstechnik
Godesberger Allee 185-189 - D-53175 Bonn - Postfach 20 03 63 - D-53133 Bonn
Phone +49 (0)3018 9582-0, Infoline +49 (0)3018 9582-111, Telefax +49 (0)3018 9582-455

Primary Security Enhancements

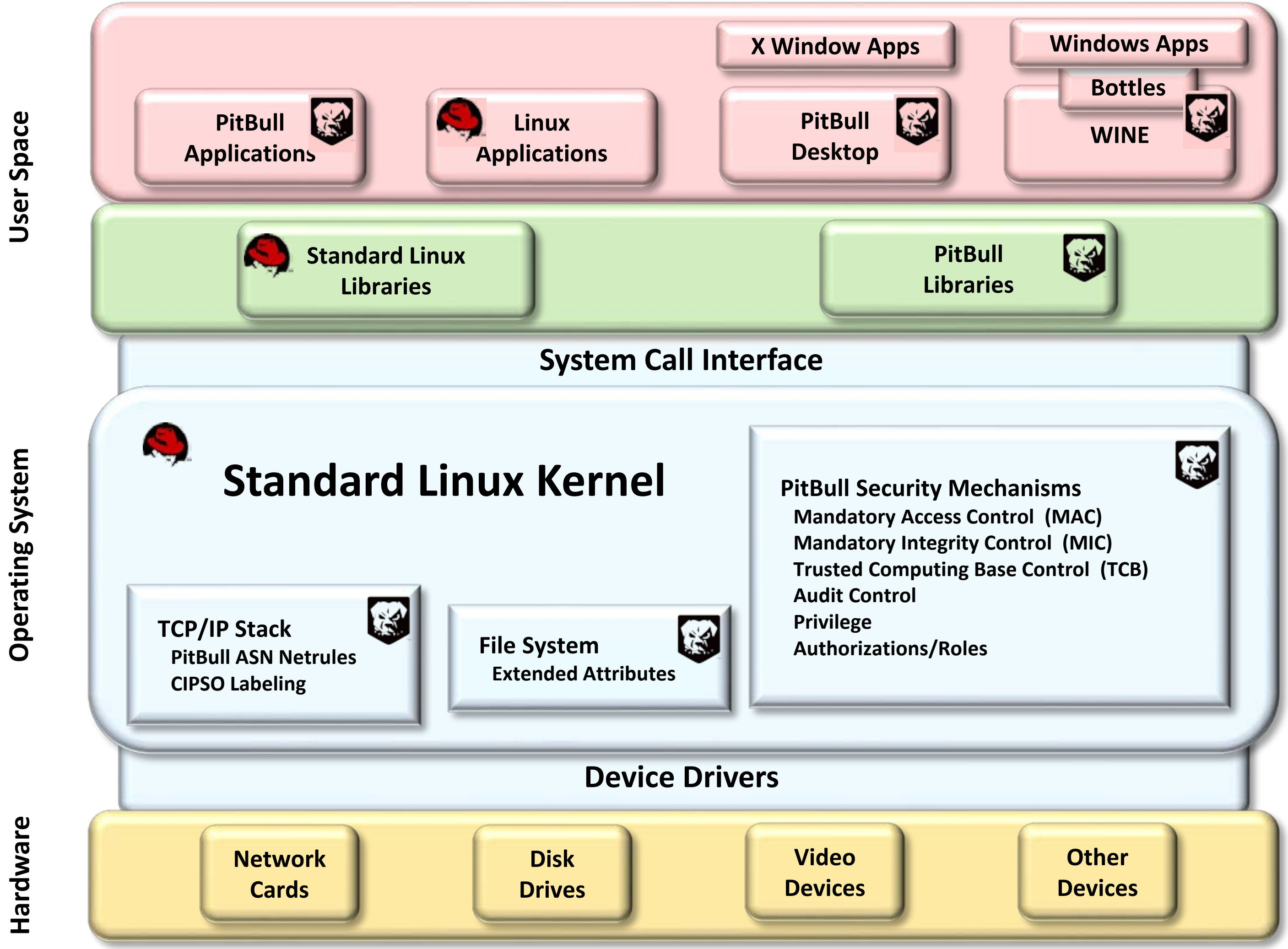
- Compartmentalization (MLS)
    - Supports 32,767 levels and 4096 compartments
    - Uses MITRE LabelEncodings file
- Privilege
    - Superuser replaced with hierarchical privilege mechanism
- Authorizations/roles
    - Site definable roles for all applications and tools
    - Enhanced functionality/security for apps based on role
- Network extensions
    - Tight integration of network MLS with OS mechanism
    - Labeled network packets and MLS NFS
    - Polyinstantiated ports

Other Security Features

- Trusted computing base (TCB) protection
  - Special controls for protecting system resources
- MLS windowing system
  - Full MLS in X window
  - MLS cut-and-paste with upgrade/downgrade
- System integrity tool
  - detects changes in file attributes and checksums
- Four-eyes login mechanism
  - Can require two administrators to be present for login
- Printer subsystem enhancements
  - MAC enforcement; header/footer & banner/trailer support

PitBull Architecture

PitBull Architecture

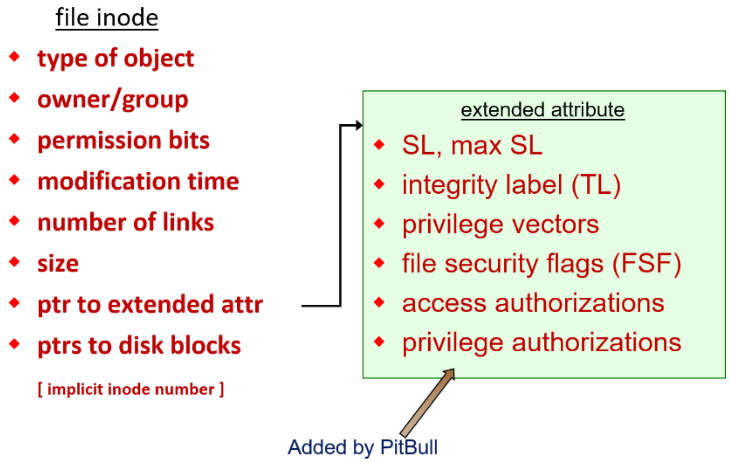


Operating System Modifications and Extensions

- The operating system has been modified
  - extended attributes on file systems, processes, IPC
  - privilege mechanism replacing all root checks
  - authorizations/roles for access to executables
  - non-overridable privilege and authorization limits
  - modes of operations: configuration and operational
  - virtualized directories based on MAC labels
  - network stack enforces MAC security and CIPSO labeling
  - polyinstantiated ports
  - audit enhancements: events, content, protections
- Not rule-based
  - support for 32K classifications and 4096 compartments

Support for Extended Security Attributes

- On a file system
  - PitBull uses extended attribute functionality of file systems
  - PitBull adds security information on a per-inode basis
  - File systems without support for extended attributes can still be supported



file inode

- **type of object**
- **owner/group**
- **permission bits**
- **modification time**
- **number of links**
- **size**
- **ptr to extended attr**
- **ptrs to disk blocks**

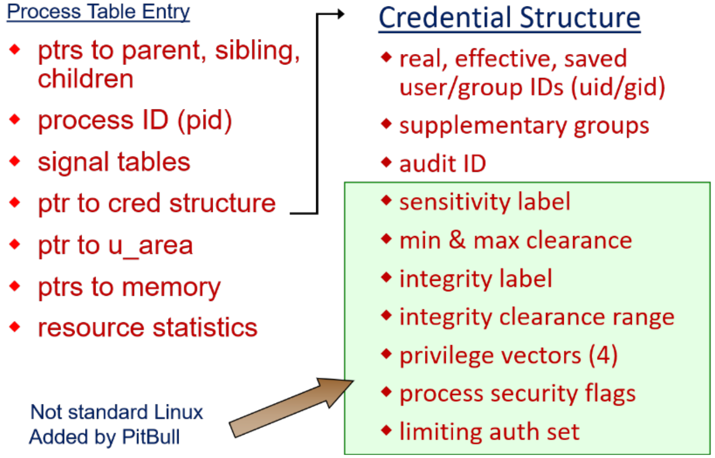
[implicit inode number]

extended attribute

- SL, max SL
- integrity label (TL)
- privilege vectors
- file security flags (FSF)
- access authorizations
- privilege authorizations

Added by PitBull

- On a process
  - The Linux per-process kernel “cred” structure has been expanded
  - Each process now has many more security attributes



Process Table Entry

- ptrs to parent, sibling, children
- process ID (pid)
- signal tables
- ptr to cred structure
- ptr to u_area
- ptrs to memory
- resource statistics

Credential Structure

- real, effective, saved user/group IDs (uid/gid)
- supplementary groups
- audit ID
- sensitivity label
- min & max clearance
- integrity label
- integrity clearance range
- privilege vectors (4)
- process security flags
- limiting auth set

Not standard Linux
Added by PitBull

Use Cases

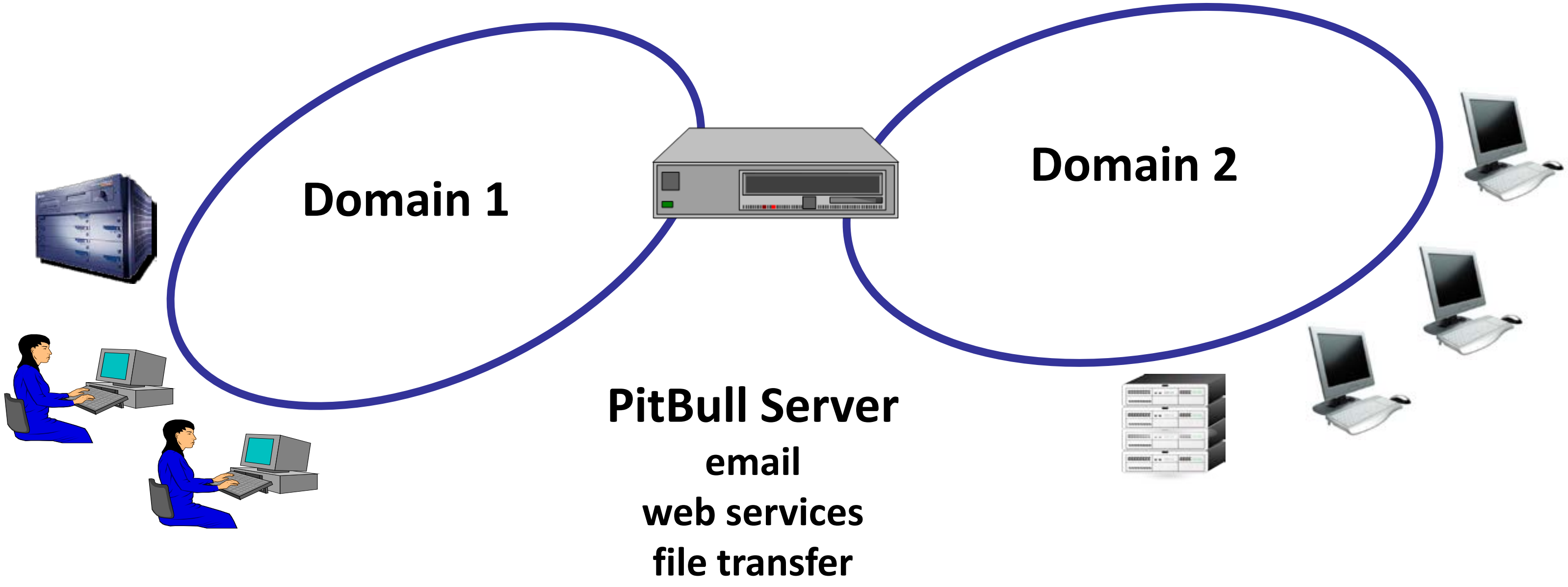
The Power of PitBull

- Two users logging in can see different views of the system depending on their accounts and the way they've logged in
- The system can safely store and process data that can never be accessed by or shared with some users (or apps)
- The system can guarantee that users (and apps) on the system can only get access to specific system and network resources
- Multiple instances of commercial software can be run simultaneously from a single installation even if the software wasn't designed to do so
- Administrators have total control over how all users and apps can communicate and share data through all mechanisms
- Multiple data flows can be forced through a preselected sequence of processes and applications.

- All of these hold even when running untested, untrusted, potentially-compromised software

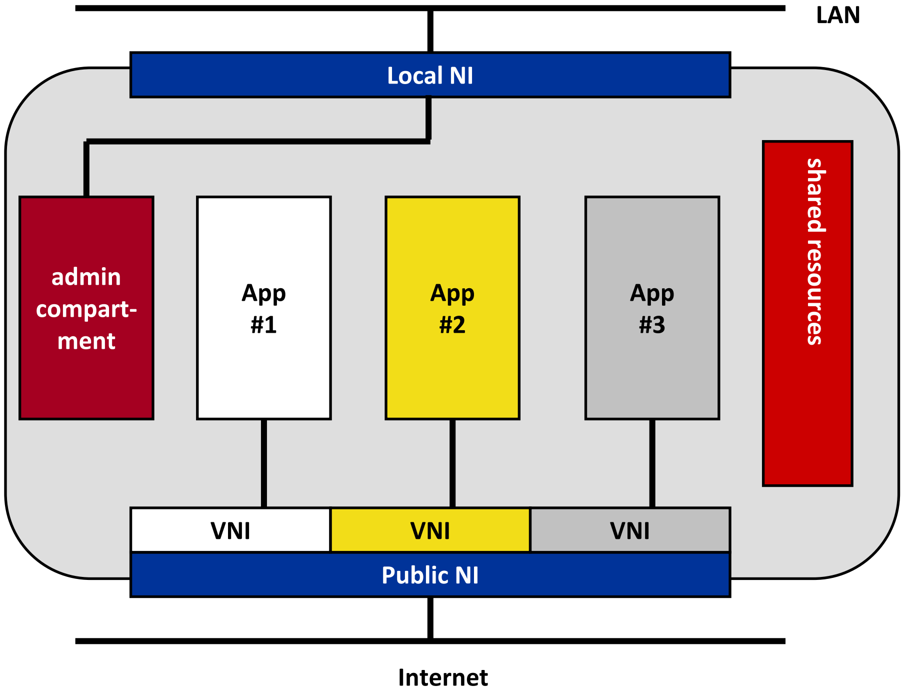
Use Case #1: Multidomain Servers

- Multiple networks can connect to a single server
- Domains can be isolated or hierarchical
- No "leakage" between domains
- Same file names / URLs can resolve to different files
  - using partitioned directories



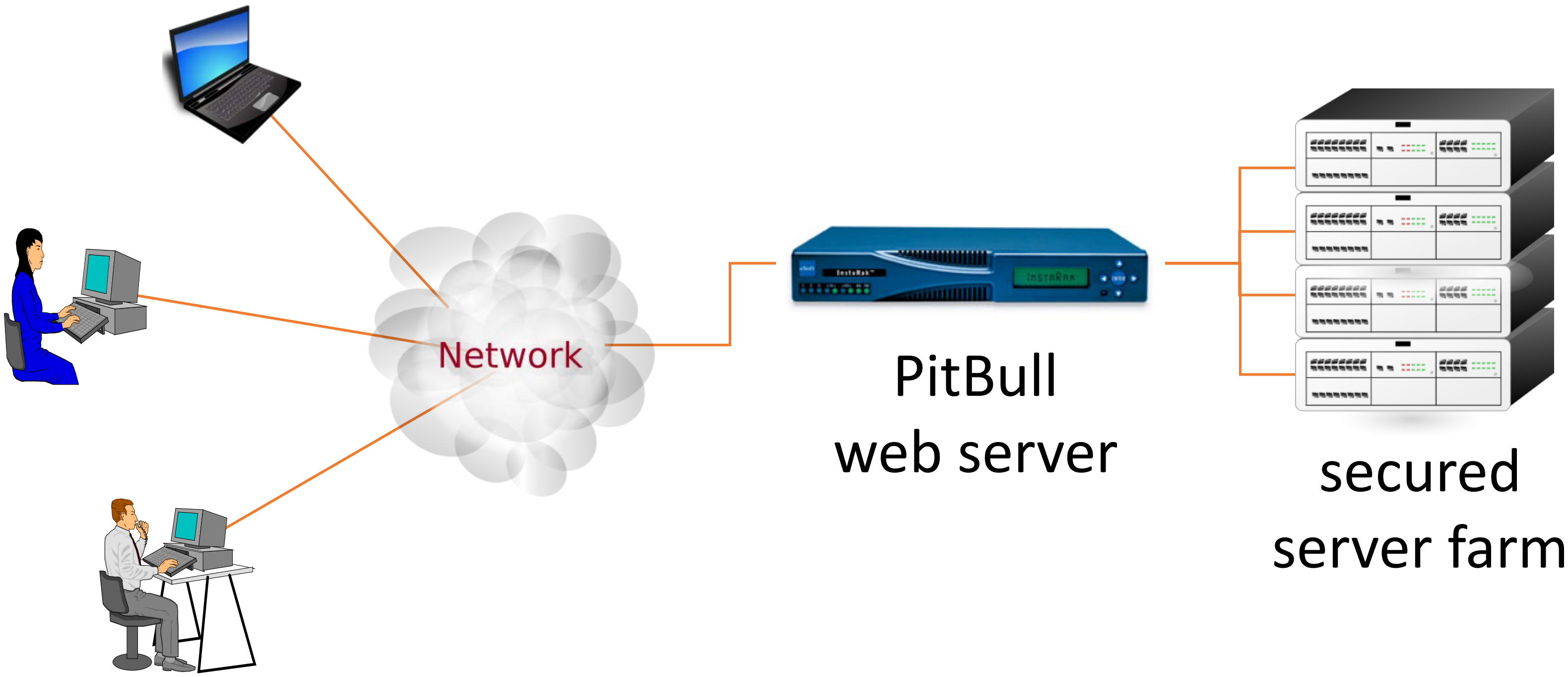
Use Case #2: Polyinstantiated App Servers

- Multiple instantiations of a single installed app
- App files are part of one administrative file system
- No danger of cross attack or data compromise



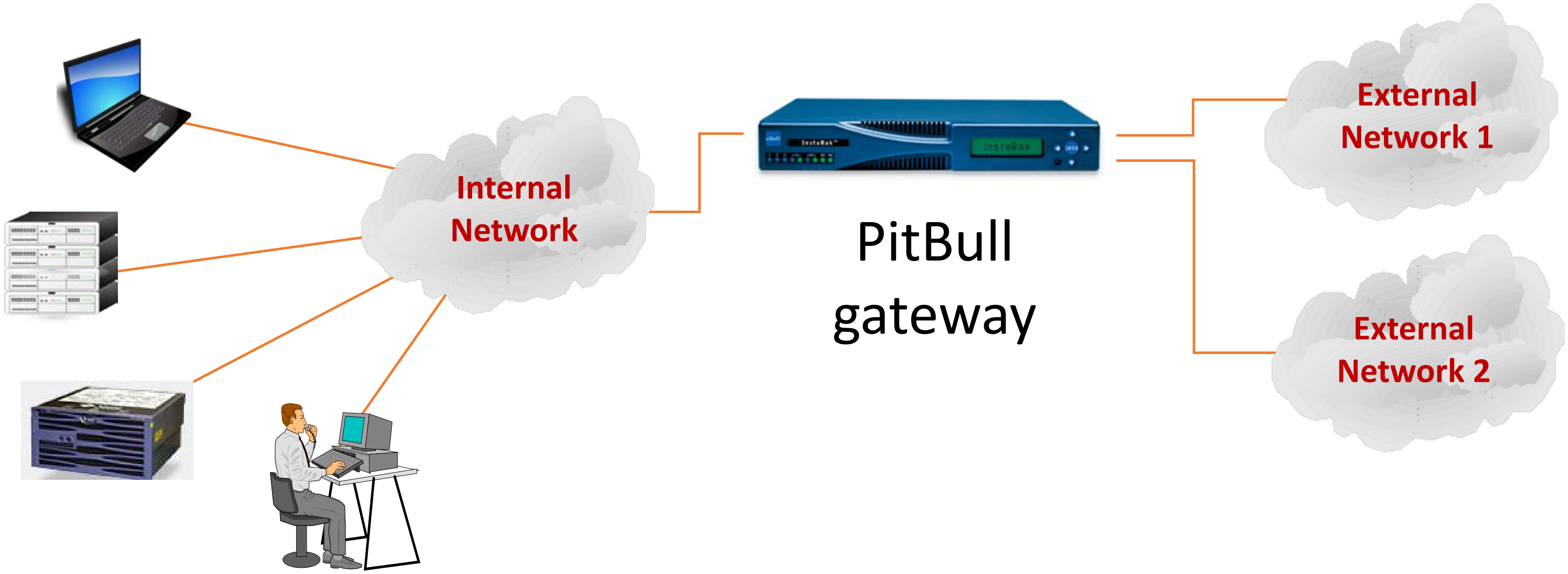
Use Case #3: High Security Web Servers

- Web apps on server are isolated from each other
- Front end / back end are strongly separated
- Strong protection against sophisticated attacks



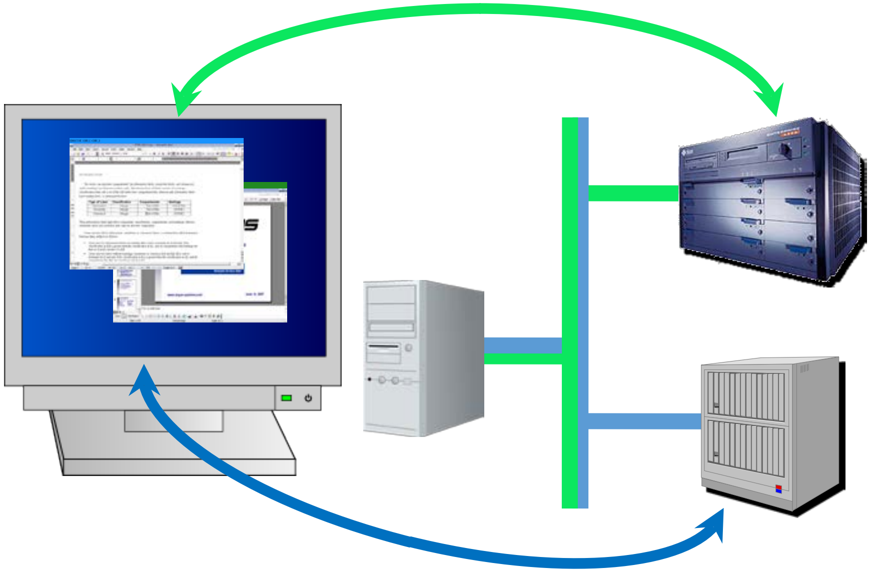
Use Case #4: Targeted Partner Gateways

- Highly secured external-internal connectivity
- Fine-grained remote access down to file level
- File/URL names resolve based on external network
- Isolation or hierarchy relationships possible on remote hosts / networks



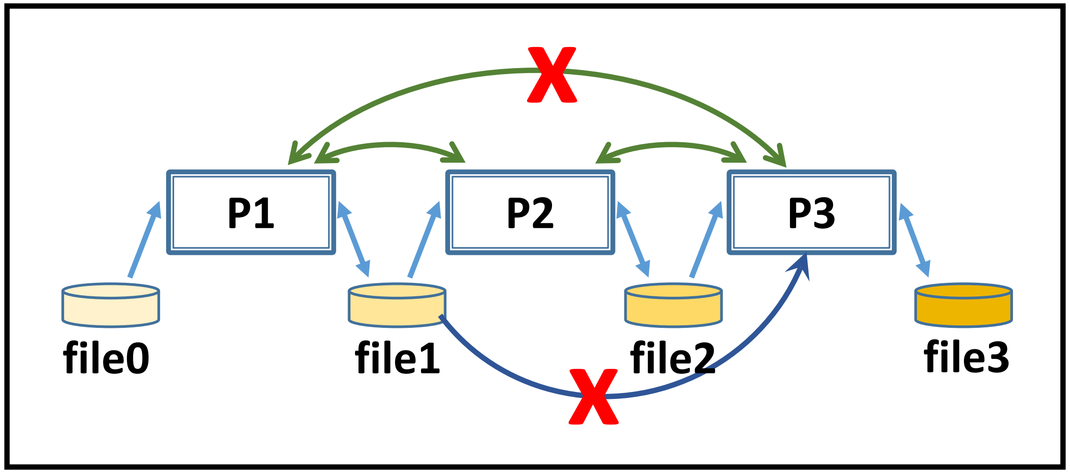
Use Case #5: Multihomed Desktops and Servers

- A desktop or server system can connect to multiple networks
- Apps run associated with one network
  - app’s SL determines network access
- No chance of internal or network-level data leakage

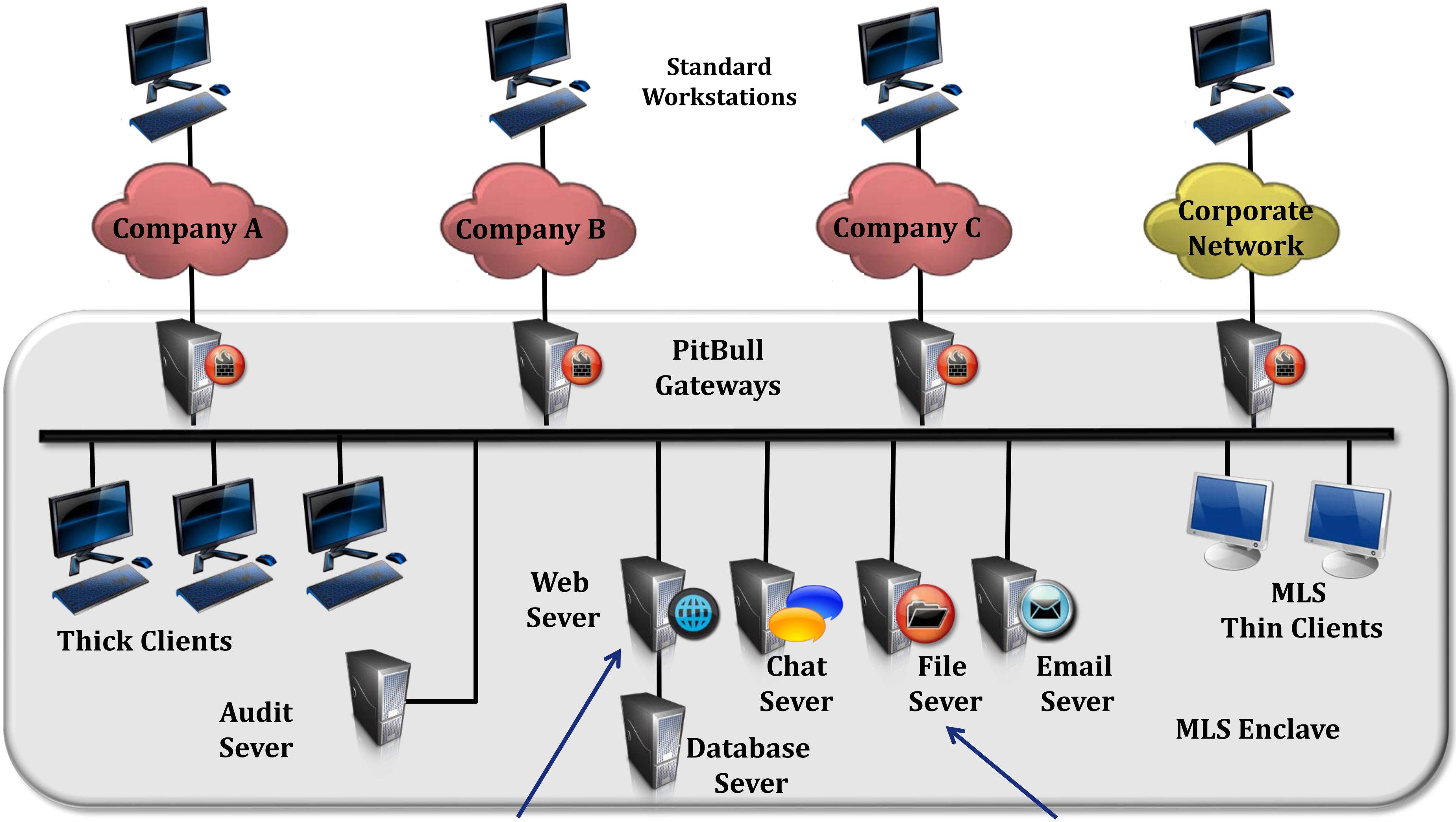


Use Case #6: Assured Pipelines (APL)

- An assured pipeline is a sequence of steps in processing data that can be assured to always
  - happen in a specified order and
  - never be bypassed
- Steps can include logging, transformations, replacements, analyses, expansions, mergings, etc.



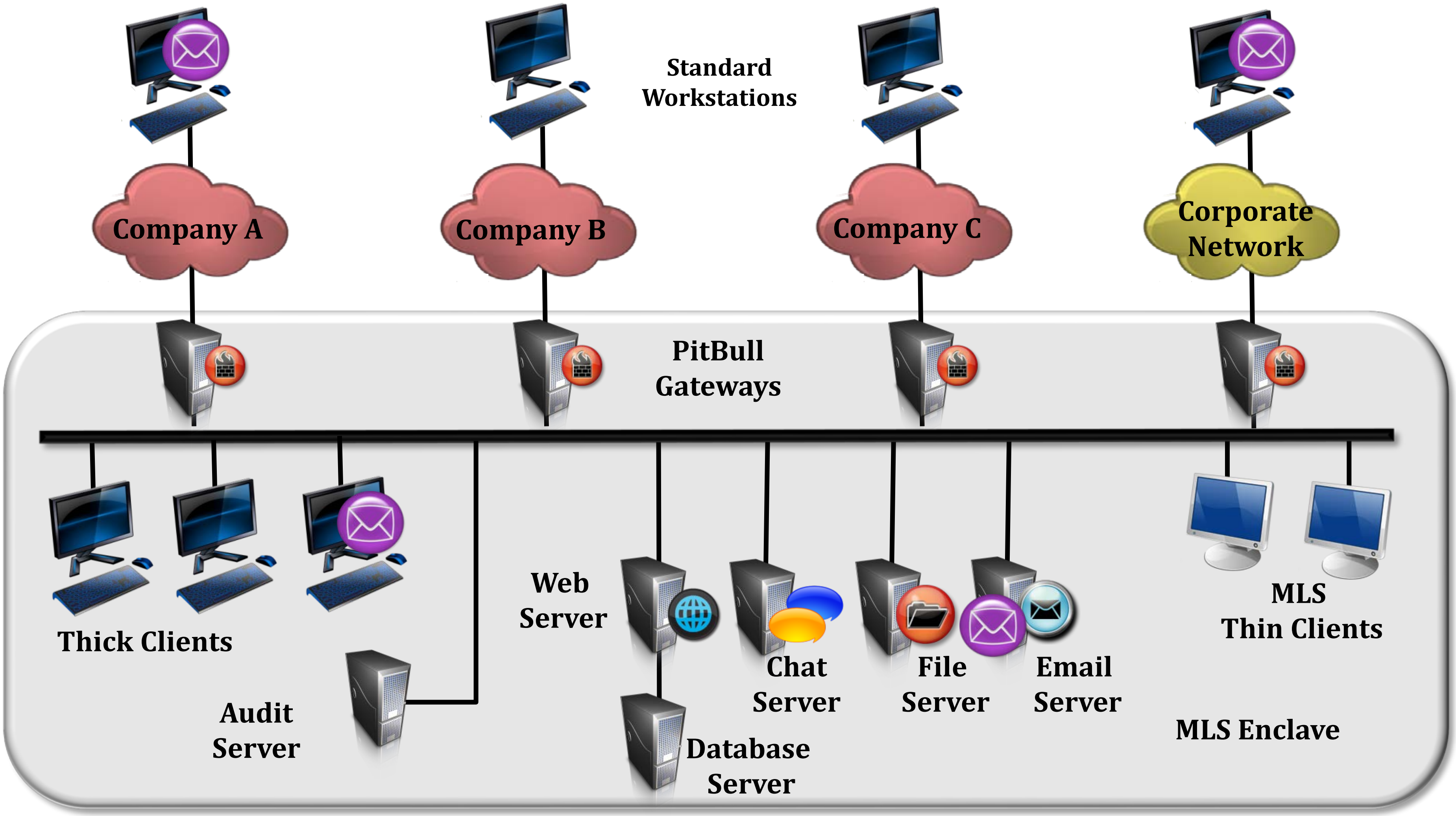
Use Case 7: MLS Enterprise Architecture



MLS website(s)

Files stored with MLS controls and access

MLS Email Between Compartments



PitBull Capabilities

PitBull Security Features and Mechanisms

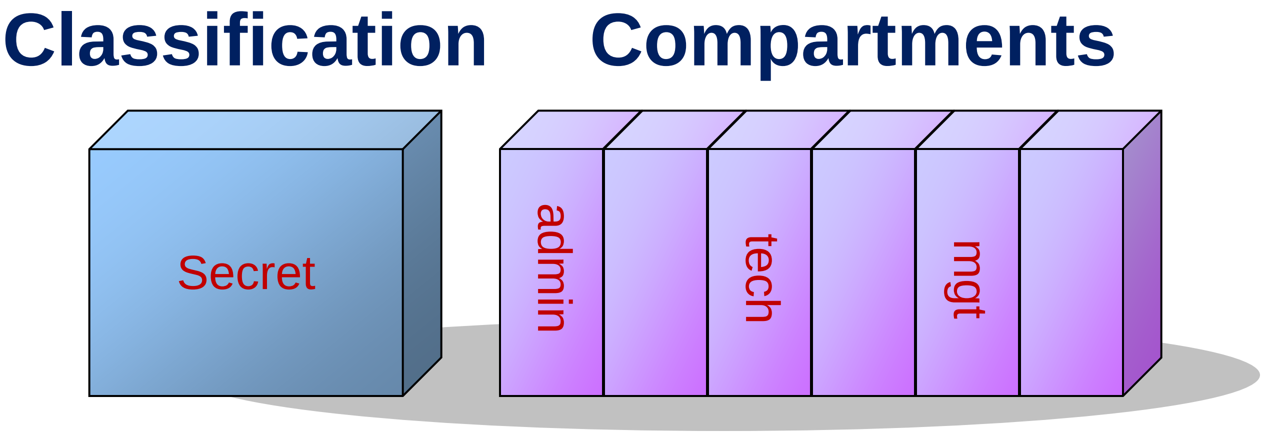
- Mandatory Access Control (MAC)
  - directory types
- Mandatory Integrity Control (MIC)
- Trusted Computing Base (TCB)
- Privileges (superuser replacement)
- Authorizations (roles)
- Foureyes (2-person login, authorization control)
- Integrity checking
- MLS Networking
- Audit enhancements

Mandatory Access Control

- System defines and enforces a system-wide policy
  - as set up by system administrators
  - file owner cannot change MAC settings without being authorized
  - file owner cannot grant access to other users
- System enforces access
  - based on level of security, represented by a sensitivity label (SL)
    - every subject has an SL
    - every object has an SL
  - system compares subject SL with object SL to determine access
  - controls read, write, execute access
  - implements the Bell-LaPadula Model for information flow control
    - read-down, write-equal

Sensitivity Labels Structure

- Classification
  - indicates level of security
  - maximum of 32,767 classifications
- Compartments or categories
  - compartments restrict users to parts of the system
  - compartments separate programs, divisions, customers, releasabilities
  - compartments can be used alone or in groups
  - maximum number of compartments is 4096



MAC Enforced System-wide

- All addressable objects have an SL
  - processes
  - files
  - directories
  - devices
  - shared memory (and all IPC objects)
  - network packets
  - printers
  - window objects (atoms, windows, fonts, graphics, etc.)
- Some objects can have an optional SL range
  - processes (clearance range)
  - directories
  - devices
  - network object (interfaces, subnets, IP addresses, ports)
  - printers

Multilevel/Ranged Directories

- Directories have two SLs
  - minimum SL (MinSL)
  - maximum SL (MaxSL)
- The labels are called the directory range
- If the labels are the same, the directory is called a single-level directory
- If the labels are different, the directory is called a ranged directory or a multilevel directory
- A process can see file names in the directory if the process SL is at or above the directory minimum
- A process can create/delete files if the process SL is within the directory's SL range
- Because this represents a potential downgrade path
  - these directories should only be used by trusted, MLS applications
  - these directories must not be directly visible to users

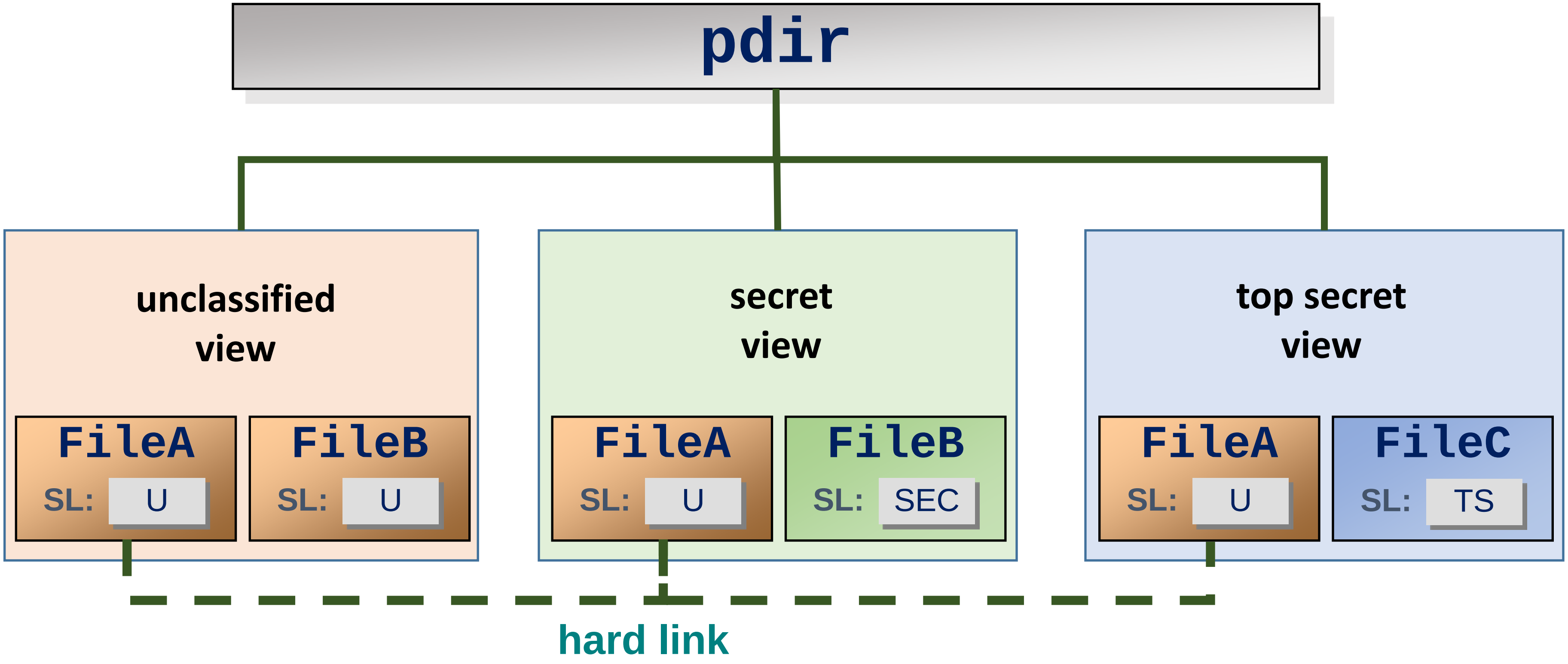
Partitioned Directories

- Partitioned directories are flagged as a special directory type by PitBull
- Partitioned directories let processes at different SLs use the same directory without clashing
- A process can see a file in the partitioned directory only if the file's SL is the same as the process's SL
- Duplicate filenames can be used as long as each file is at a different SL

```
/home/user1/dir1/fileA        (secret)
/home/user1/dir1/fileA        (unclassified)
```

- Partitioned directories cause a process to have a different view of the directory based on the SL of the process
  - if a process changes its SL, it will see a completely different set of files in a partitioned directory

Sharing Files Across Partitioned Directory



```
$ ls pdir
```

shows three different sets of files depending on the user’s SL

Comparing PitBull Directory Types

- PitBull supports three basic directory types:
  - single-level directories
  - multilevel directories (ranged directories)
  - partitioned directories

Type	MLS Capability	Read Down?	Read Up?	User Secure?
Single-level	All files are at the same SL	YES	NO	YES
Multilevel	All files are within range of SLs	YES	YES (file names)	NO
Partitioned	Files at any SL	NO (unless pdlinked)	NO	YES

Integrity Concept

- Some processes and files on the system can be “trusted” more than others
- A high-integrity process should not be “corrupted” by data from a low-integrity file
- A high-integrity file should not be corrupted by data from a low-integrity process

- This is the basis of the Biba integrity model

Mandatory Integrity Control (MIC)

- All processes and all file system objects also have an integrity label (TL)
- Controls read, write, execute access
  - default set for write only
- System enforces access
  - as set up by system administrators
- File owner cannot change MIC settings unless authorized
- File owner cannot necessarily grant access to others unless authorized

Note: the MAC and MIC policies are completely orthogonal

- even if a process has MAC write, it may not have MIC write
- even if a process has MAC read, it may not have MIC read

MAC/MIC comparison

	MAC (Bell-LaPadula)	MIC (Biba)
purpose	confidentiality	integrity
policy	read-down, write-equal	read-up, write-down
label structure	hierarchical & non-hierarchical	hierarchical only
directory types	single, multilevel, partitioned	single
clearance ranges	supported	supported
special labels	SLSL, SHSL	SLTL, SHTL, NOTL
override mechanism	priv & FSF (LEF, MAC_EXMPT)	NOTL on either file or process
kernel policy	MAC on/off	READ on/off , WRITE on/off
network support	yes	no
printing support	yes: access & header/footer	no
X Window support	yes	no

Trusted Computing Base (TCB) Goals in PitBull

- Protect OS files from modification
- Protect key directories and files from modification
- Protect dynamically linked libraries from modification and "replacement"
- Protect critical devices from access
- Implement and enforce "modes" of operation
    - operational mode
    - configuration mode

Operational and Configuration Modes

- System operates in one of two modes
  - **operational mode**
    - for day-to-day, multiuser use
  - **configuration mode**
    - for critical, security-relevant administration
- Different kernel security flags are enforced for each mode
  - both sets displayed during boot
- When in operational mode, no one, including administrators can:
  - modify kernel security flags for either mode
  - modify TCB objects
    - create
    - delete
    - modify
    - rename
    - change attributes

Note: this is orthogonal to the DAC, MAC, and MIC security policies

Least Privilege

- On any system, there are certain operations that are restricted to administrative users or services
  - use a low-numbered network port
  - ignore the permission bits on a file
  - perform a shutdown of the system
  - create a new device file
  - send a signal to a process owned by another user
- The principle of *least privilege* is that a program is given no more privilege than it needs to do its job
- Linux uses a single privilege—root—to bypass security constraints
  - if anything goes wrong, everything goes wrong
  - minor requirements (e.g., using port 80) requires major power (can erase all disks)

PitBull Privileges Overview

- PitBull has over 100 distinct privileges
- Privileges are associated with processes, not users
  - exists as a process attribute, like the process uid, and implemented as a bit set
- On fork, the child keeps all parent privileges
- On exec, a process loses all privileges
  - getting or keeping privileges on exec is possible
- Enables process to perform otherwise restricted actions. For example:
  - an unprivileged process cannot bind to a privileged port
  - a process with **PV_ASN_PORT** can bind to a privileged port

Privilege Hierarchy Example

A process with PV_DAC…

… is treated as if it had all these

- **PV_DAC**
  - **PV_DAC_R**
  - **PV_DAC_W**
  - **PV_DAC_X**
  - **PV_DAC_O**
    - **PV_DAC_SIG**
  - **PV_DAC_UID**
  - **PV_DAC_GID**

Administrator Controls Over Privileges

- Administrators determine what privileges a process can use
- Administrators can control which, if any, privileges a programmer or process can pass across an exec
    - system security officers, not programmers, have the final say on how privileges are assigned and used
- Administrators can permanently prevent a process and its children from ever inheriting privileges from any source
    - this restriction cannot be overridden by ANY PitBull mechanism
- Administrators can enable privileges for programs not written for PitBull systems (and thus will not internally enable privileges)
    - 3rd party software is completely supported within the PitBull security model

Authorizations (Roles)

- An authorization is an attribute that can be given to an account to grant access to programs
- Authorizations define a set of functions a user is allowed to do
- Users may have multiple authorizations
- Example:
  - network commands require user to have NETCONFIG authorization
- Site can add custom authorizations

Authorization Hierarchy

- Lower-level authorizations can be grouped under a single higher-level authorization
- Authorizations can be in multiple groups
- Groups can contain both groups and individual authorizations
- Users with higher-level authorization pass authorization check for all lower-level authorizations

Authorizations

- Three kinds of control:
  - **execution access**
    - command may require user to have a certain authorization to run
  - **privilege**
    - users with authorization may have more privileges for certain commands
  - **functionality**
    - commands may perform differently if users have different authorizations

Authorizations

- 4 High-level authorizations define roles:
  - Information Systems Security Officer (**ISSO**)
    - establishes and maintains security policy
  - System Administrator (**SA**)
    - creates user accounts, groups, etc.
    - installs software packages
  - System Operator (**SO**)
    - archives file system
    - manages line printer
    - shuts down system
  - Authorization Manager (**AUTH**)
    - manages authorization subsystem

- These can be changed, expanded, split, and combined by the site security administrators

foureyes

- Site can restrict access by users with specific authorizations
- A database of restricted authorizations can be created
  - for each restricted authorization, one or more enabling authorizations are specified
- Enforcement:
  - a user with a restricted authorization attempts to log in
  - after the user enters his account name and password, the system prompts for a second user/password
  - the second account must have at least one of the enabling authorizations to allow the original user to get access to a session on the computer
- Examples
  - ISSO:ISSO
  - SA:ISSO,SA
  - ISSO1:ISSO2
  - HELPDESK:ISSO,SA,SO,HELPDESK

Integrity Checking

- Detect changes to file attributes and content
  - compare current state to an earlier snapshot (database)
- When to run an integrity check
  - automatically during each boot (installation default)
  - after a system crash
  - whenever a violation is suspected
  - as part of system audit
  - as part of software update / distribution
- Options when encountering a discrepancy:
  - fix object
  - update snapshot (database)
  - report and continue

Checking File Integrity

- Options when performing the integrity check
  - repair files (change files to conform to database)
  - update database (change database to conform to file)
  - update only timestamps and checksums in database
  - run in interactive mode and prompt for action (repair, update, ignore)
  - check checksums
  - prepend path name (new root directory for files)
  - specify database file
  - specify directory containing database files to run

Integrity Checking for System Management

- This functionality can be used for things other than checking for problems
  - *creating tar or cpio archives*
    - create an integrity database of the files and include it in the archive
    - run the integrity program when extracting/restoring
  - *distributing a new software package or release*
    - include an integrity database as part of the distribution
    - run the integrity program as part of the installation process
  - *making configuration changes on multiple machines*
    - define the new configuration as an integrity database
    - distribute the database and run the integrity program to update each system

Advanced Secure Networking High-level View

- ASN allows a system to control exactly what network resources each process can use
- ASN prohibits a process from usurping another network function or spoofing a network service
- ASN separates and isolates hosts and networks
- ASN controls all flow between interfaces, hosts, and internal processes
- ASN connects process and file system security with network security

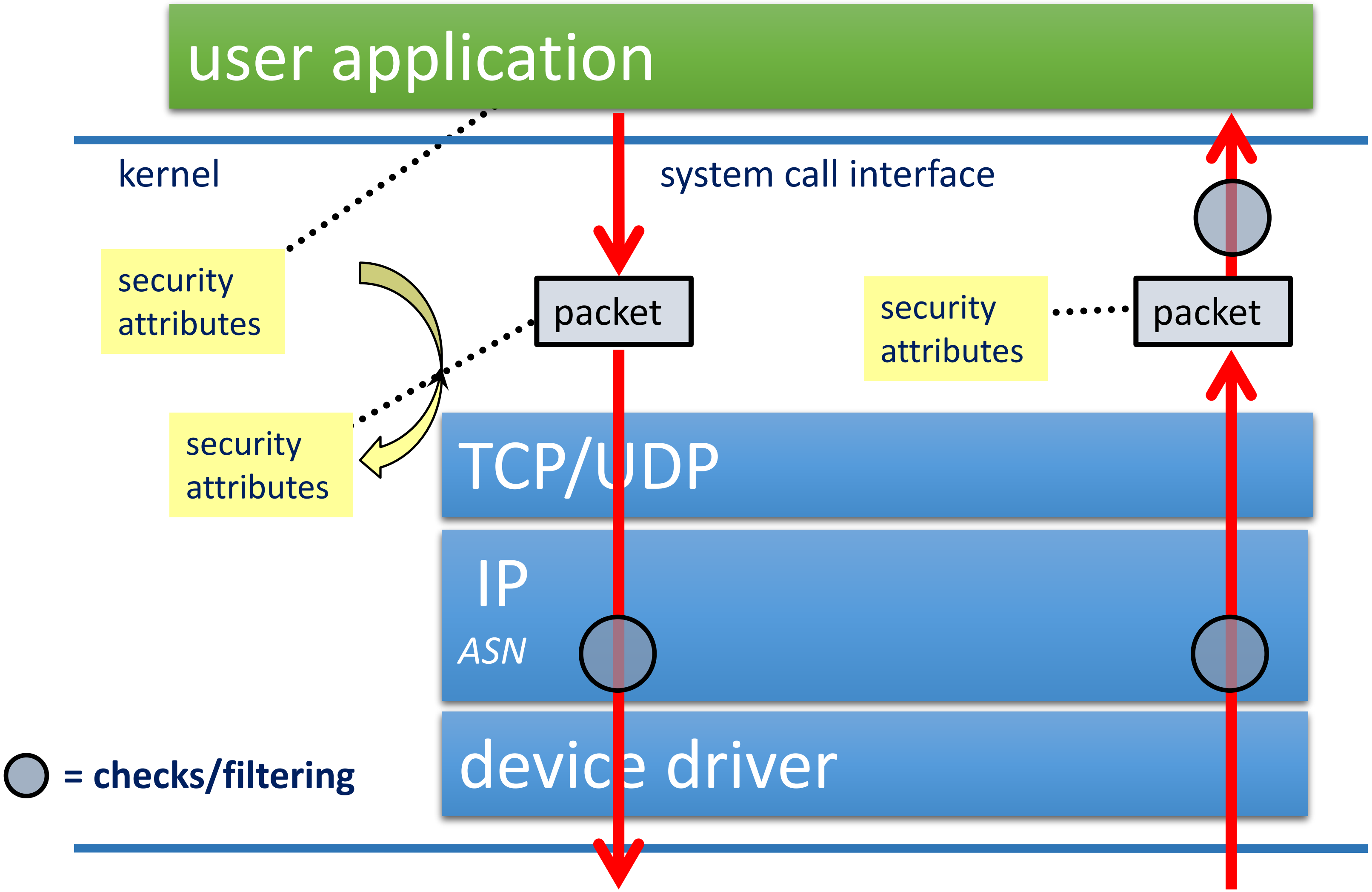
MAC and Networking

- When packets come in, they are assigned an SL
  - using the SL from the packet header (can be ignored via netrules)
  - using the SL from a matching rule
- When packets go out, they have an SL
  - SL of process that created the packet
  - privileged processes can choose another SL
  - SL can be embedded in the outgoing packet header (based on rules)
- SL ranges exist for all interfaces
  - packets dropped if not within the range of the interface
- SL ranges can exist for any combination of IP addresses, ports, port ranges, and/or protocols
  - packets are dropped if not within the specified restrictions

Advanced Secure Networking (ASN)

- Applies security attributes to packets
  - both incoming and outgoing
- Restricts network traffic
  - according to assigned attributes
- Filters traffic based on
  - interface / host / port / protocol
- Adds basic firewall functionality to protect the local machine
- Includes trusted network file system (TNFS)
  - extends security attributes across network-mounted file systems
- Allows polyinstantiation of network ports

ASN Processing



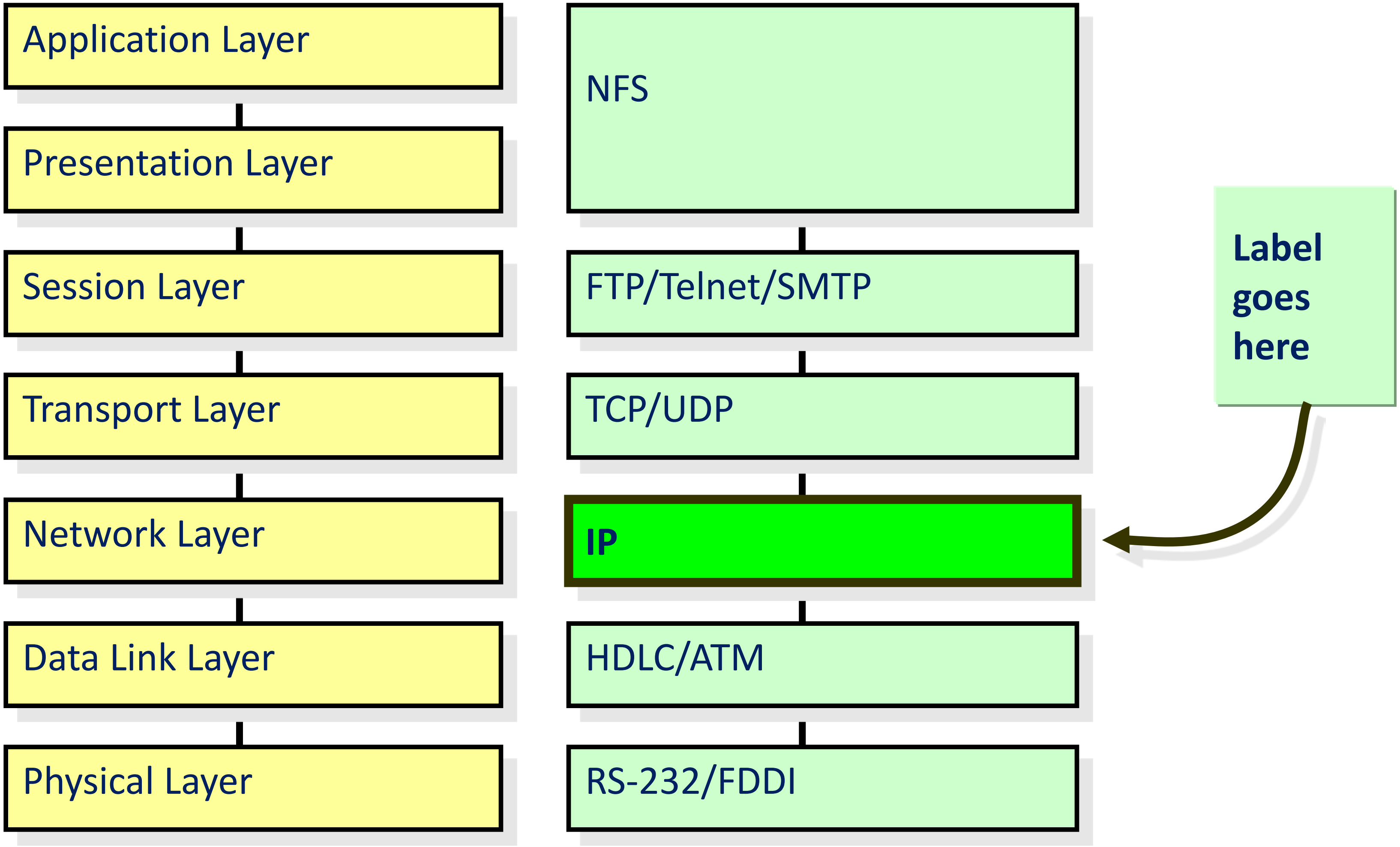
Unlabeled Networking Traffic

- Used for most Internet traffic because
  - IP label options can confuse some devices
  - IP label options are often dropped by routers
- Incoming
  - ASN assigns label based on default SL specified in the matching netrule
- Outgoing
  - ASN does not insert label into packet
- SL checking (MAC) is still done for both incoming and outgoing packets

Labeled Network Traffic

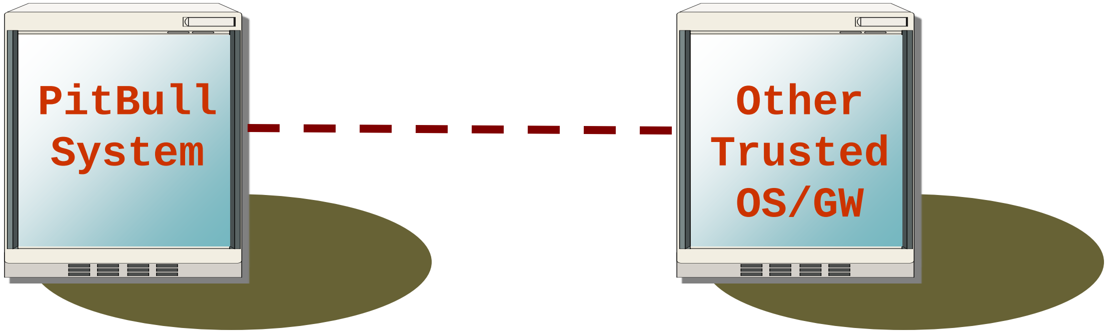
- Used for communications among trusted systems
  - sender and receiver must properly interpret label
- Incoming
  - ASN uses label from the packet's IP label option
  - should only be used if label can be trusted
    - closed network
    - encrypted packet headers
- Outgoing
  - ASN inserts label option into packet's IP header
  - uses the label of originating process

OSI Network Model



Network Label Protocols

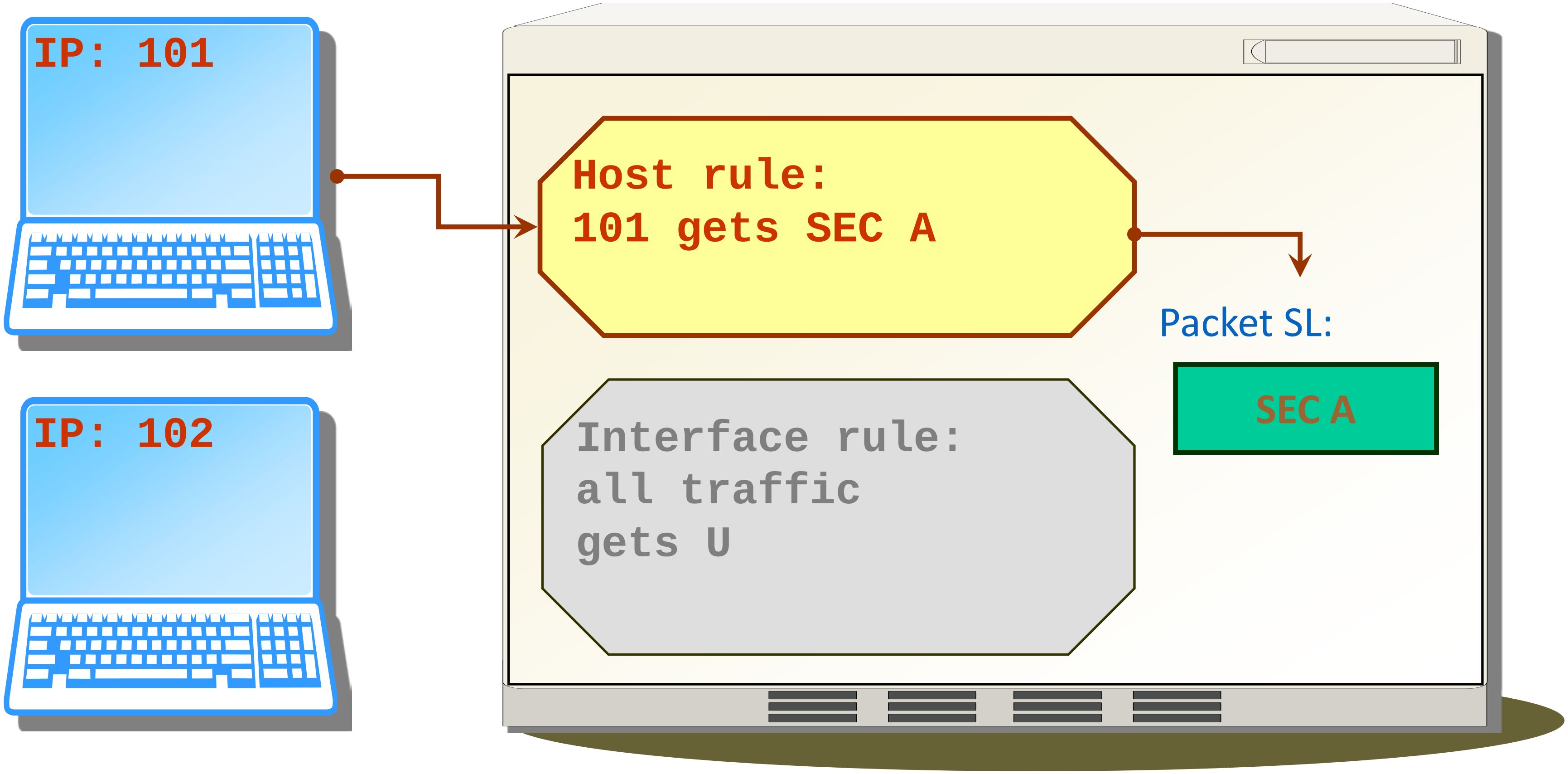
- For MLS communication between PitBull and non-PitBull machines
- CIPSO
    - Commercial/Common IP Security Option
    - also called CSL, Common Security Label
    - used in both commercial and defense networks
    - PitBull uses domain of interpretation (DOI) 0x1000
- CIPSO does not involve encryption, only passing labels in IP header options



Routing

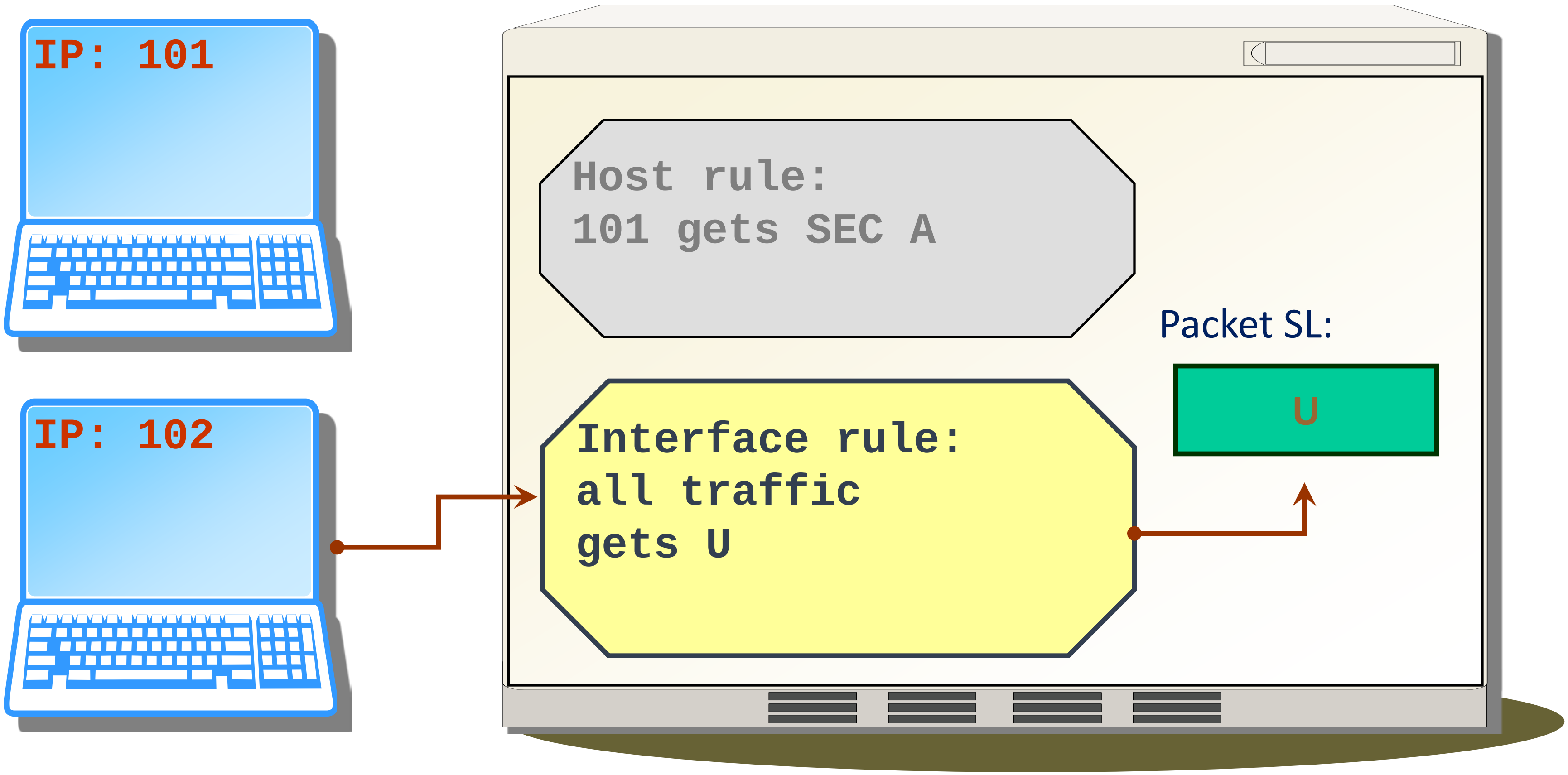
- If the system is routing packets
    - interface and incoming host rules apply to the packet on its way in
    - interface and outgoing host rules apply to the packet on its way out
- Allows total separation of networks or limited, fine-grained data flow

Host/Interface Example



**Traffic from 101 matches host rule;
bypasses interface rule.**

Host/Interface Example



IP: 101

IP: 102

Host rule:
101 gets SEC A

Interface rule:
all traffic
gets U

Packet SL:

U

**Traffic from 102 does NOT match host rule;
interface rule used.**

Trusted Network File System (TNFS)

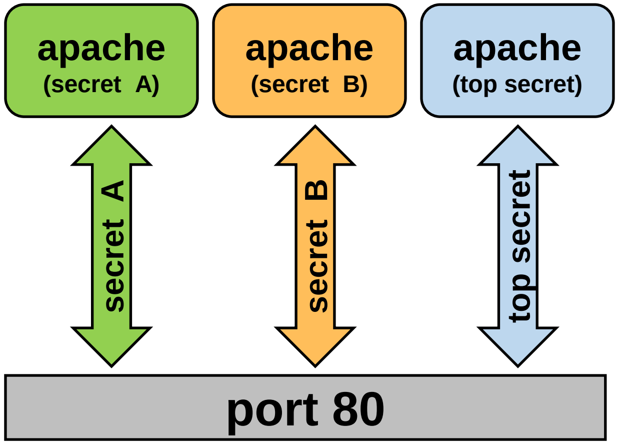
- Remote files systems can be mounted from both PitBull and non-PitBull servers
- For non-PitBull servers, NFS is used and all files on the remote file system pick up the PitBull security attributes of the mount point
- For PitBull servers, TNFS is used and all remote (server) attributes are properly used and manipulated from the client

Polyinstantiated Ports

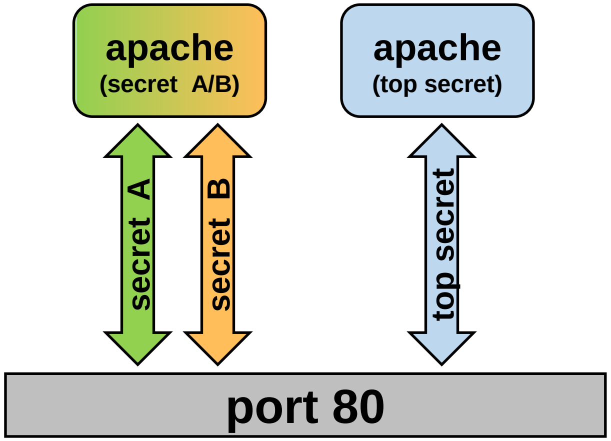
- On standard operating systems, many processes can be using a port, but only one process can listen on a port
- On a PitBull system, a process can listen on a port
    - at a single SL
    - at a range of SLs
- Multiple processes can listen on the same port as long as the associated SLs or SL ranges do not overlap
    - allows multiple copies of the same program to be running simultaneously, each handling traffic at a specific SL or range of SLs

Polyinstantiated Port Solution

- Multiple processes can be listening to the same port at the same time
  - no super server required
  - default behavior for non-PitBull programs



- A single process can be listening at multiple levels
  - any label range can be specified (uses process clearance range)
  - can have multiple super servers, each servicing a label range (e.g., secret, top secret)



PitBull Audit Modifications

- Audit trail content expanded
  - more kinds of events
  - more information for all events (SLs, privs, etc.)
- More audit configuration information
  - new command run at system boot to turn on PitBull audit rules
  - new rules added to the system audit configuration directory
- Authorizations added for audit management
  - seven separate audit authorizations added for
    - starting audit daemon, printing audit reports, getting audits stats, etc.
- New protections added for audit files
  - if a file is marked as being an audit file
    - AUDIT READ privilege required to read file
    - AUDIT WRITE privilege required to write or delete file
- Privilege required for a non-kernel process to add an audit record

General vs Targeted Access Control Mechanisms

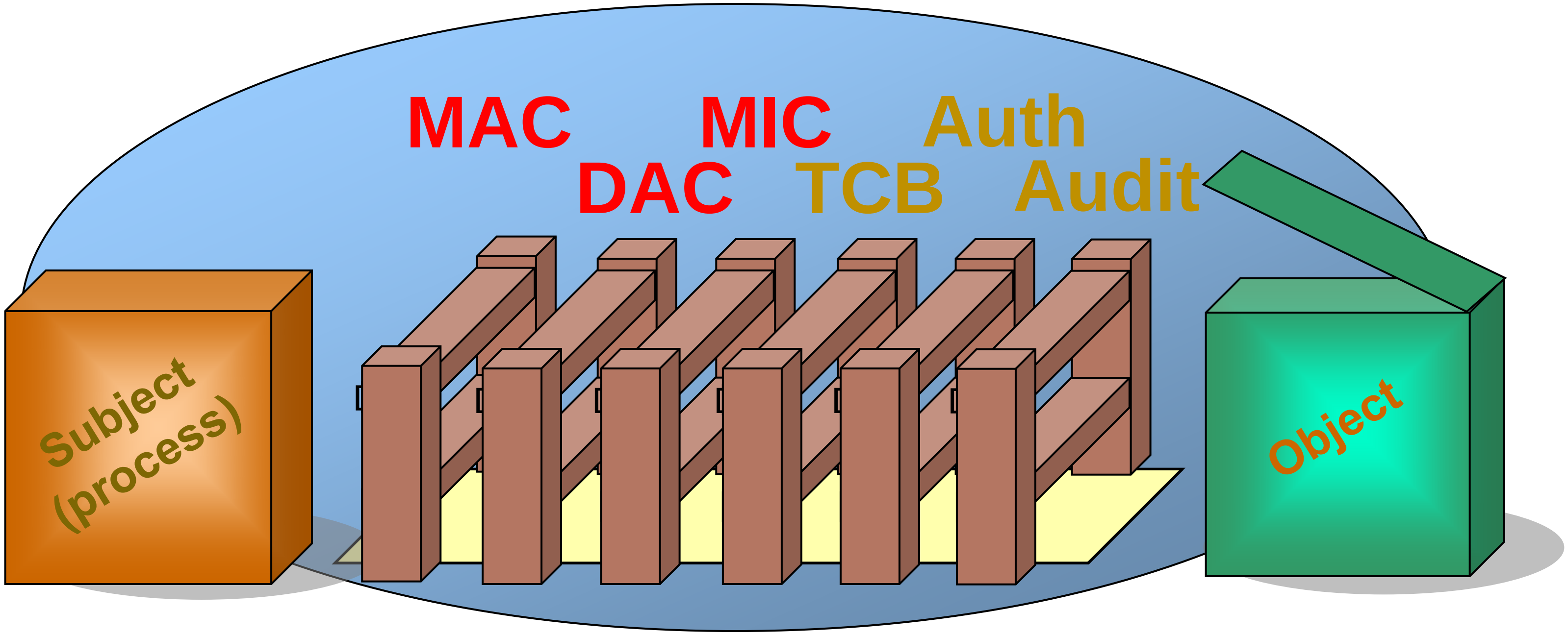
- General mechanisms—apply to all objects
  - discretionary access control
  - mandatory access control
  - mandatory integrity control

- Targeted mechanisms —apply to certain classes of objects
  - TCB protection
  - program access protection
  - audit subsystem protection

- All access control mechanisms and policies are orthogonal
  - each policy is entirely independent of the others
- All access control mechanisms have separate override mechanisms
  - a process can be given the right to override one mechanism while remaining subject to all the others

PitBull Access Control Mechanisms

- All general security hurdles must be passed to get access to an object
- All applicable targeted security hurdles must also be passed to get access to an object



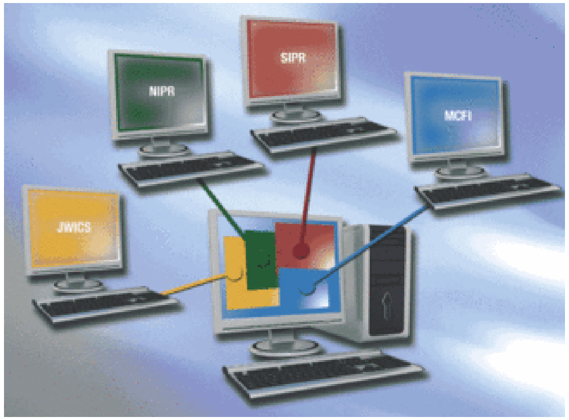
Overall PitBull Security Goal

- Overall Goal of PitBull security:
  - limit damage by rogue programs
  - limit damage by administrators and rogue users
  - control data flow
- How PitBull does that:
  - compartmentalizes system (MAC)
  - grants processes minimal privileges
  - grants users minimal authorizations
  - protects TCB

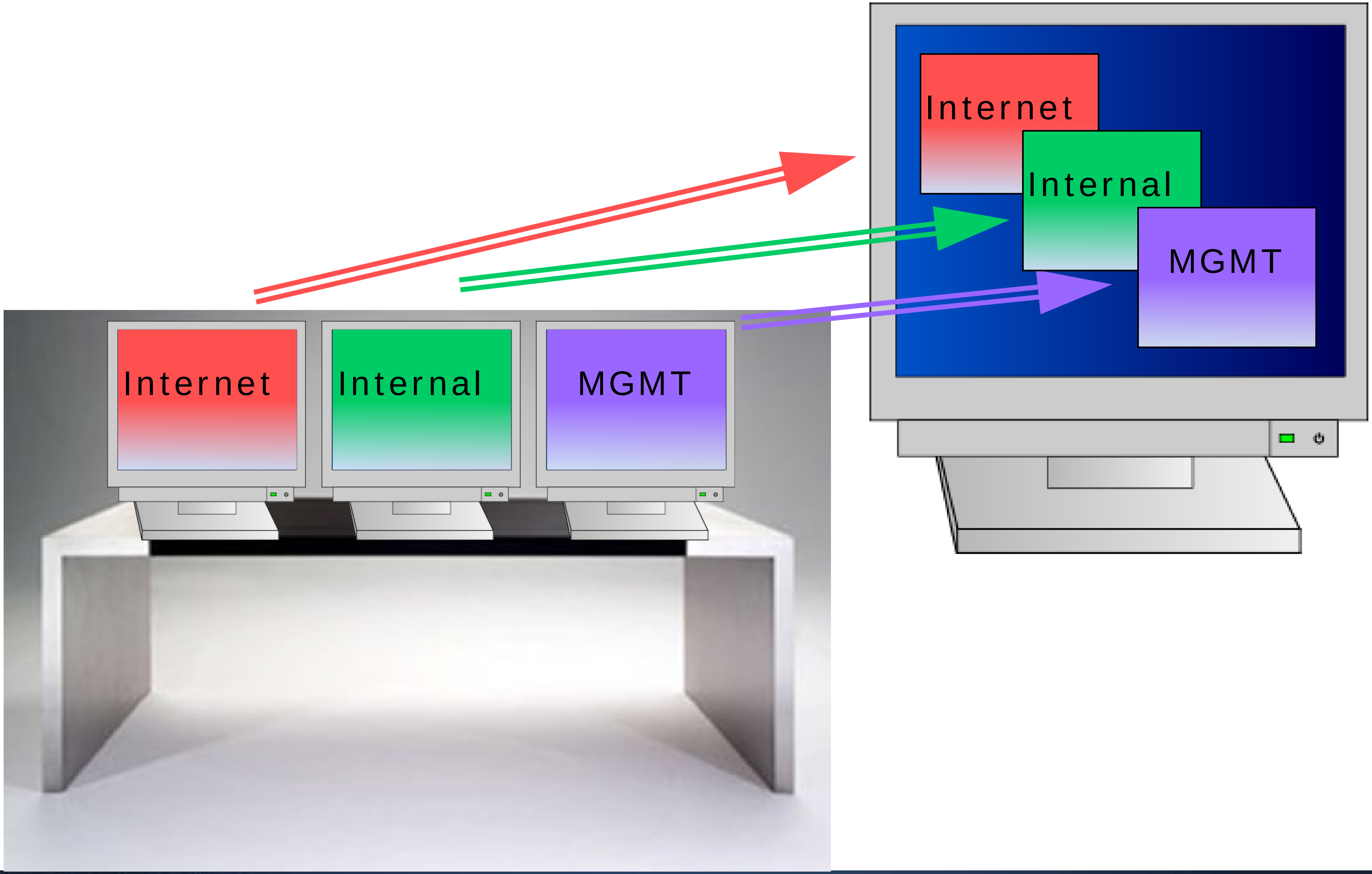
Multilevel Desktops

Multilevel Desktops

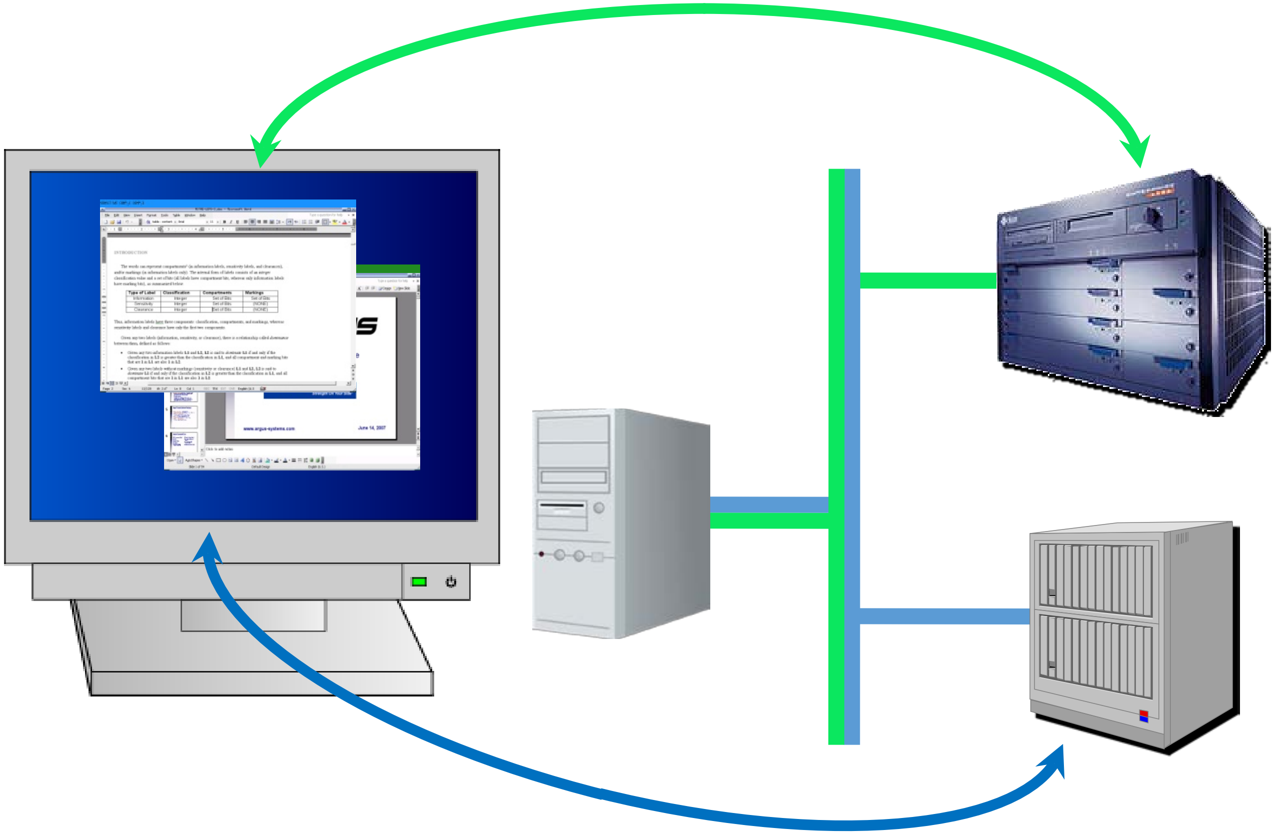
- PitBull solves the multiple box problem
- PitBull networking, OS, and X Window security create an integrated MLS desktop environment
- Cut-and-paste between windows is fully supported
  - Can be limited based on roles of users



Consolidating user hardware



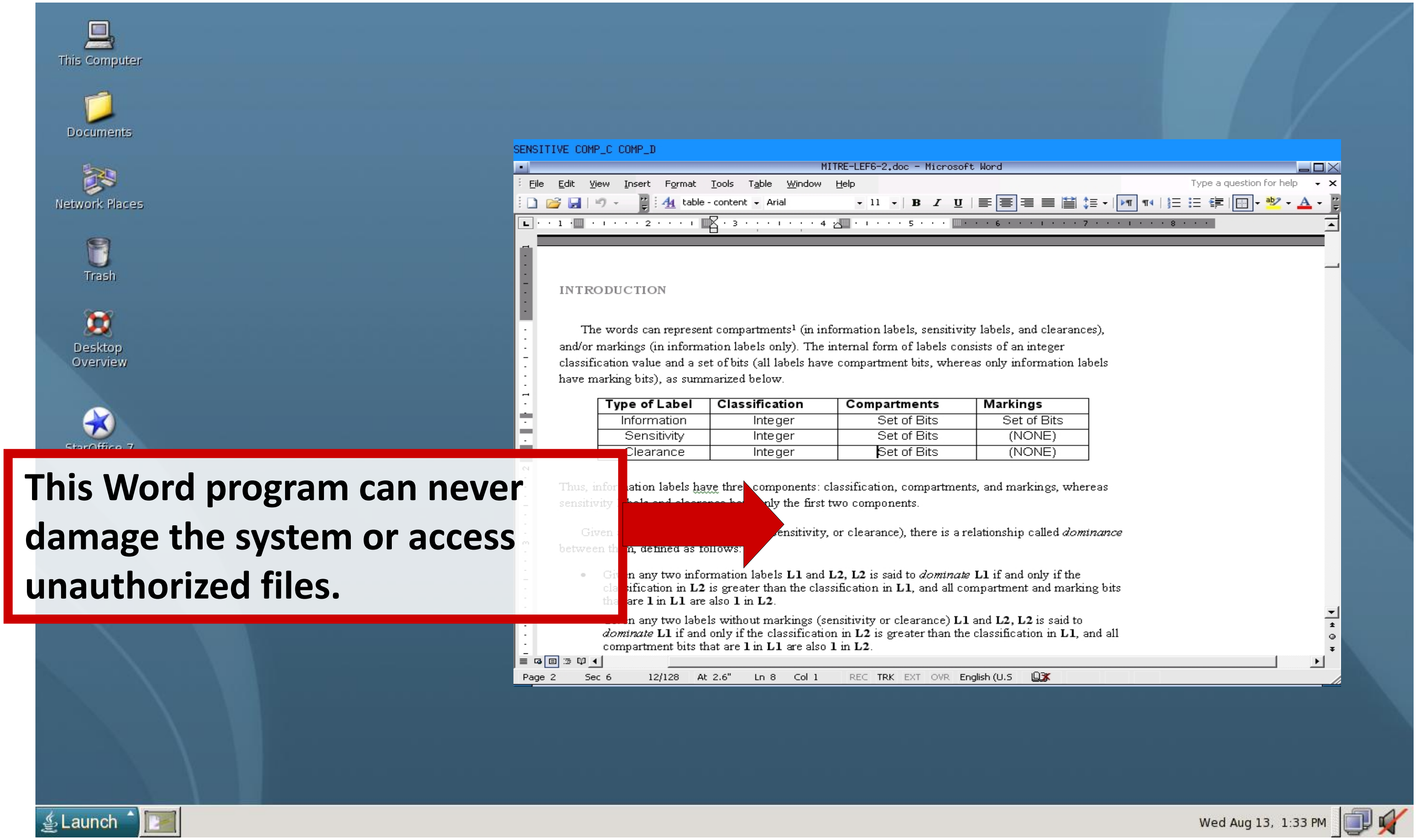
MLS From Desktop to OS to Network



Example uses of a secure desktop

- Simultaneous browser sessions securely open to internal and external web servers
- External servers accessible for copying information into sensitive documents with no danger of system attack or data leakage to outside networks
- Documents of different security levels simultaneously viewed and edited without danger of accidentally releasing restricted data
- Using the system as a personal desktop while the system is supporting administrative and infrastructure services

Example #1: Operating system protection



INTRODUCTION

The words can represent compartments[1] (in information labels, sensitivity labels, and clearances), and/or markings (in information labels only). The internal form of labels consists of an integer classification value and a set of bits (all labels have compartment bits, whereas only information labels have marking bits), as summarized below.

Type of Label	Classification	Compartments	Markings
Information	Integer	Set of Bits	Set of Bits
Sensitivity	Integer	Set of Bits	(NONE)
Clearance	Integer	Set of Bits	(NONE)

Thus, infor ation labels have thre components: classification, compartments, and markings, whereas sensitivity nly the first two components.

Given ensitivity, or clearance), there is a relationship called *dominance* between th n, defined as follows:

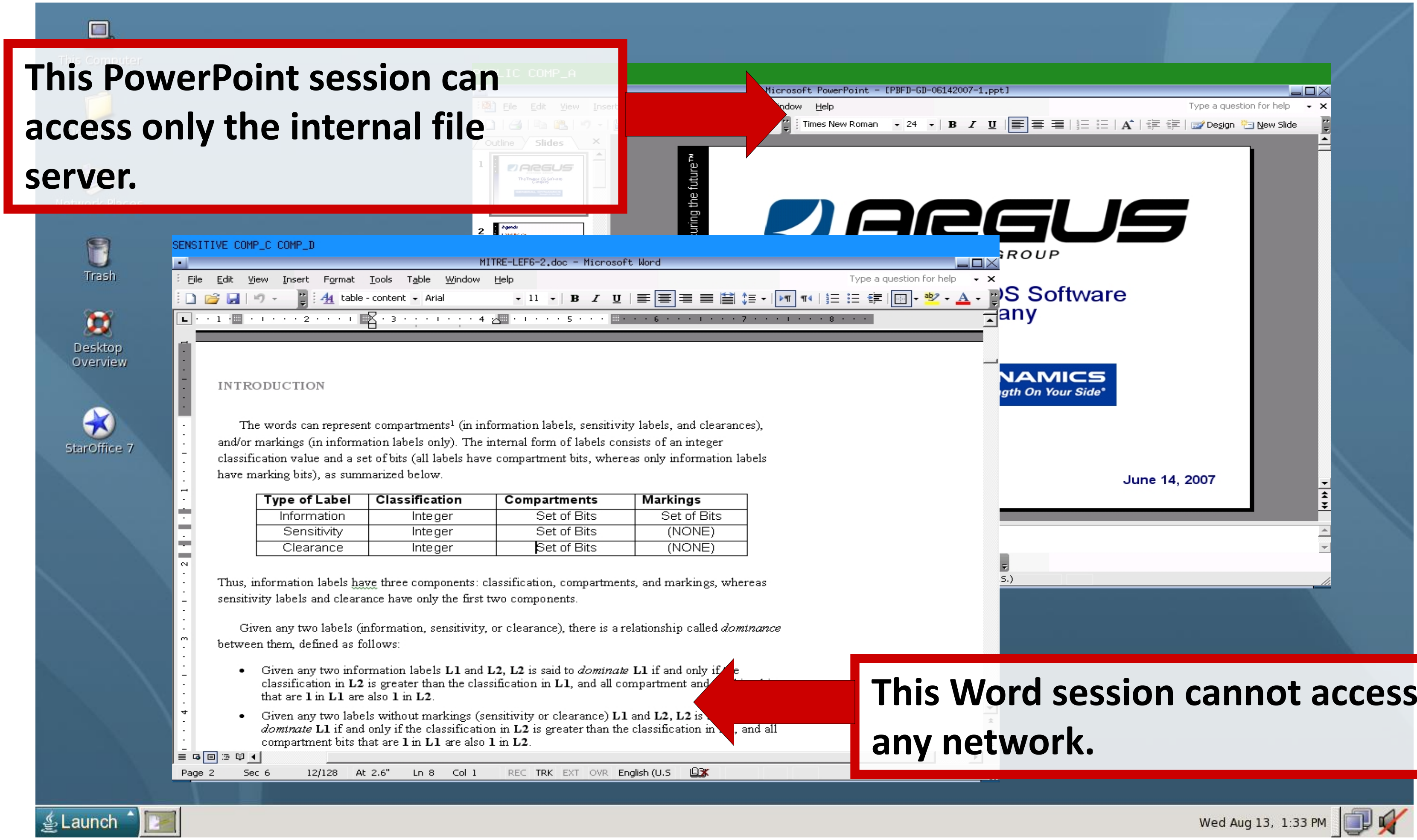
- Gi n any two information labels **L1** and **L2**, **L2** is said to *dominate* **L1** if and only if the cla ification in **L2** is greater than the classification in **L1**, and all compartment and marking bits tha are **1** in **L1** are also **1** in **L2**.
- n any two labels without markings (sensitivity or clearance) **L1** and **L2**, **L2** is said to *dominate* **L1** if and only if the classification in **L2** is greater than the classification in **L1**, and all compartment bits that are **1** in **L1** are also **1** in **L2**.

This Word program can never damage the system or access unauthorized files.

Markings are for demonstration purposes only

Example #2: Network protection

This PowerPoint session can access only the internal file server.



INTRODUCTION

The words can represent compartments[1] (in information labels, sensitivity labels, and clearances), and/or markings (in information labels only). The internal form of labels consists of an integer classification value and a set of bits (all labels have compartment bits, whereas only information labels have marking bits), as summarized below.

Type of Label	Classification	Compartments	Markings
Information	Integer	Set of Bits	Set of Bits
Sensitivity	Integer	Set of Bits	(NONE)
Clearance	Integer	Set of Bits	(NONE)

Thus, information labels have three components: classification, compartments, and markings, whereas sensitivity labels and clearance have only the first two components.

Given any two labels (information, sensitivity, or clearance), there is a relationship called *dominance* between them, defined as follows:

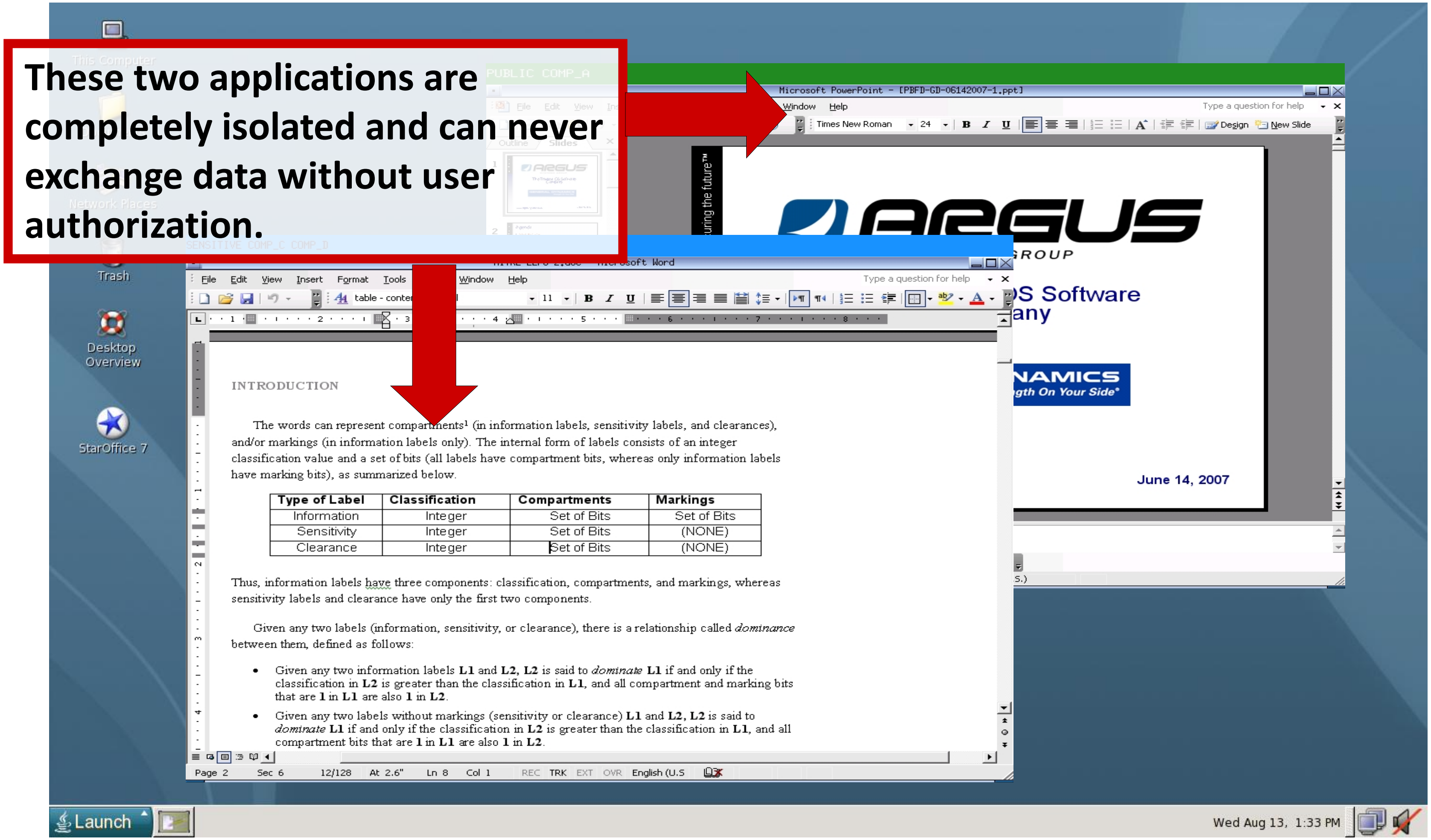
- Given any two information labels **L1** and **L2, L2** is said to *dominate* **L1** if and only if e classification in **L2** is greater than the classification in **L1**, and all compartment and that are **1** in **L1** are also **1** in **L2**.
- Given any two labels without markings (sensitivity or clearance) **L1** and **L2, L2** is *dominate* **L1** if and only if the classification in **L2** is greater than the classification in , and all compartment bits that are **1** in **L1** are also **1** in **L2**.

This Word session cannot access any network.

Markings are for demonstration purposes only

Example #3: Application isolation

These two applications are completely isolated and can never exchange data without user authorization.



INTRODUCTION

The words can represent compartments[1] (in information labels, sensitivity labels, and clearances), and/or markings (in information labels only). The internal form of labels consists of an integer classification value and a set of bits (all labels have compartment bits, whereas only information labels have marking bits), as summarized below.

Type of Label	Classification	Compartments	Markings
Information	Integer	Set of Bits	Set of Bits
Sensitivity	Integer	Set of Bits	(NONE)
Clearance	Integer	Set of Bits	(NONE)

Thus, information labels have three components: classification, compartments, and markings, whereas sensitivity labels and clearance have only the first two components.

Given any two labels (information, sensitivity, or clearance), there is a relationship called *dominance* between them, defined as follows:

- Given any two information labels **L1** and **L2, L2** is said to *dominate* **L1** if and only if the classification in **L2** is greater than the classification in **L1**, and all compartment and marking bits that are **1** in **L1** are also **1** in **L2**.
- Given any two labels without markings (sensitivity or clearance) **L1** and **L2, L2** is said to *dominate* **L1** if and only if the classification in **L2** is greater than the classification in **L1**, and all compartment bits that are **1** in **L1** are also **1** in **L2**.

Page 2 Sec 6 12/128 At 2.6" Ln 8 Col 1 REC TRK EXT OVR English (U.S

Markings are for demonstration purposes only

More about PitBull

Compatibility Issues – Program Integration

- PitBull is binary and source compatible with RHEL
  - enhanced API and ABI
- Any program without a kernel module can be made to work on PitBull
  - all can be made more secure
  - some can be made very secure
- Most programs with a kernel component can be made to work
- PitBull software can be compiled on non-PB systems
  - requires only header files and libraries

Compatibility Issues – Non-PitBull Systems

- PitBull systems integrate seamlessly into networks
- A PitBull system can impose security on non-PB systems
  - you specify the security of network interfaces and hosts
- A PitBull system can connect a remote port, host, subnet, or network interface with its file system security policy
- PitBull supports both NFS and TNFS

Compatibility Issues – User Apps

- Linux applications and X Window apps will work on PitBull
  - StarOffice, etc.
- In general, all RHEL device drivers work on PitBull
  - it is possible that some driver might be found that exercises an incompatibility
- Software packages with kernel components may have problems
  - PitBull uses a modified cred structure
  - PitBull doesn’t use UID 0 (superuser) as a kernel override mechanism
- A VM running Windows can be run on PitBull
  - VM networking will be controlled by PitBull
- Microsoft ® apps can use WINE for PitBull compatibility
  - Crossover from CodeWeavers supports MS Office®

Software Development Issues

- PitBull extends the RHEL API
    - new system calls and libraries
    - extended functionality in existing system calls
- Software for PitBull can be compiled on non-PitBull systems
- Main topics programmers need to know:
    - programming issues related to privilege
    - label manipulation library calls
    - secure programming practices
- An advanced training course is available for software developers writing programs to be run on PitBull

What to Tell New Software Developers

- PitBull is a form of Red Hat Enterprise Linux (RHEL)
- The API and ABI for PitBull are backwards compatible
  - Software that compiles on RHEL will compile on PitBull
  - Standard software development tools for RHEL are fine for PitBull
- Files on a PitBull system have many more attributes than just user, group, and permission bits
- Processes on a PitBull system have many more attributes
- If a process is not going to work at multiple security levels, then PitBull for the most part can be ignored
- PitBull doesn’t have superuser
  - if you need UID 0 / root capabilities, you’ll need to use privileges
  - running admin tools from the shell require authorization, not root
  - you can use STAP (systemtap) to find privileges that a COTS program needs

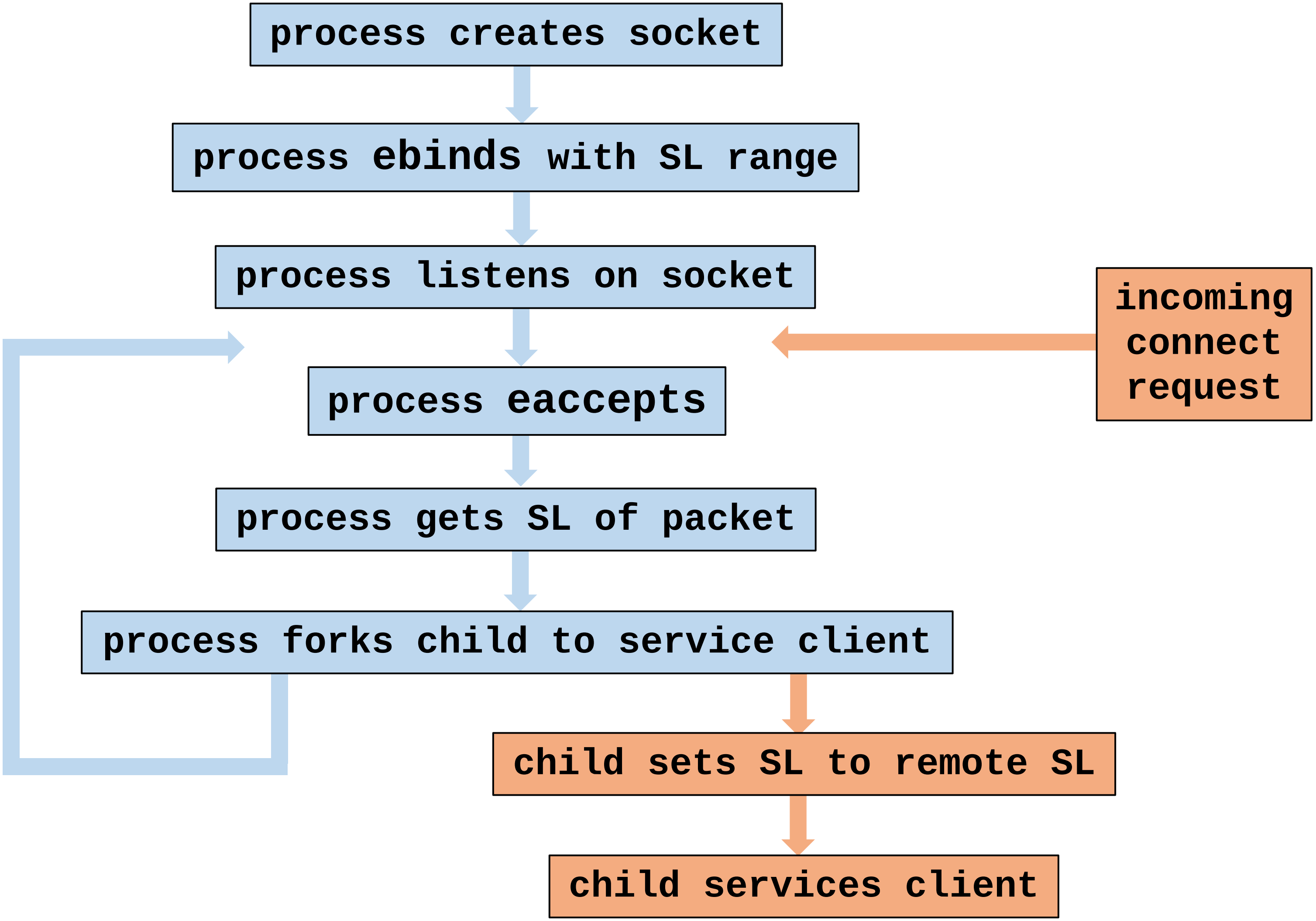
Polyinstantiation

- Sometimes a third-party program needs to be run at (or accessible by) multiple security levels
- Programs are frequently designed with the assumption that only one instance will be run at a time:
  - hard coded configuration files
  - hard coded network ports
  - hard coded temporary file names
  - hard coded log file names
- PitBull solves this problem by polyinstantiating network ports and directories
- Multiple instances of COTS programs at different SLs can be run simultaneously on PitBull systems without any modification to the programs or their configuration files
  - two different processes simultaneously can be listening on the same port and writing to the same file without any conflict

How to Build an MLS Network Server

- A privileged daemon (DP) sets its clearance range from SL1 to SL2
  - e.g., SEC to SEC A B C
- DP listens on port number X
  - it will capture all incoming connection requests for port X from remote clients that are within the SL1 to SL2 range
- When the connection is made from a remote client running at SL3, DP forks a child (DC) to handle the request
- DP goes back and waits for the next connection request
- DC gets the label (SL3) off the connection request
- DC changes its label to SL3, throws away its privileges, then services the request
  - all communication will be at SL3 since client and server are both at that SL
  - DC can read files that SL3 dominates
  - DC can only write into SL3 files and uses SL3 partitioned directory files
- Example: network time server

MLS Networking Flow Chart


process creates socket
process ebinds with SL range
process listens on socket
incoming connect request
process eaccepts
process gets SL of packet
process forks child to service client
child sets SL to remote SL
child services client

PitBull Training Courses

- **PitBull Introductory Training** course
  - 3-days, lecture and hands-on
- **PitBull Software Developer Training** course
  - 2-days, lecture and hands-on
  - PitBull Introductory Training is a prerequisite
- **PitBull Web App Developer Training** course
  - 1-day, lecture and hands on
  - PitBull Introductory Training is a prerequisite
  - written and conducted once, still being refined
- **PitBull Administrator Training** course
  - 3-days, lecture and hands on
  - PitBull Introductory Training is a prerequisite
  - in development
  - when available, PitBull Introductory will be cut to 2 days

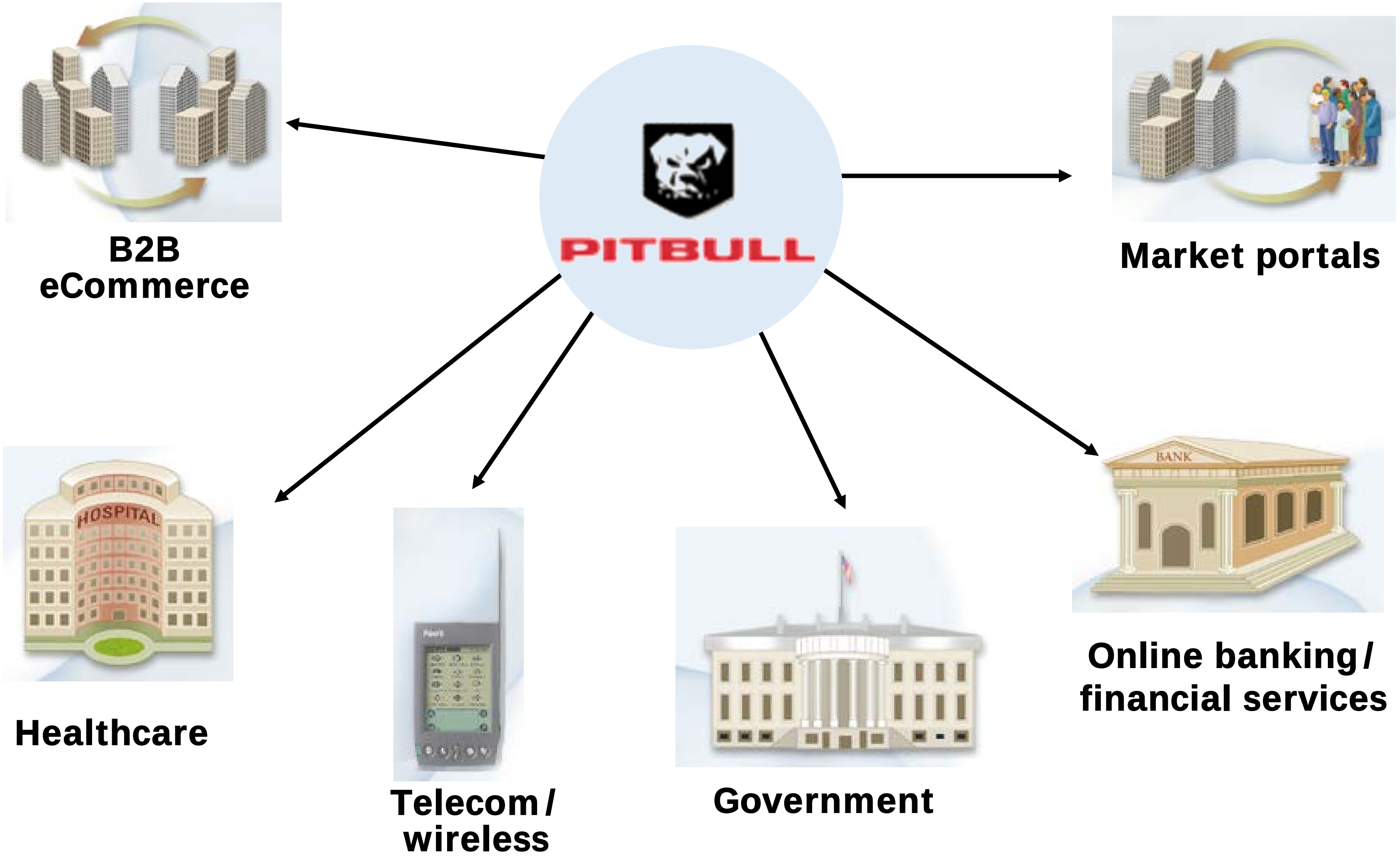
Key "Only" Points about PitBull

- Security only
  - no new functionality is added
- Software only
  - no hardware components
- Linux only
  - but interoperable with other systems
- Operating system only
  - no encryption, firewall, access control

Why PitBull?

- It’s REAL
  - It has been deployed in operational environments for over 25 years.
- It’s FUNCTIONAL
  - It has been evaluated at EAL4 under LSPP. Networking is fully integrated. Unmatched features.
- It’s EASY
  - Installation is trivial. Lock down scripts and tools are included. Training is available.
- It’s COMPATIBLE
  - All applications work without modification.

Where is PitBull needed?



B2B eCommerce

Market portals

Healthcare

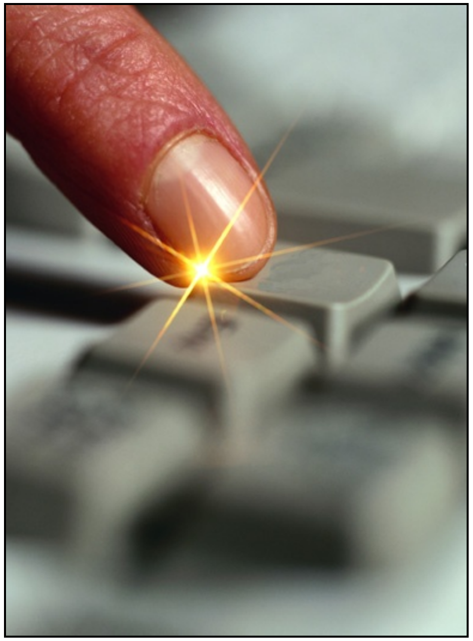
Telecom / wireless

Government

Online banking / financial services

Environments Ideal for PitBull Security

- Electronic Commerce
- Internet Banking
- Financial Services
- Multinational Commands
- ASP/CSP/ISP Servers
- Transaction Database Servers
- Medical/Health Services
- Secure Web Servers
- PKI / Certificate Authorities
- Trusted Firewalls



Questions and Answers

GENERAL DYNAMICS

PITBULL