PitBull and SELinux Mandatory Access Control Systems

Frank Caviggia

June 14, 2018

MITRE

DISCLAIMER

- **The MITRE Corporation is a not-for-profit organization that operates research and development centers sponsored by the federal government. We take on some of our nation's most critical challenges and provide innovative, practical solutions.**

- **Reference herein to any specific commercial products, process, or service by trade name, trademark manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring.**

Overview

- **Introduction and History**
- **Red Hat Enterprise Linux**
- **Common Criteria**
- **Basic Concepts**
  - Discretionary Access Control (DAC)
  - Role Based Access Control (RBAC)
  - Mandatory Access Control (MAC)
  - Polyinstantiation
  - Network Labeling
- **PitBull Overview**
- **SELinux Overview**
- **Applications of Technology**
- **Conclusions and Summary**

Who am I?

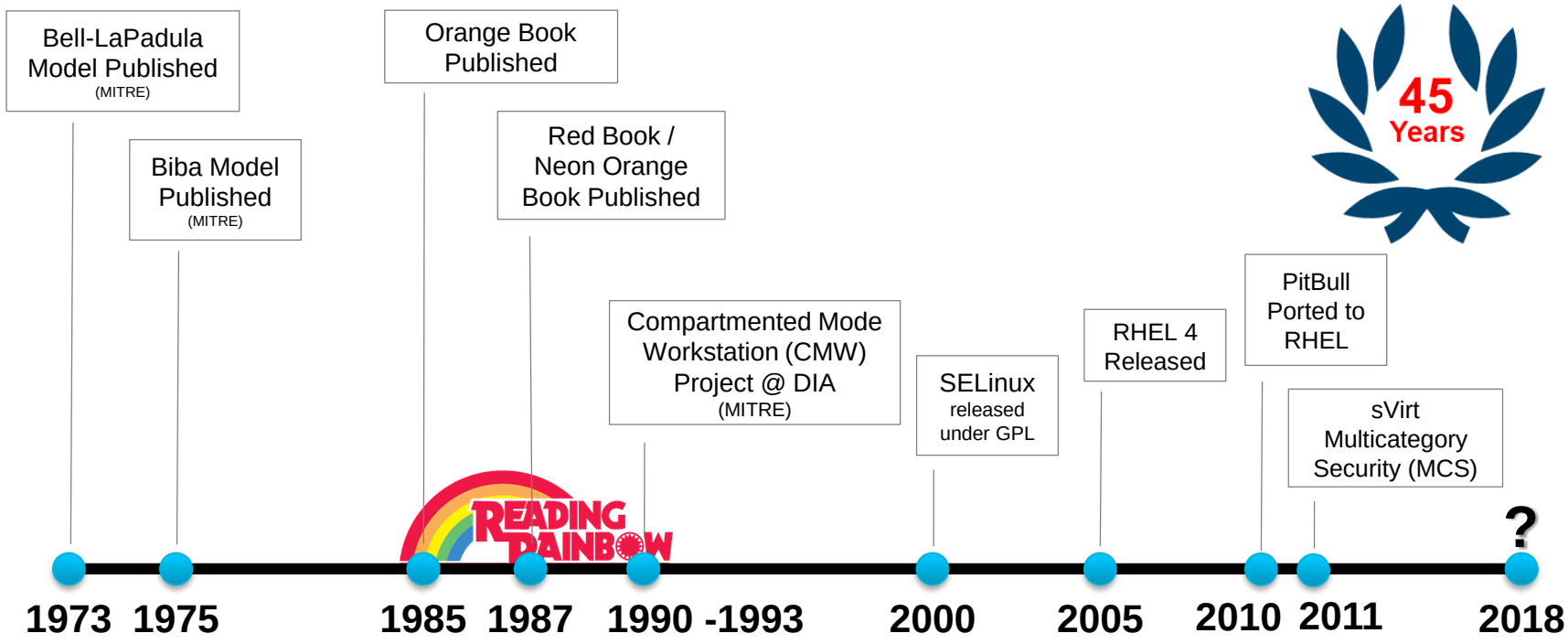
- **Lead Cybersecurity Engineer at MITRE Corporation**
  - Specialized in cross domain solutions and Linux security
- **Previous Employment**
  - Red Hat (2 ½ years)
  - Lockheed Martin (8 years)
  - Energetic Materials Research and Testing Center (5 years)
  - Compaq Computer Corporation (2 years)

Chasing the Security Rainbow

CC0, https://www.pexels.com/photo/rainbow-830829/

MITRE

History of Mandatory Access Control (Part I)



Currently Available Commercial Operating Systems with Mandatory Access Control and Multilevel Security (MLS) Support

- Honeywell SCOMP / BAE Systems STOP OS
- Solaris (SunOS MLS, CMW, Trusted Solaris, Solaris with Trusted Extensions)
- Red Hat (4.x + with SELinux)
- PitBull (AIX, Solaris, Linux)

MITRE

Red Hat Enterprise Linux (RHEL)

- **RHEL is commercially used Linux Operating System (OS)**
  - Common Criteria, FIPS 140-2, and DISA STIG evaluations have been completed for RHEL 6.x and 7.x
  - Both SELinux and PitBull utilize RHEL as the base OS

RED HAT® ENTERPRISE LINUX®

Life-cycle Dates

All future dates mentioned for "End of Production 1" and "End of Production 2" are close approximations, non definitive, and subject to change.

Version	General Availability	End of Production 1	End of Production 2	End of Production 3 (End of Production Phase)	End of Extended Life-cycle Support	End of Extended Life Phase	Last Minor Release
3	October 23, 2003	July 20, 2006	June 30, 2007	October 31, 2010	January 30, 2014	January 30, 2014	
4	February 14, 2005	March 31, 2009	February 16, 2011	February 29, 2012	March 31, 2017	Ongoing	4.9
5	March 15, 2007	January 8, 2013	January 31, 2014	March 31, 2017	November 30, 2020	Ongoing	5.11
6	November 10, 2010	May 10, 2016	May 10, 2017	November 30, 2020	June 30, 2024	Ongoing	
7	June 10, 2014	~Q4 of 2019	~Q4 of 2020	June 30, 2024	N/A	Ongoing	

Source: https://access.redhat.com/support/policy/updates/errata

Common Criteria Evaluations

- **Common Criteria is an internationally recognized standard required by National Information Assurance Partnership (NIAP)**
  - Evaluation Assurance Level (EAL) provides an initial security assessment of a product
  - Protection Profiles (PPs)

    - Previous Evaluations:
      - CAPP – Controlled Access Protection Profiled (DAC)
      - RBPP – Role Based Protection Profile (RBAC)
      - **LSPP – Labeled Security Protection Profile (MAC)**
        - Both PitBull and SELinux have had previous LSPP evaluations

    - Current Evaluations: OSPP – Operating System Protection Profile
      - Combined Previous Measures
        - Now required read details on DAC, MAC, RBAC!

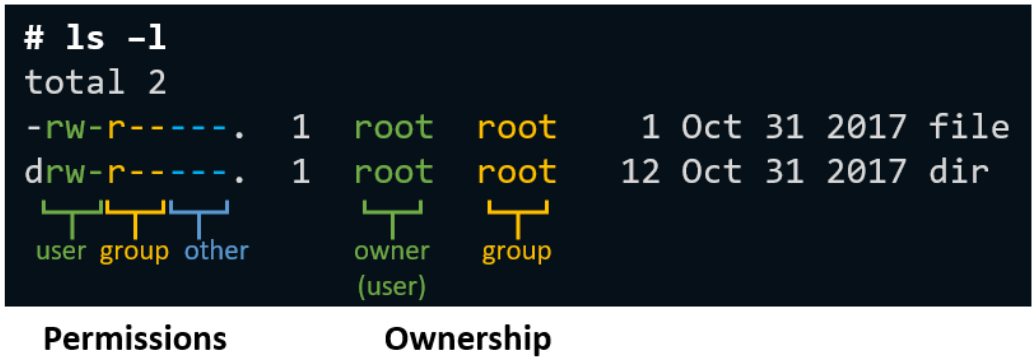
Linux Discretionary Access Control (DAC)

Discretionary Access Control (DAC)

Means of restricting access to objects based on the identity and need-to-know of users and/or groups to which the object belongs. *Controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (directly or indirectly) to any other subject.*

CNSSI 4009 Definition

- **DAC is implemented as Portable Operating System Interface (POSIX) Permissions**
  - User, Group, Other
  - Read, Write, Execute
  - Extended by Access Control Lists (ACLs)



Role Based Access Control (RBAC)

- **Systems should divide duties and roles into least privilege to prevent compromise and insider threat:**
  - PitBull and SELinux provide a mechanisms to enforce role separation
- **Minimum Roles should include:**
  - Security Administrator (security) and System Administrator (operations)
- **Additional Roles can include, but are not limited to:**
  - Log Administrator, Policy Administrator, Policy Approver, Backup Administrator, etc.
- **These roles should be used in combination with Unix RBAC using the `/etc/sudoers`[1] file and `sudo` command to ensure their integrity.**
- **Roles that can make security significant changes may also require Two-Person ("Four Eyes") and Two-Factor authorization (CAC/SIPR Token, RSA, YubiKey, etc.) to assume that role**

[1] https://www.unixtutorial.org/?s=sudoers

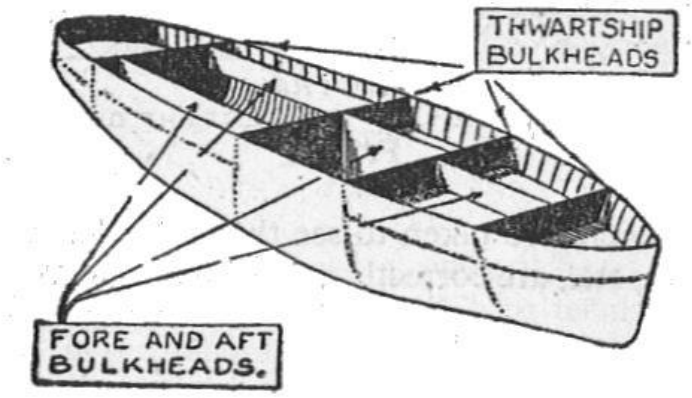
Mandatory Access Control (MAC)

- **Mandatory Access Control (MAC) is a security term defined by the Orange Book (5200.28-STD) in the "Rainbow Series" published by DoD and National Computer Security Center (NCSC) as part of the definition of a Trusted Computing Base (TCB).**
  - Control, contain, and constrain interactions between **Subjects** (processes, users) and **Objects** (data, files, devices)
    - Serves to supplement and reinforce Discretionary Access Controls (DAC)
  - Allows for Multitenancy (segregation/isolation) of data, processes, and users

Mandatory Access Control (MAC)

Means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization (i.e., clearance, formal access approvals, and need-to-know) of subjects to access information of such sensitivity.

CNSSI 4009 Definition

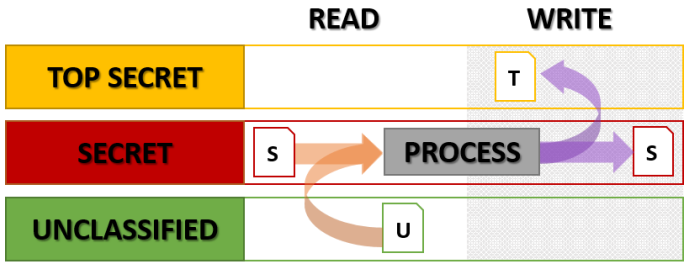


Source: https://en.wikipedia.org/wiki/Bulkhead_(partition)

MITRE

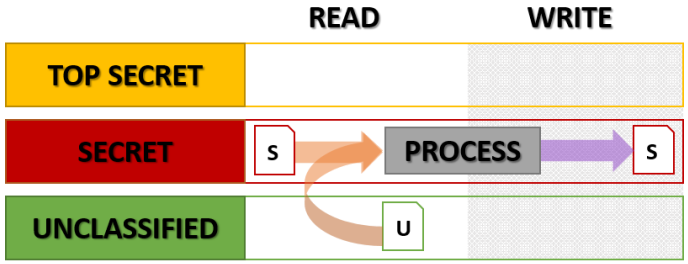
Mandatory Access Control (MAC) Theory

- **Bell-LaPadula Model**
  - Focused on **CONFIDENTIALITY** of Data
  - Read Equal/Down
  - Write Equal/Up



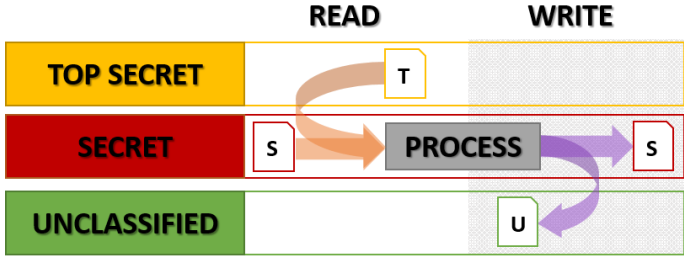
Bell-LaPadula Model

- **Strong Star Model**
  - Modification of Bell-LaPadula Model
  - Read Equal/Down
  - *Write Equal*



Strong Star Model

- **Biba Model**
  - Focused on **INTEGRITY** of Data
    - Higher Classification = Higher Integrity
  - Opposite of Bell-LaPadula Model
  - Read Equal/Up
  - Write Equal/Down



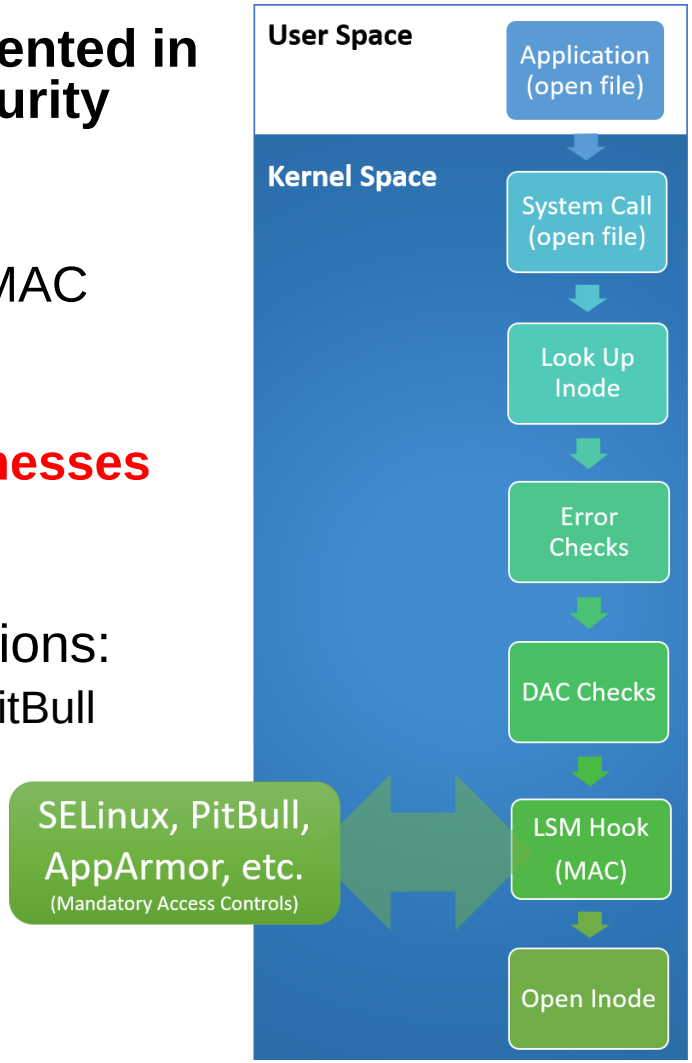
Biba Integrity Model

Credit: Caviggia, Frank C. (MITRE, 2017)

MITRE

Mandatory Access Control (MAC) on Linux

- **Mandatory Access Control is implemented in the Linux kernel under the Linux Security Module (LSM)**
  - DAC permissions are checked before MAC permissions
  - **MAC does not protect against weaknesses or exploits in the Linux Kernel**
  - Examples of Linux MAC Implementations:
    - General Dynamics Mission Systems PitBull
    - Security Enhanced Linux (SELinux)
    - AppArmor[1]
    - GrSecurity[1]
    - PARSEC[2]

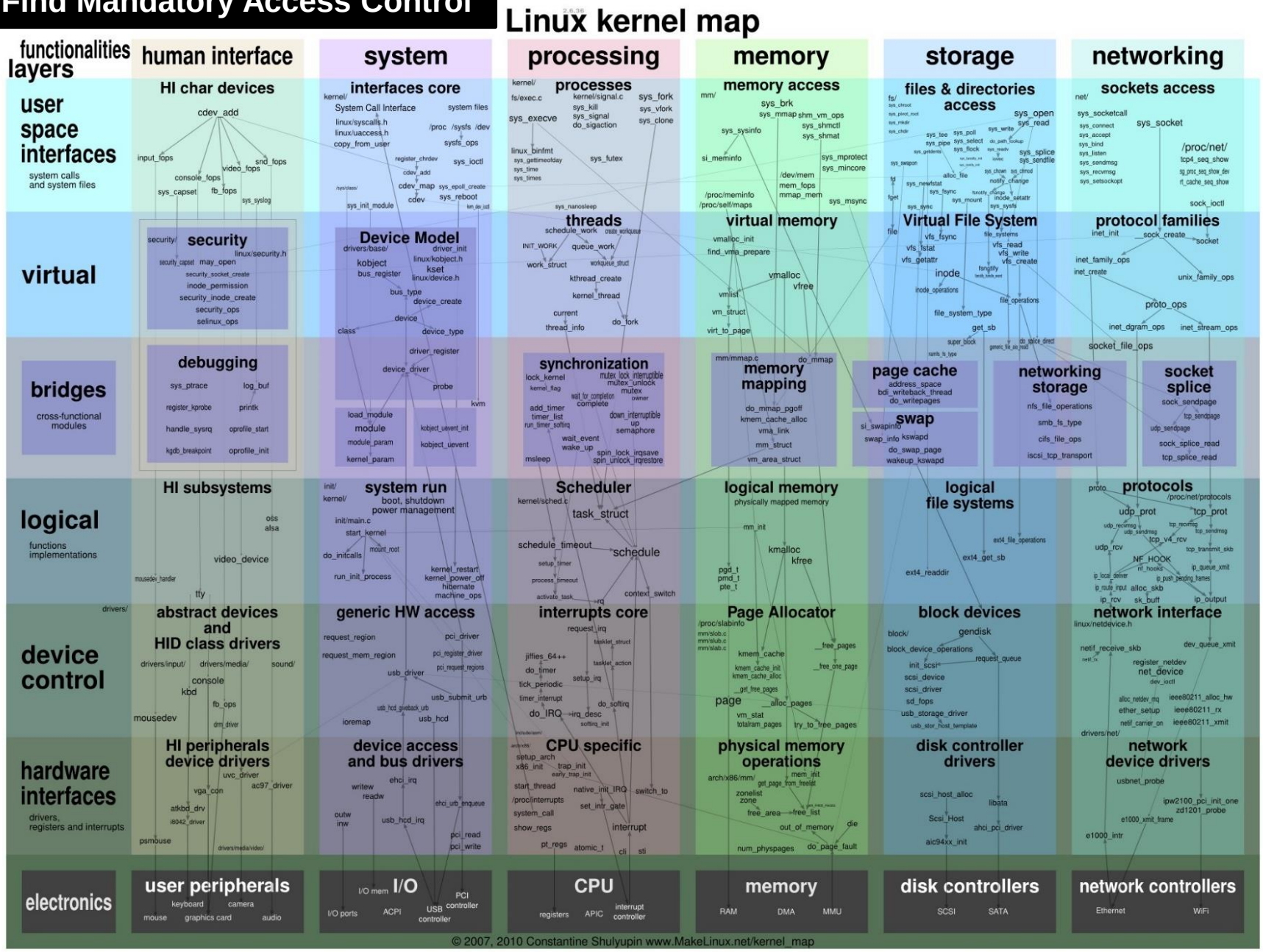


[1] No formal US Government assessment or authorization for use for MAC

[2] PARSEC is on Astra Linux (Russian)

MITRE

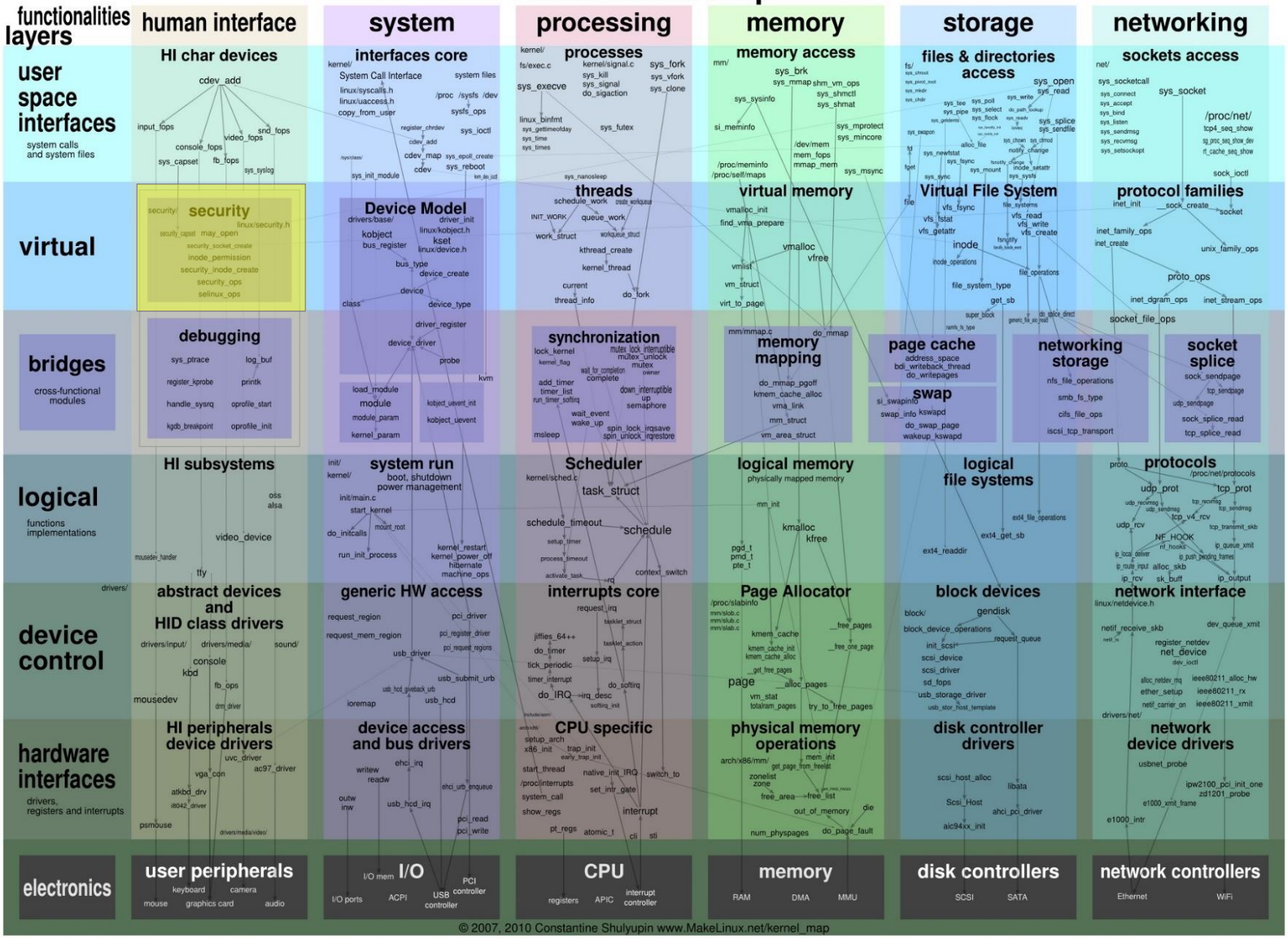
Find Mandatory Access Control



Source: http://www.makelinux.net/kernel_map/

MITRE

Linux kernel map

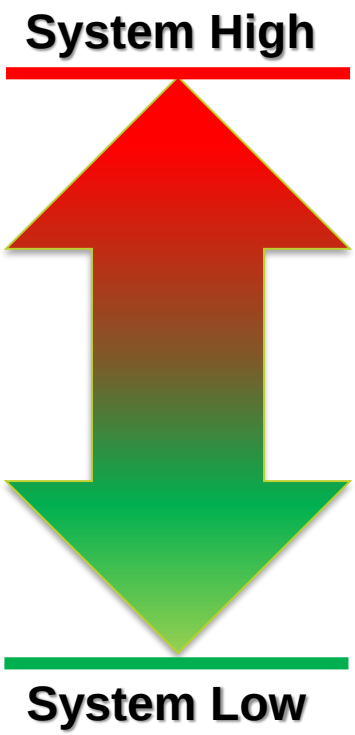


Source: http://www.makelinux.net/kernel_map/

MITRE

System High and System Low

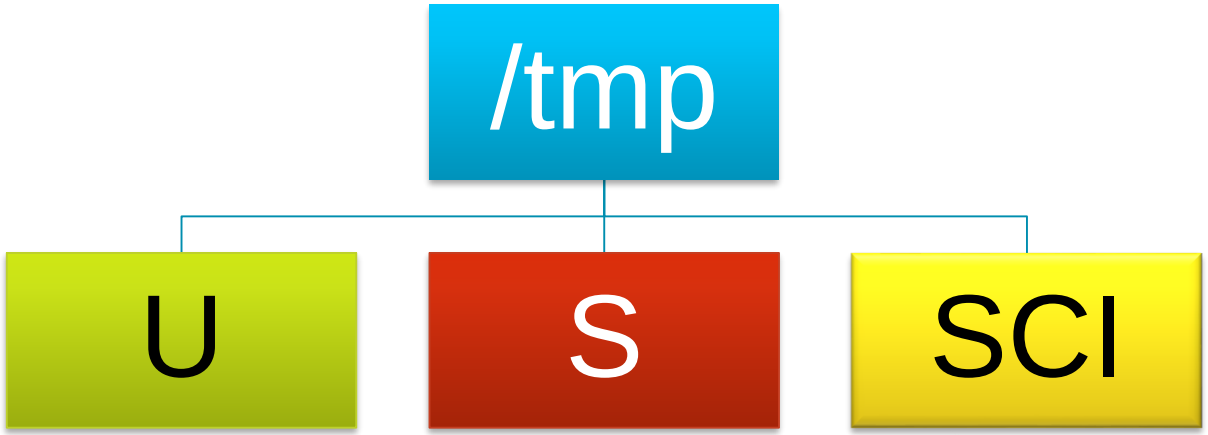
- **System High is the highest Sensitivity Level and Category being processed on the system**
- **System Low is the lowest Sensitivity Level being processed on the system**
- **DCID 6/3 referred to most systems (operating at a single level) as System High**
  - **Protection Levels 1-3 (PL1, PL2, PL3)** are for users are cleared for all information on the system, but without need to know for all of it.
  - **Protection Level 4 (PL4)** applies when at least one user lacks sufficient clearance for access to some of the information on the IS, but all users have at least a SECRET clearance
  - **Protection Level 5 (PL5)** applies when at least one user lacks any clearance for access to some of the information on the IS.
  - NIST 800-53 does not have any equivalent descriptions to DCID 6/3 for PL4 and PL5 systems



MITRE

Polyinstantiation

- **Polyinstantiation[1] isolates data to prevent data from bypassing MAC enforcement**
  - Apply concept around temporary folders (e.g. `/tmp`, `/var/tmp`, etc.), shared memory, networking, cronjobs, or user home folders
  - The kernel hides those isolations from the users, redirects storage to an isolated folder transparently



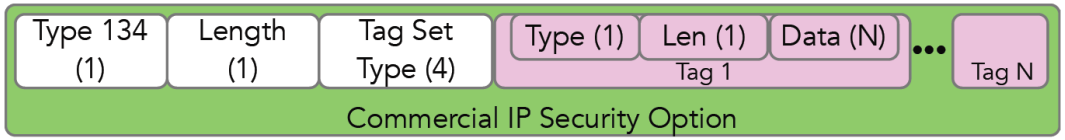
[1] https://www.ibm.com/developerworks/library/l-polyinstantiation/index.html (SELinux Examples)

Network Labeling

- **Standards developed and implemented to share data between systems that have implemented MAC (Solaris, Linux, etc.)**
  - Application of labels at an IP packet level
    - Commercial IP Security Option (CIPSO)
      - https://tools.ietf.org/html/draft-ietf-cipso-ipsecurity-01
      - Never officially adopted
      - IPv4 labels (Options 130, 133, 134)
      - Domain of Interpretation (DOI) and Tag Types (1, 2, 5, 7)
    - Common Architecture Label IPv6 Security Option (CALIPSO)
      - https://tools.ietf.org/html/rfc5570
      - IPv6 labels
      - Only supports Domain of Interpretation (DOI) and “free form” Tags
- **CIPSO and CALIPSO are only useful on internally controlled networks or VPN solutions due to risk of label alteration**

Network Labeling (IPv4)

- **Commercial IP Security Option (CIPSO)**
  - Option 134
    - Domain of Interpretation (DOI) and Tag Type (1, 2, 5, 7)
      - SELinux supports Tag Types 1, 2, 5
      - PitBull supports Tag Types 1, 2, 5, 7
        - 7 Requires PitBull to PitBull networking



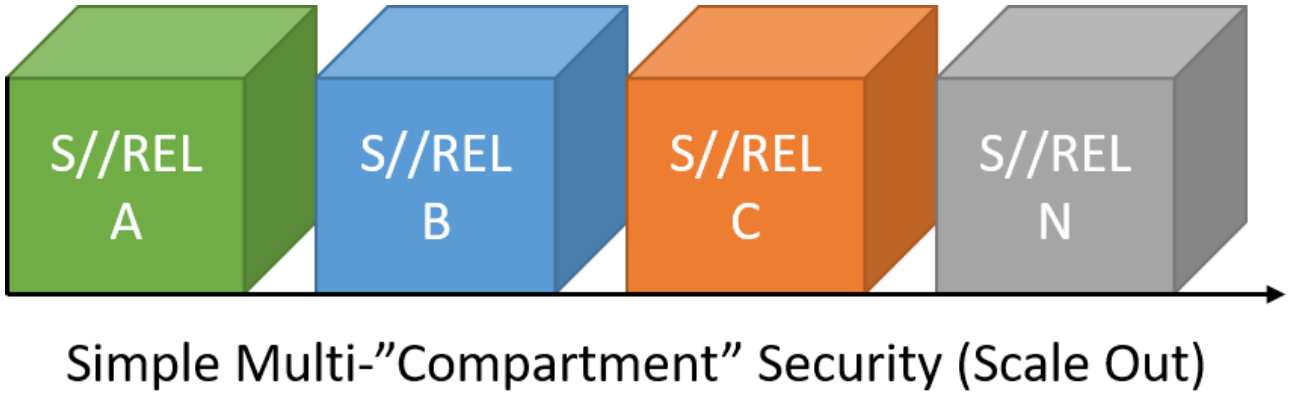
Tag Set Type

Numerical designator:
1 - Restrictive Bit Map for compartments/ categories (240 bits)
2 - Enumerated list of attributes
5 - Attribute Ranges
6 - Permissive Bit Map for release markings
7 - Free Form Characters

Credit: Irizarry Jr., Nazario (MITRE, 2018)

Multicategory Security (MCS)

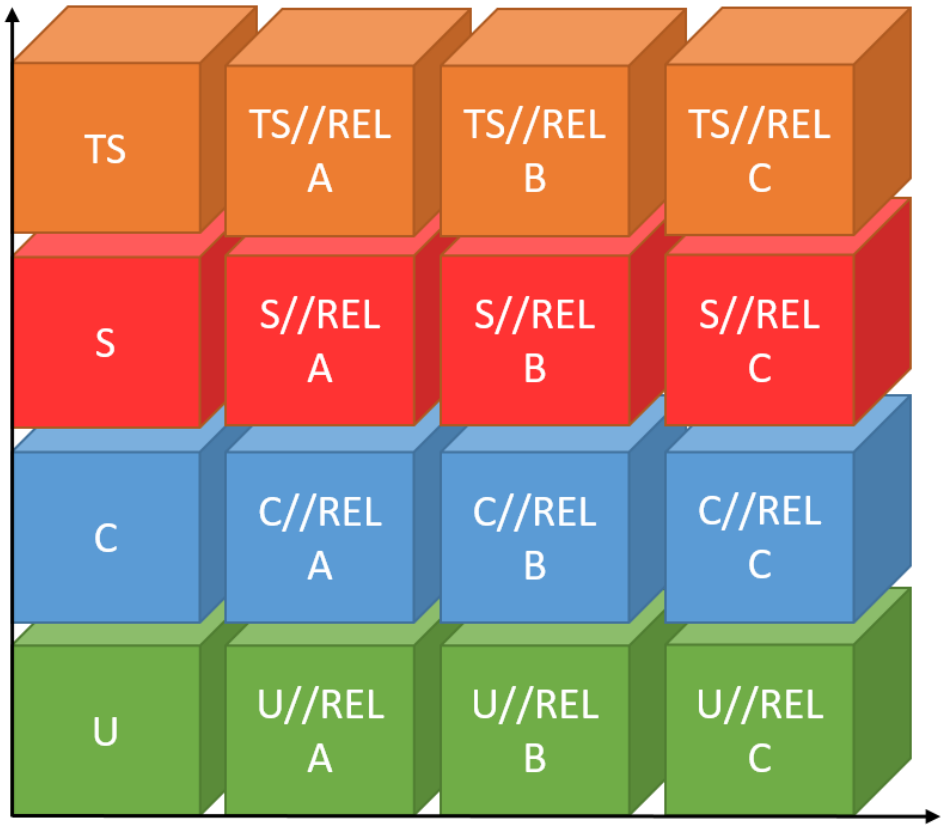
- **Multicategory Security**
  - Single Sensitivity Level, multiple compartmented sets of data
    - Business (Engineering, Operations, Finance, Management)
    - Government (Coalitions, Compartmented Data)



Credit: Caviggia, Frank C. (MITRE, 2017)

Multilevel Security (MLS)

- **Multilevel Security**
  - Multiple Sensitivity Levels and Multiple Compartments
  - Complexity
    - Adding dimensions of enforcement



Simple Multilevel Security (Scale Up and Out)

Credit: Caviggia, Frank C. (MITRE, 2017)

MITRE

PitBull Overview

MITRE

PitBull History

- **1987**
  - Addamax Corporation begins work on B1-level product called B1st Kit
- **1988**
  - Harris Corporation wins DIA contract for a compartmented mode workstation (CMW) and begins development of the Harris CMW product (HCMW) on SVR3
- **1991**
  - Addamax purchases HCMW from Harris Corporation. Renames it ACMW
- **1993**
  - Argus System Group, Inc.
- **1994**
  - ACMW ported to Solaris 2.4
- **1997**
  - First commercial installation (Credit Suisse, Switzerland)
- **1998**
  - ACMW renamed Gibraltar
- **1999**
  - Gibraltar ported to IBM AIX operating system
- **2000**
  - Gibraltar renamed to PitBull
- **2003**
  - Innovative Security Systems, Inc. formed and purchases all assets of Argus Systems Group
- **2007**
  - PitBull for AIX sold to IBM
- **2008**
  - General Dynamics begins port of TNE to PitBull
- **2011**
  - PitBull ported to Red Hat Enterprise Linux
  - General Dynamics purchases all assets of Innovative Security Systems
- **2012**
  - First release of PitBull for RHEL
- **2013**
  - General Dynamics discontinues development for PitBull on Solaris

MITRE Label Encoding File (LEF)

- **PitBull uses labels defined in the MITRE Label Encodings File**
  - Developed with assistance from MITRE for DIA's Compartmented Mode Workstation (CMW) configuration
  - Standard used as basis for Trusted Solaris, IRIX, AIX, and PitBull

```
CLASSIFICATIONS:
name=IMPLEMENTATION LOW;        sname=IMPL_LO;   value=0;
name=UNCLASSIFIED;              sname=U;         value=20;
name=PUBLIC;                    sname=PUB;       value=40;
name=SENSITIVE;                 sname=SEN;       value=60;
name=RESTRICTED;                sname=RES;       value=80;
WORDS:
name=ALL COMPARTMENTS;          sname=ALL;       compartments=1-5;
name=EXECUTIVES;                sname=EXECS;     compartments=1;
name=SALES;                     sname=SALES;     compartments=2;
name=FINANCE;                   sname=FINANCE;   compartments=3;
name=LEGAL;                     sname=LEGAL;     compartments=4;
name=ENGINEERING;               sname=ENG;       compartments=5;
REQUIRED COMBINATIONS:
COMBINATION CONSTRAINTS:
```

Compartmented Mode Workstation Labeling: Encodings Format;
MTR 10649 Rev. 1;
DIA DDS-2600-6216-93;
September 1993

PitBull Management

- **There is no "policy" like SELinux**
  - Labels use MITRE Labels Encoding File (LEF)
  - Labels and configuration is applied via UNIX-like atomic commands
    - Easier to train people who understand UNIX command line
    - Easier to script and manage with CM tools like Ansible

- **All PitBull label configurations stored under `/etc/security` directory and applied as extended attributes (XATTR) to the filesystems and processes**
  - System definition of security levels, categories, device labels, and user clearances

PitBull Sensitivity Levels and Categories

- **PitBull is a full Multilevel Security (MLS) configuration**
  - Works in Kernel at Linux Security Module (LSM)

- **Sensitivity Levels (SLs) and Category**
  - 32,768 potential Sensitivity Level labels
    - `0-32,767`
  - 1,024 or 4,096 (RHEL 6.8 PitBull) potential Categories
    - `0-4,095`

PitBull Mandatory Integrity Control (MIC)

- **PitBull utilizes Mandatory Integrity Control (MIC)**
  - Integrity Label (`xattr`) applied to files
    - Running at Kernel LSM provides adherence to the Biba model
    - Orthogonal and independent to enforcement of MAC Policy
  - PitBull provides the `chkintegrity` tool for checking alteration of data
    - 16-byte (128-bit) Hashed Check Block (HCB)
      - Based UK MoD Medium standard
      - Note: CRC-16 is a *simple integrity check*, it does not check for alteration of data and is not FIPS 140-2 compliant
    - Run during system startup or by ISSO user

- **Both labels (MAC) and integrity (MIC) are checked Kernel LSM**
  - SELinux relies on an external tool, such as AIDE or Tripwire, to provide integrity checks
  - PitBull can utilize AIDE or Tripwire in addition to its native MIC and `chkintegrity` tool.

PitBull Role Based Access Control

- **PitBull has the following roles predefined:**
  - Information System Security Officer (ISSO)
    - Establishes and maintains the security policy
  - System Administrator (SA)
    - Creates user accounts, groups, installs packages
  - System Operator (SO)
    - Performs backup duties, system shutdown
  - Authorization Manager (AUTH)
    - Manages authorization subsystem, approval for system changes

PitBull Modes of Operation

- **PitBull has two modes of operation related to the Trusted Computing Based (TCB) flag:**
  - **Configuration Mode**
    - Allows modification to TCB labeled files
  - **Operation Mode**
    - Running operational state
    - No privilege to override or modify TCB labeled files.

PitBull Directory Polyinstantiation

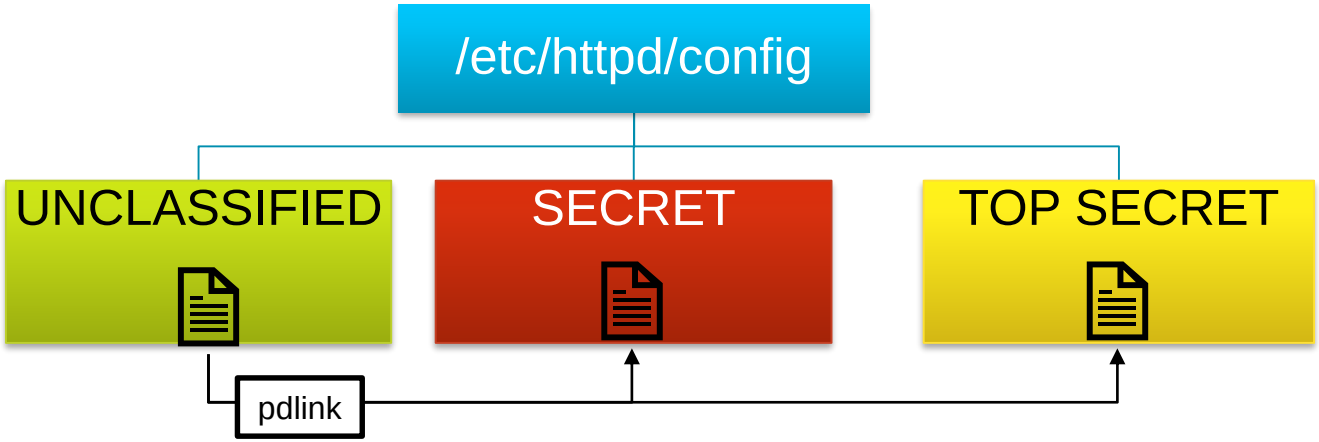
- **PitBull provides several types of directory polyinstantiation**

Type	Capability	Read Down	Write Up	User Secure
Single-Level	All files at same Sensitivity Level	YES	NO	YES
Multilevel	Range of Files at Multiple Levels	YES	YES (file names only)	NO
Partitioned	Files separated at Sensitivity Level	NO (unless pdlinked)	NO	YES

- **Polyinstantiation configuration provides flexibility when installing software that is not MLS aware compared to SELinux**

PitBull Directory Polyinstantiation (continued)

- **Example of a Partitioned directory**
  - Unclassified file can be linked to higher levels with `pdlink` command
    - Perfect for use with read only common configuration file.



PitBull Network Polyinstantiation

- **PitBull provides polyinstantiated network labeling**
  - Implemented at a kernel network-stack level
  - Processes listening to a network port (e.g. 443/TCP) at a defined clearance range
    - Allows software to share a single network port at multiple Sensitivity Levels and Categories

- **Network Traffic Labeling (`netrule` command)**
  - One utility to manage network labeling - `netrule`
  - Provides labeling to network packets

- **Trusted NFS between PitBull systems**
  - Supports full set of attributes between PitBull machines
  - Supports regular NFS mechanisms for non-PitBull Systems
    - Inherits label attributes of mount point

Other PitBull Capabilities

- **PitBull provides support for labeled X-Windows**
  - Graphical applications using XACE
  - SELinux MLS policy does not support X-Windows
- **Support for Windows Applications via Codeweavers Crossover**
  - Utilizes WINE (WINE Is Not an Emulator) Windows API port
  - More Information: https://www.codeweavers.com/pitbull

SELinux Overview

MITRE

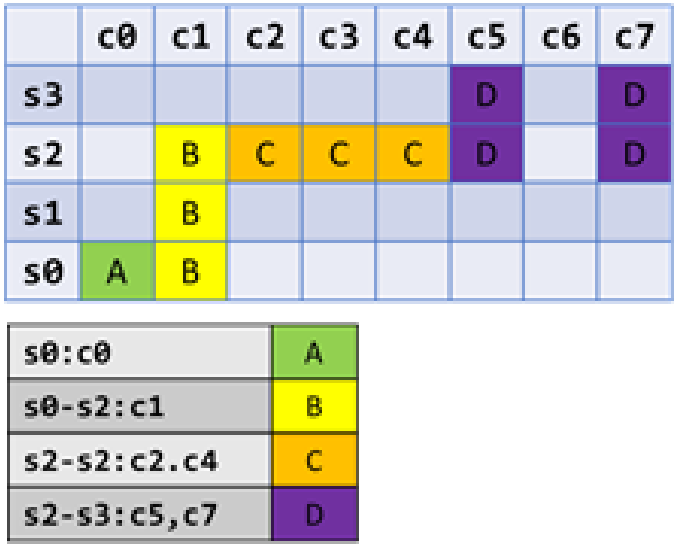
SELinux History

- **Security-Enhanced Linux (SELinux) was developed under the direction of the National Security Agency (NSA).**
  - Jointly developed with Secure Computing Corporation (SCC), MITRE, and University of Utah as the Flux Advanced Security Kernel (FLASK) operating system security architecture in the mid-1990s[1]
  - Released source code under GNU Public License (GPL) in December 2000
  - Mainlined into Linux Kernel in 2003
  - Enabled for general use in Red Hat Enterprise Linux 4 in 2005

[1] https://www.nsa.gov/what-we-do/research/selinux/

SELinux Sensitivity Levels and Categories

- **SELinux can use Sensitivity Levels and Categories in addition to Type Enforcement (TE)**
  - Scaling on Category is considered Multicategory Security (MCS)
  - Scaling on both Sensitivity Levels and Categories is considered Multilevel Security (MLS)
  - SELinux has 16 potential Security Levels (`s0-s15`)
  - 1024 potential Categories (`c0-c0123`) which can be used in combination

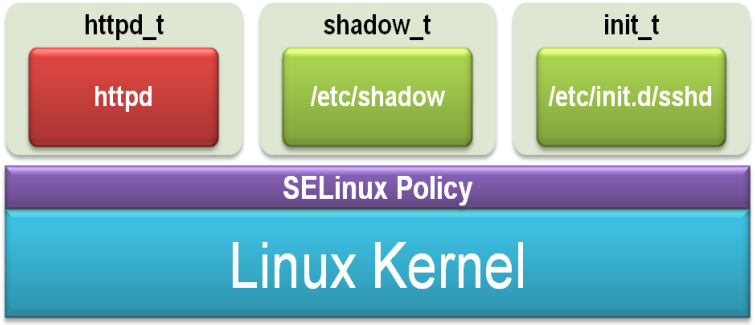


	c0	c1	c2	c3	c4	c5	c6	c7
s3						D		D
s2		B	C	C	C	D		D
s1		B						
s0	A	B						

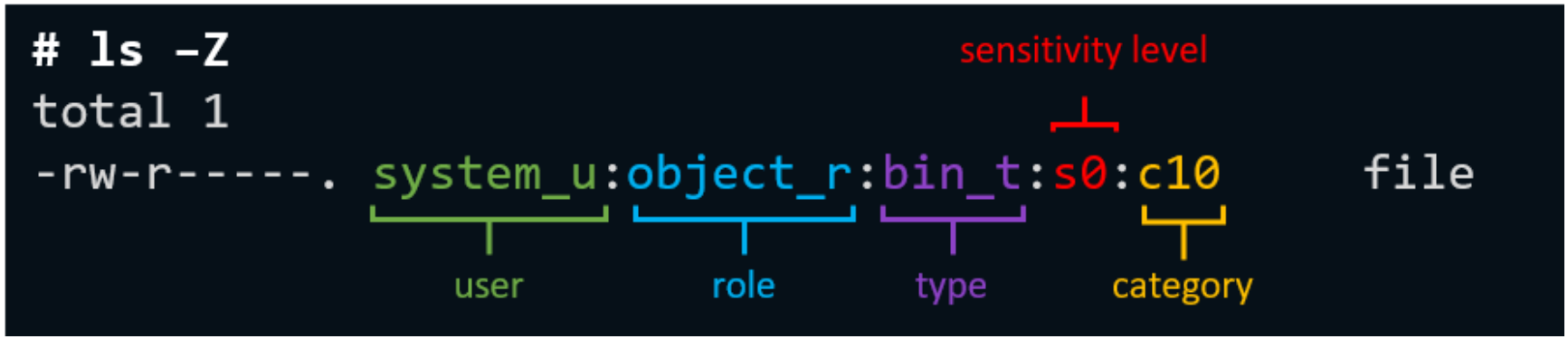
s0:c0	A
s0-s2:c1	B
s2-s2:c2.c4	C
s2-s3:c5,c7	D

SELinux Type Enforcement (TE)

- **Type Enforcement (TE) is used by SELinux to confine like services and control interactions with files and processes**
  - Example: Apache HTTPD
    - Executing process labeled with the type **httpd_t**
    - The executables themselves are labeled with **httpd_exec_t**
      - Allowed to read the configuration files labeled **httpd_config_t**
      - Allowed to read the web page content labeled **httpd_sys_content_t**
      - Allowed to write to logs labeled **httpd_log_t**
      - No access the **/etc/shadow** file, which is labeled with **shadow_t**



Sensitivity and Category (SELinux)



SELinux Role Based Access Control

- **Roles are defined in the Strict and MLS policy:**
  - User Role (user_r)
  - Staff Role (staff_r)
  - System Administrator Role (sysadm_r)
  - Audit Administrator Role (auditadm_r)
  - Security Administrator Role (secadm_r)

- **Users can transition between roles with `newrole` command**
  - By default System Administrator and Security Administrator roles are coupled in MLS policy
    - Roles should be further separated

SELinux Directory Polyinstantiation

- **SELinux has directory polyinstantiation based on per user, per level, per context, or any combination**
  - Administrative users are an exception (can see all directories)
  - Can be automatically applied by SystemD in RHEL 7.x using `PrivateTmp` feature*

```
# cat/etc/security/namespace.conf
/tmp /tmp-inst/ level root,admin
/var/tmp /var/tmp/tmp-inst/ level root,admin
$HOME $HOME/$USER.inst/ level
```

* https://access.redhat.com/blogs/766093/posts/1976243

SELinux Policy

- **Three components**
  - **Base policy:** Protects core system (e.g., kernel) and defines policy abstractions
  - **Application policies:** Protections specific to application
  - **Defined Roles:** Role Based Access Control
- **Policy modules types**
  - Type Enforcement File (te): Contains allow rules and interface calls associated with the confined domain
  - File Context File (fc): Contains all the default labeling file context
  - Interface File (if): Contains all interfaces used by other domains to interact with confined domain
- **Policies may be very large**
  - 1000s of types
  - >30,000 source rules
  - >300,000 type relationships
- **Possible to leverage existing vendor and community policies**
  - Original policies derived from NSA available as open source
  - Red Hat includes SELinux Policies for Targeted (default) and MLS
  - Tresys and Quark Security provide the Certifiable Linux Implementation Platform (CLIP), which is the basis for a number of Cross Domain Solutions (CDS)

Targeted SELinux Policy

- **Targeted SELinux Policy comes stock on Red Hat Enterprise Linux and Fedora Operating Systems**

Advantages	Disadvantages
• Default Configuration • Type Enforcement (TE) only • Can use **MCS** and Type Enforcement (TE) attributes applied via sVirt in Kernel Virtual Machine and Containers	• **Unconfined users** ○ **Processes started by users are unconfined** • **Services start as Unconfined transition to Type** • **No RBAC enforcement**

Strict SELinux Policy

- **Strict SELinux Policy was Distributed with RHEL 4+**
  - Currently, a subset of *Targeted* SELinux Policy with RHEL 7
  - Custom policy development would increase time for evaluation and authorization

Advantages	Disadvantages

• Subset of Targeted (RHEL 7+)
• Type Enforcement (TE) only
• RBAC applied to users
  ○ Confined users
• Can use **MCS** SELinux Labels and Type Enforcement (TE) attributes applied via sVirt in Kernel Virtual Machine and Containers
	• Complex configuration needed
	  ○ Development of custom Types, Roles
	  ○ MCS work required (Category only)
	• Significant testing required due to custom policy

Multilevel Security (MLS) SELinux Policy

- **Distributed with Red Hat Enterprise Linux**
  - Extremely complex to develop policies
  - Custom policy would increase time for evaluation and authorization

Advantages	Disadvantages

• Full Security Context Enforcement
  ○ All processes and users contained using Type Enforcement (TE) and **MLS** Security Context
• RBAC applied to users
  ○ Confined users
	• Complex configuration needed
	  ○ Development of custom Types, Roles
	  ○ User, data, process isolation must be defined
	  ○ MLS work required (Sensitivity and Category)
	• Significant testing required due to custom policy

Certifiable Linux Integration Platform (CLIP)

- **Certifiable Linux Integration Platform (CLIP) developed for Red Hat Enterprise Linux by Quark Security and Tresys, Inc.**
  - Base policy used by a number of Cross Domain Solution (CDS) vendors

Advantages	Disadvantages

• Full Security Context Enforcement
  ○ All processes and users contained using Type Enforcement (TE) and **MLS** Security Context
• RBAC applied to users
  ○ Confined users
• **SELinux Policy is the basis for several CDS systems which should lead to shorter assessment / authorization time**
	• Complex configuration needed
	  ○ Development of custom Types, Roles
	  ○ User, data, process isolation must be defined
	  ○ MLS work required (Sensitivity and Category)

https://github.com/QuarkSecurity/CLIP
https://github.com/TresysTechnology/clip/

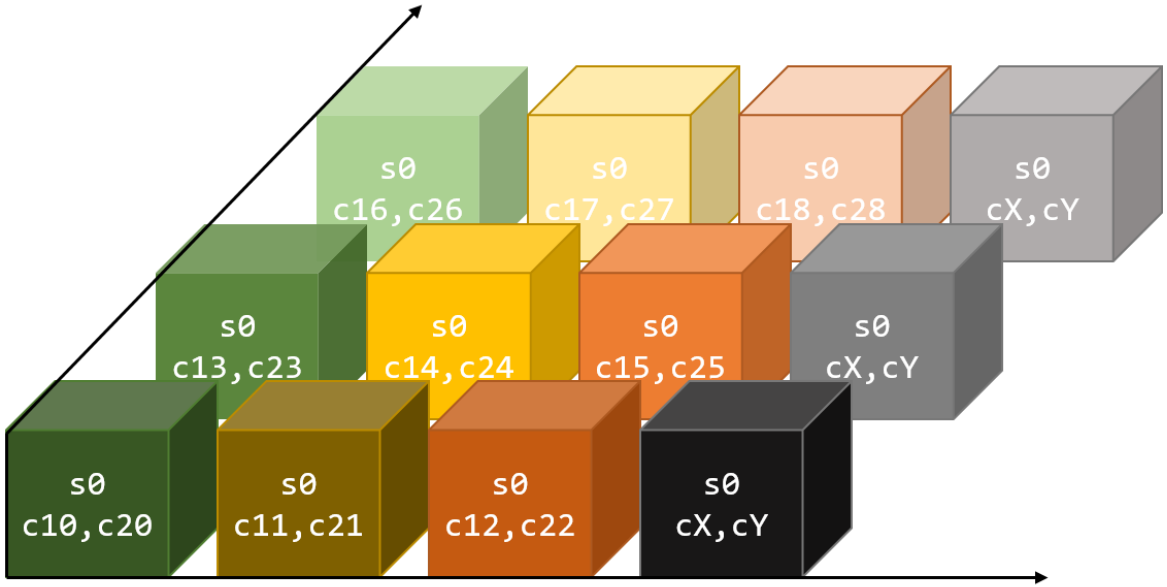
SELinux Network Labeling

- **There are three mechanisms to SELinux labels to networking:**
  - SECMARK and CONSECMARK
    - IPTables rules to label traffic and session
  - Netlabels
    - CIPSO IPv4 support (Tags 1, 2, 5)
    - CALIPSO support (IPv6)
    - Needed for MLS labeling
  - Labeled IPSEC
    - Supports a label set over Libreswan/Strongswan VPN (VPN per label)
    - High overhead for multiple labels
- **NFSv4.2 supports a "limited" number of SELinux labels**
  - Available in RHEL 7, default in RHEL 8
  - https://fedoraproject.org/wiki/Changes/LabeledNFS

More Information: http://www.paul-moore.com/presentations

SELinux Multicategory Security (MCS)

- **Multicategory Security (MCS) is utilized by Red Hat with Kernel Virtual Machine (KVM) hypervisor and Containers (e.g. Docker)**
  - Works at a single Sensitivity Level, with multiple Categories
  - Can be utilized by Targeted and Strict SELinux polices



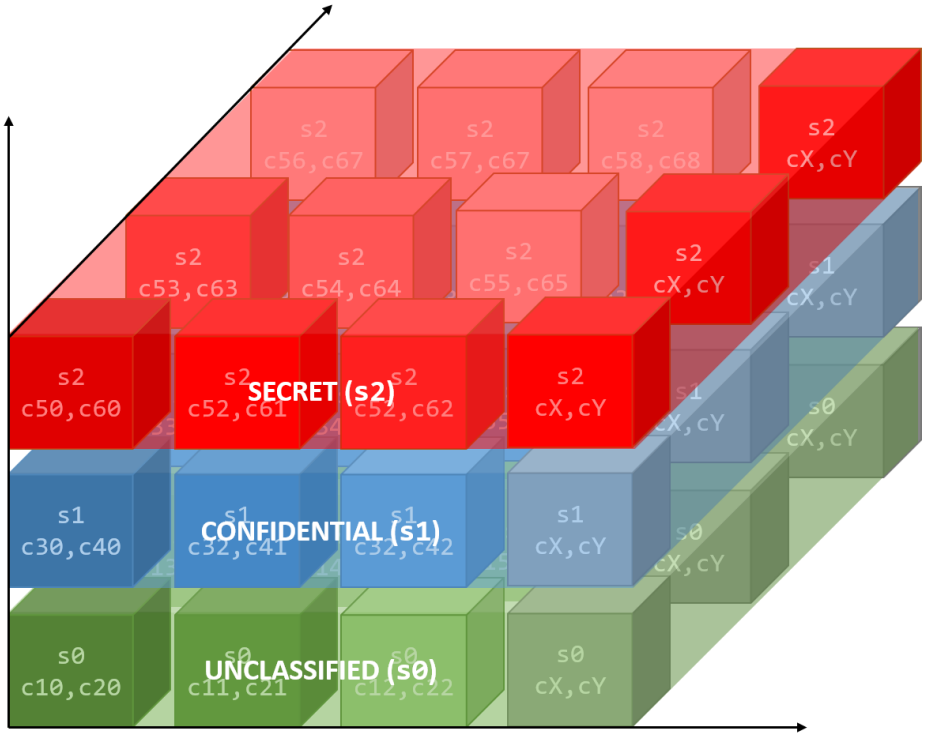
Multicategory Security (Scale Out – 2 Planes on Category)

Credit: Caviggia, Frank C. (MITRE, 2017)

MITRE

SELinux Multilevel Security (MLS)

- **Multicategory Security (MLS) works at a multiple Sensitivity Levels and multiple Categories**
  - SELinux (CLIP and MLS) polices utilize design



Credit: Caviggia, Frank C. (MITRE, 2017)

Multilevel Security (Scale Up and Out – 3 Planes on Sensitivity and Compartments)

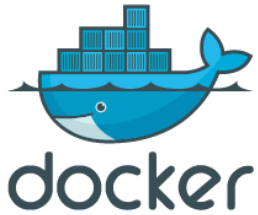
MITRE

Applications of Mandatory Access Control

MITRE

Containers, Virtualization, and Multitenancy

- **Containers are a popular way to deploy applications**
  - Docker runs on Windows, Linux, and Macintosh
  - Google Kubernetes and OpenShift
  - Windows now supports Kubernetes

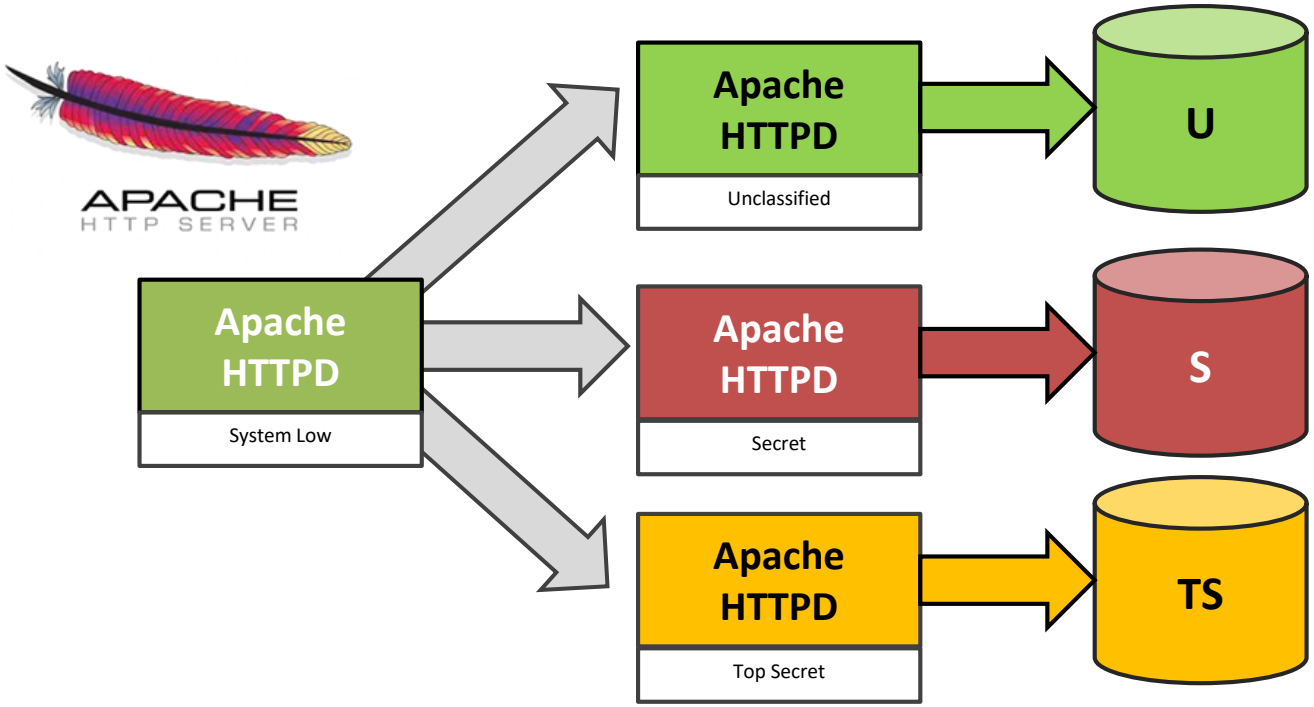


- **Virtualization is the foundation of cloud-based architectures**
  - AWS, Azure, Google Compute
  - Both Xen and KVM utilize MAC to provide separation, VMware doesn't utilize MAC

- **Does your cloud platform provide Mandatory Access Control?**
  - What enforces separation of processes and data? Multitenancy?
  - PL4 "cloud platforms"?

Multilevel Security & Web Servers

- **Displaying Apache HTTPD content at different Sensitivity Levels**



Multilevel Security & Databases

- **Oracle Labeled Security**
  - Ported to SELinux and PitBull

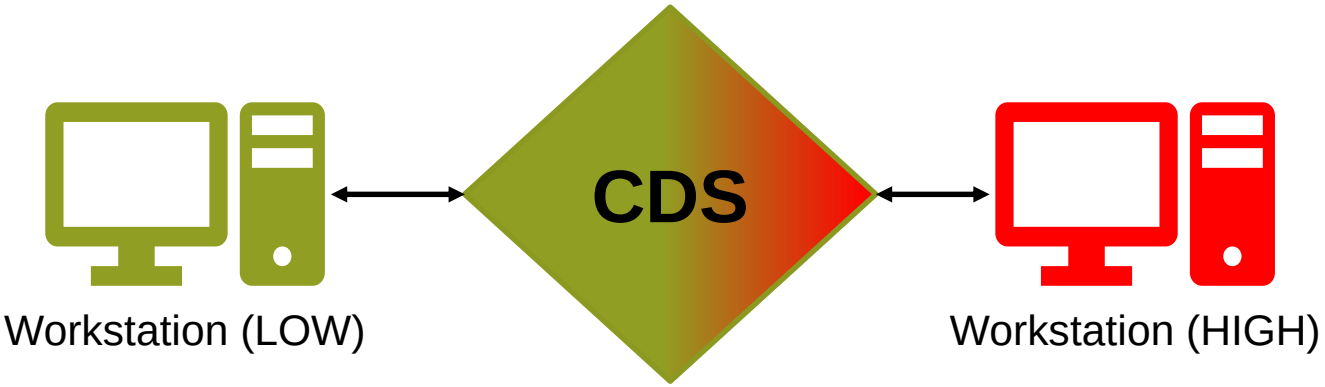
- **Crunchy Data Solutions PostgreSQL**
  - Ported to SELinux and PitBull

- **SQLite**
  - File-based datastore
  - Can be used with MLS

- **Apache Accumulo**
  - Key-Value Store, NoSQL
  - Originally developed by NSA
  - Potential for MLS use, but needs development

Cross Domain Solutions

- **Red Hat Enterprise Linux has become the foundation for most new CDS development**
  - PitBull and SELinux are the only validated MAC systems for Linux kernel
- **Oracle Solaris and BAE STOP OS are still around**
- **Lab Based Security Assessment (LBSA)**
  - Government lab assessment for verification of vendor claims
  - Typically done for Cross Domains Solutions



Conclusions

MITRE

Conclusions (PitBull)

Advantages

- Fully MLS configurable out of the box
  - Greater range of Sensitivity Levels (SLs) and Categories than SELinux
- Provides mandatory integrity checks at the Kernel LSM level
- PitBull provides a simpler management experience with an easier to understand configuration (MITRE LEF) and UNIX-like commands
  - Easier to add new sensitivity levels and categories
- Directory polyinstantiation flexibility compared to SELinux
- Full network polyinstantiation
  - Single utility to manage network labels (`netrule` command)

Easier to accommodate legacy (non-MLS aware) applications

- Full labeled NFS shares between PitBull systems
- Full support for X-Windows and Windows programs using Codeweavers Crossover/WINE

Disadvantages

- Proprietary to General Dynamics Mission Systems
- Lacks Type Enforcement (can possibly be made up with additional SLs)
- Slight lag behind on Red Hat Enterprise Linux support (current RHEL 7.5, PitBull on RHEL 7.3)
- Checksum algorithm not NIST FIPS 140-2 compliant (UK Standard)
- Does not support CALIPSO (IPv6 labeling)

Conclusions (SELinux)

Advantages

- Available by default on Red Hat Enterprise Linux
- Fine grained policy allows detailed configuration
- Type Enforcement provides a near unlimited way to scale
- Multicategory Security (MCS) scales well for containers and virtualization
- Free and Open Source Software, lots of published information

Disadvantages

- Complex policies and requires specialized review from Quark Security or Tresys to verify policy enforcement for changes
  - Applications may need custom policies developed to fully take advantage of MLS
- MLS is extremely difficult to manage and change, no X-Windows support in MLS
- Limited polyinstantiation capabilities
- Limited support for labeled NFS

Additional Conclusions

- **Kernel weaknesses and exploits remain an issue despite Mandatory Access Control implementations**
  - There is currently lack of diversity in secure operating systems development
- **Diversity of Mandatory Access Control systems is a good thing**
  - Diversity helps prevent exploits
- **Multilevel Security is extremely difficult to implement**
  - Specialized training for operations and management
  - Expensive and time consuming to do right
  - Requirements for use must be well understood before implementation

PitBull and SELinux: Summary

	PitBull	SELinux
Sensitivity Levels (Clearance)	**32,768**	16 (`s0-s15`)
Categories (Compartments)	1,024/**4,096** (current RHEL 6.8 release)	1,024 (`c0.c1023`)
Type Enforcement	None	Unlimited
Bell-LaPadula Model	Yes (Full and Strong Star)	Yes (Full and Strong Star)
Biba Integrity Model	**Yes** (Mandatory Integrity Control file checks at Kernel LSM independent of MAC)	No
System Integrity Checking	Yes, `chkintegrity`, AIDE or Tripwire, integrity checks on cron schedule	Yes, AIDE or Tripwire, integrity checks on cron schedule
Polyinstantiation of Networking	**Yes**	No
Polyinstantiation of Directories	Yes (single-level, multilevel, partitioned)	Yes (per user, per level, per context; root/privileged users exempted)
Networking Labels	CIPSO IPv4 (tags 1,2,5,**7**) / (CALIPSO IPv6 in Development)	CIPSO IPv4 (tags 1,2,5) / CALIPSO IPv6
Labeled Networking Filesystem	Modified NFS (Trusted NFS, PitBull unique) + CIPSO (Tag Type 7 provides full label set)	NFS v4.2 (limited number of tags)
Network Labeling Utilities	`netrule` (PitBull unique)	NetLabels, SECMARK/CONSECMARK (IPTables), Labeled IPSEC
Latest RHEL Support	6.8, 7.3	7.5
MAC Policy Configuration	MITRE Labels encoding file format (Compartmented Mode Workstation – MTR-10649, Rev 1, September 1993)	Tresys or Quark base policies (Certified Linux Integration Platform (CLIP)), Red Hat Multilevel Security (MLS) Policy
MAC Policy Implementation	Atomic, UNIX-like permissions, TCB configuration/operational mode	Policy implemented in fine-grained rule-based modules

Questions?

Conference Feedback Link:

https://bit.ly/2sGaBFk

MITRE

Discretionary Access Control (DAC) Permissions

	File	Directory
Read	Permission to read file contents	Permission to read file names in directory, but not read file content or metadata (size, type, ownership, and permissions)
Write	Permission to write file contents	Permission to create, delete, rename files in a directory
Execute	Permission to execute file contents	Permission to read file metadata and content and execute file, but not list files in a directory

Octal	Binary (rwx)	Permission
0	000	none
1	001	execute only
2	010	write only
3	011	write and execute
4	100	read only
5	101	read and execute
6	110	read and write
7	111	read, write, and execute (full permissions)

Access Control Lists (ACLs)

- **Access Control Lists provide fine grained control of DAC permissions**

```
# getfacl file
# file: file
# owner: root
# group: root
user::rw-
group::r--
other::---
```

```
# setfacl -m "u:jane:r--" file
# getfacl file
# file: file
# owner: root
# group: root
user::rw-
group::r--
other::---
jane::r--
```

Filesystem Attributes

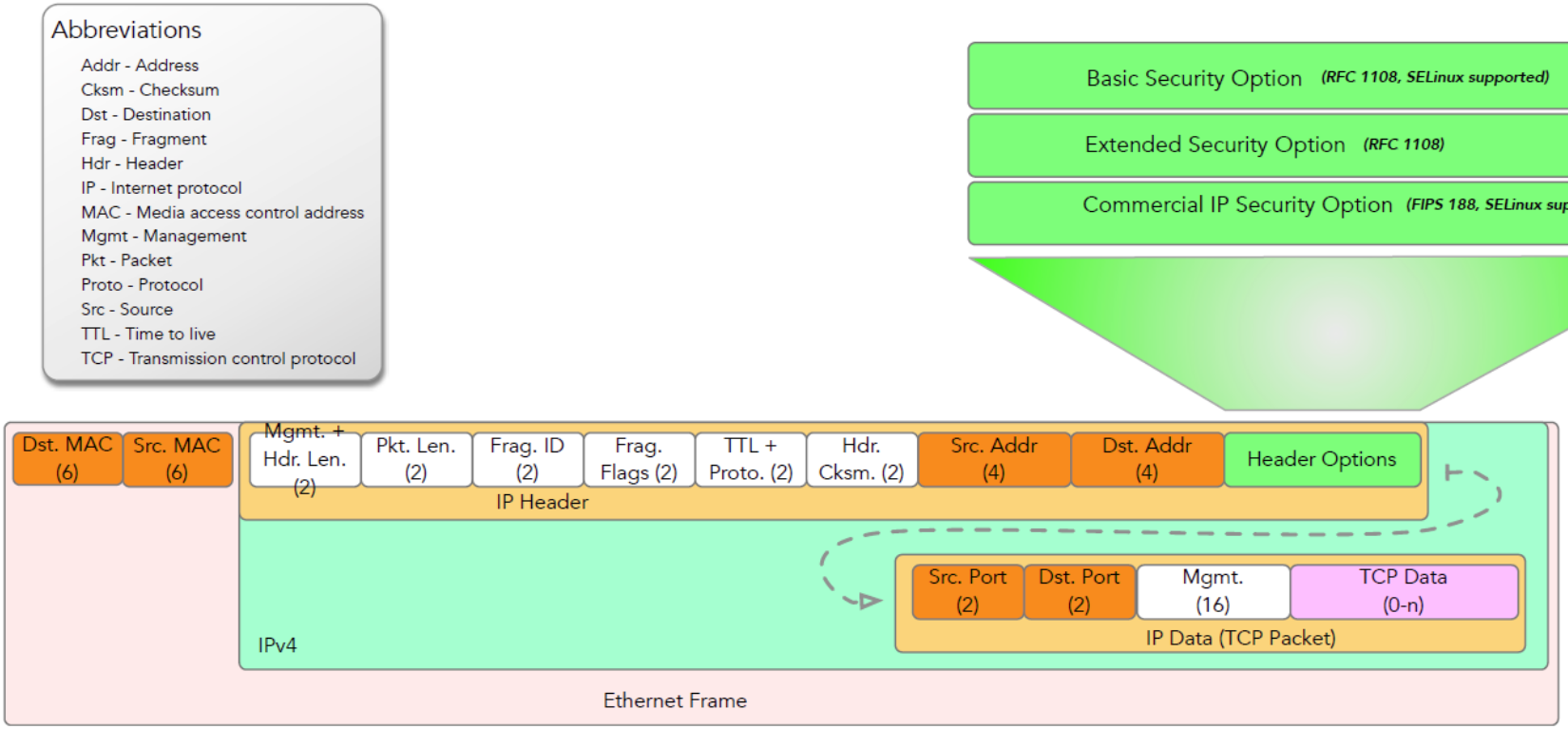
- **Using filesystem (ext3/4, XFS, etc.) attributes to provide additional hardening to configurations**
  - `chattr` command in Linux

```
# chattr +i /etc/sysconfig/selinux
```

a	**Append Only**
c	Compressed
d	No Dump
e	Extent Format
i	**Immutable**
j	Data Journaling
s	**Secure Deletion**
t	No Tail-merging
u	**Undeletable**
A	No atime Updates
C	No Copy on Write
D	Synchronous Directory Updates
S	Synchronous Updates
T	Top of Directory Hierarchy

Network Labeling (IPv4)

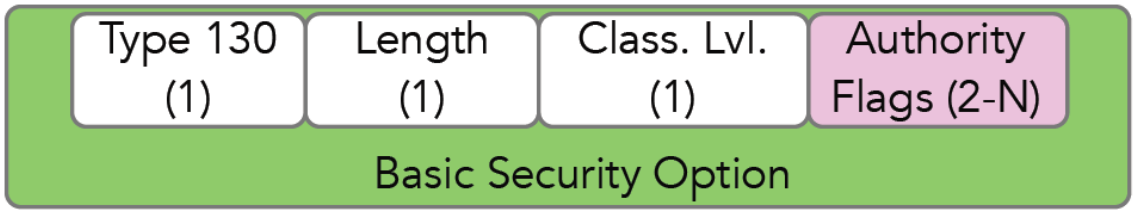
- **Commercial IP Security Option (CIPSO)**
  - Options 130, 133, 134



Credit: Irizarry Jr., Nazario (MITRE, 2018)

Network Labeling (IPv4)

- **Commercial IP Security Option (CIPSO)**
  - Option 130



Classification Levels

Reserved 4
Top Secret
Secret
Confidential
Reserved 3
Reserved 2
Unclassified
Reserved 1

Authority

National

GENSER
SIOP-ESI
SCI
NSA
DOE

Special Access

Not defined

Credit: Irizarry Jr., Nazario (MITRE, 2018)

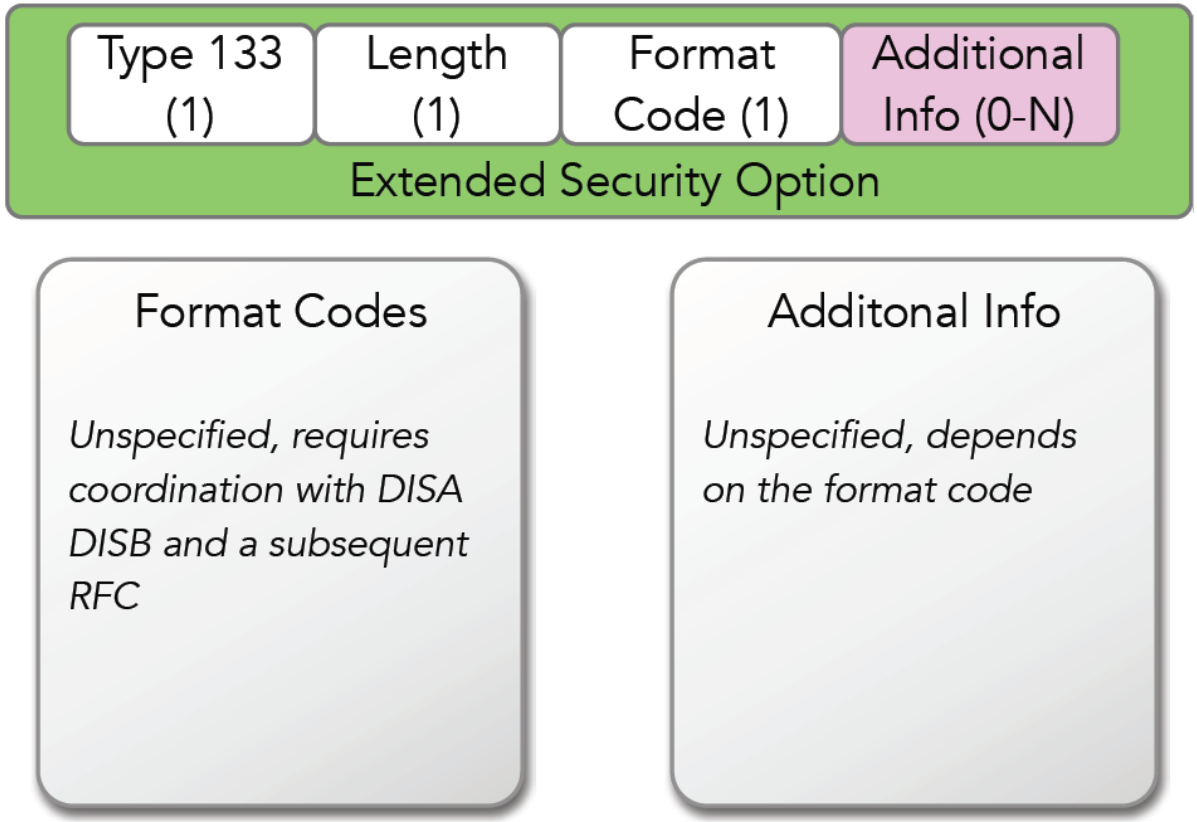
Minimum # of flags = 14

Number grows by increments of 7

MITRE

Network Labeling (IPv4)

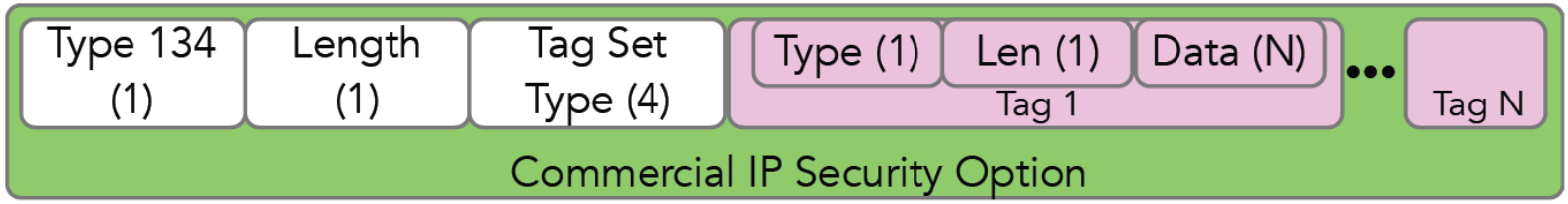
- **Commercial IP Security Option (CIPSO)**
  - Option 133



Credit: Irizarry Jr., Nazario (MITRE, 2018)

Network Labeling (IPv4)

- **Commercial IP Security Option (CIPSO)**
  - Option 134
  - Domain of Interpretation and Tag Type (1, 2, 5, 7)



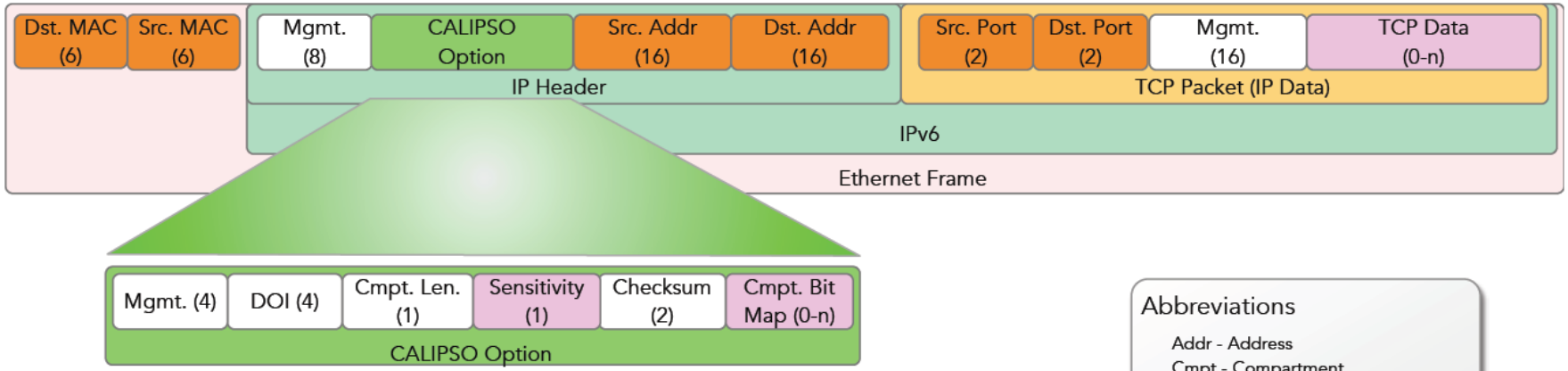
Tag Set Type

Numerical designator:
1 - Restrictive Bit Map for compartments/ categories (240 bits)
2 - Enumerated list of attributes
5 - Attribute Ranges
6 - Permissive Bit Map for release markings
7 - Free Form Characters

Credit: Irizarry Jr., Nazario (MITRE, 2018)

Network Labeling (IPv6)

- **Common Architecture Label IPv6 Security Option (CALIPSO)**
  - IPv6 Labels
  - Only supports DOI and “free form” Tags



Abbreviations

Addr - Address
Cmpt - Compartment
DOI - Domain of interest
Dst - Destination
IP - Internet protocol
MAC - Media access control address
Mgmt - Management
Src - Source
TCP - Transmission control protocol

Credit: Irizarry Jr., Nazario (MITRE, 2018)

MITRE

Evaluating Permissions (PitBull)

- **`secls –s <file>`**
  - Show sensitivity labels for file
- **`secls –t <file>`**
  - Show integrity labels for file
- **`getpsl <pid>`**
  - Show sensitivity labels for process
- **`getptl <pid>`**
  - Show integrity labels for process
- **`secps <pid>`**
  - Show process security attributes
- **`azlist <user>`**
  - Show authorizations for user
- **`azcheck <authorization>`**
  - Check if user has authorization; 1=Authorization
- **`netrule il`**
  - List interface rules
- **`netrule hl`**
  - List host rules

Management Commands (PitBull)

- **`setfsl –a “<SL>” <file>`**
  - Sets sensitivity labels for file
- **`setpsl –a “<SL>” <pid>`**
  - Set sensitivity labels for process
- **`setftl “<TL>” <file>`**
  - Set integrity labels for file
- **`setptl –eMma “<SL>” <pid>`**
  - Set integrity labels for process
- **`setfpv –a “<priv>” <file>`**
  - Set privilege for file
- **`setppv –a “<priv>” <pid>`**
  - Set privilege for process
- **`setuclear –b “<SL>” <username>`**
  - Set clearance for user

Management Commands (PitBull) (continued)

- **`seels <file>`**
  - See label attributes in binary
- **`chkintergrity –c [<file>]`**
  - Generate checksums (128-bit CRC-16 HCB) identify integrity issues
- **`makeidb <file>`**
  - Create new integrity database
- **`sumargus <file>`**
  - Generate checksum (128-bit CRC-16 HCB) for a file
- **`setfsf –e “<FSF1,FSF2>” <file>`**
  - Set security flags like Read Only (FSF_and Append Only (FSF_APPEND)
- **`checklef –f <file>`**
  - Verify MITRE Labels Encoding (LEF) file
- **`setsyslab`**
  - Load MITRE Label Encoding File (LEF) into the Kernel
- **`setrunmode (o|c)`**
  - Set system to opearational or configuration mode
- **`asninit –m init <file>`**
  - Initialize CIPSO rules

Management Commands (PitBull) (continued)

- **`chazdb –m LOGIN:u+user1,user2`**
  - See label attributes in binary
- **`setafauth +p <AUTHORIZATION> <file>`**
  - Set authorization for a file
- **`setkat`**
  - Enable authorization database in the kernel.

Evaluating Permissions (SELinux)

- **`ls -alZ`**
  - Will show DAC permissions along with the SELinux security contexts
- **`find -Z`**
  - Searches for files/directories based on specific criteria
- **`ps -aefZ`**
  - Current process list with SELinux security contexts
- **`id -Z`**
  - Displays current user security context
- **`umask`**
  - Shows current file creation permissions

SELinux Status

- **`getenforce`**
  - Shows current status of SELinux
- **`sestatus`**
  - Shows the current status of SELinux as well as the policy being used
- **`setenforce {0|1}`**
  - Changes the mode of SELinux between enforcing and permissive mode

SELinux Configuration File

- **Settings in `/etc/selinux/config`**
  - `SELINUX=enforcing`
    - SELINUX policy is enforced
  - `SELINUX=permissive`
    - SELINUX policy warns instead of prohibiting action
  - `SELINUX=disabled`
    - SELINUX policy is disable
  - `SELINUXTYPE=targeted`
    - Only targeted daemons are protected (Type Enforcement)
  - `SELINUXTYPE=mls`
    - Full SELinux protection
  - `SELINUXTYPE=clip`
    - Quark Security (SELinux Maintainers) custom version of SELinux (sometimes used on CDS systems)

SELinux Commands

- **`sestatus`**
  - Shows the running policy and enforcement status
- **`chcon`**
  - Change the security context (relabel)
  - Does not permanently change security context (relabel) – restorecon will go back to default
- **`restorecon`**
  - Restores the default SELinux context for files
- **`fixfiles`**
  - Checks and corrects security context on the filesystem
- **`getsebool`**
  - Shows the SELinux boolean value(s)
- **`setsebool`**
  - Toggle policy booleans
  - Use -P to make change persistent

    - Disclaimer: May break how system functions

Additional SELinux Commands

- **`semodule`**
  - List and manage running SELinux modules. (`semoudule -l`)
- **`semanage`**
  - Role control (`semanage user -l`)
  - User clearance control (`semanage login -l`)
  - Permanently change security context (relabel) – (`semanage fcontext -l`)
- **`audit2why`**
  - Translates SELinux audit message into description of why access was denied
  - Use `/var/log/audit/audit.log` as input
- **`audit2allow`**
  - Generate Policy allow rules from denied operations in log messages
- **`avcstat`**
  - Shows statistics for the SELinux Access Vector Cache (AVC)
- **`checkpolicy`**
  - SELinux policy compiler

SELinux Unconfined Domains

- **Unconfined Domains run without SELinux Protections**
- **Should only be defined for user processes not services or daemons**

- **Check by looking for unconfined/permissive domains listed in the policy**
  - `semanage permissive -l`
  - `seinfo -aunconfined_domain_type -x`
  - `seinfo --permissive -x`
  - `semodule -l | grep permissive`
- **Check live system by looking for processes with unconfined**
  - `ps -eZ | grep unconfined`

SELinux Policy Analysis Tools (SETools)

- **`apol`**
  - Graphical policy analysis tool
- **`seaudit`**
  - Analyze audit messages
- **`seaudit-report`**
  - Generates customizable audit reports
- **`sechecker`**
  - Used to modularly check SELinux policy
- **`sediff`**
  - Policy comparison tool
- **`secmds`**
  - Analyze and search SELinux policy

Policy Analysis is extremely difficult, if custom SELinux policy has been used, independent analysis may be required.

https://quarksecurity.com/
(Current Maintainers of SELinux Reference Policy)

http://www.tresys.com/