Proactive Security in Linux

Vit Mojzis

About me

- Vit Mojzis
- Security SELinux Userspace developer at Red Hat
- RHEL & Fedora Contributor (policycoreutils, setroubleshoot, libselinux, libsemanage, libsepol, udica)
- vmojzis@redhat.com
- https://github.com/vmojzis

Agenda

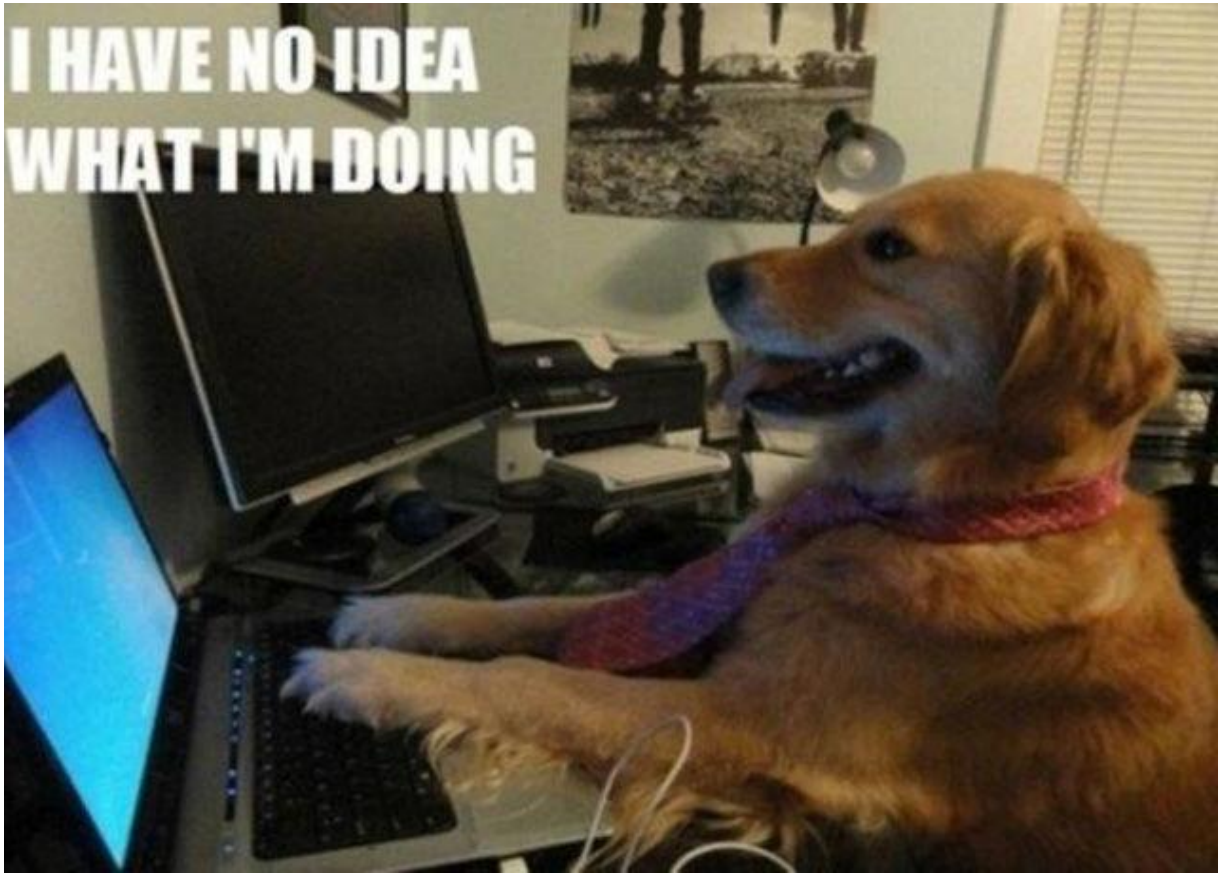
- Proactive Security
- Traditional Linux Security
- SELinux Security Policy
- SELinux and Cloud
- AVC Messages

Proactive Security

WHEN DO PEOPLE CARE ABOUT SECURITY?

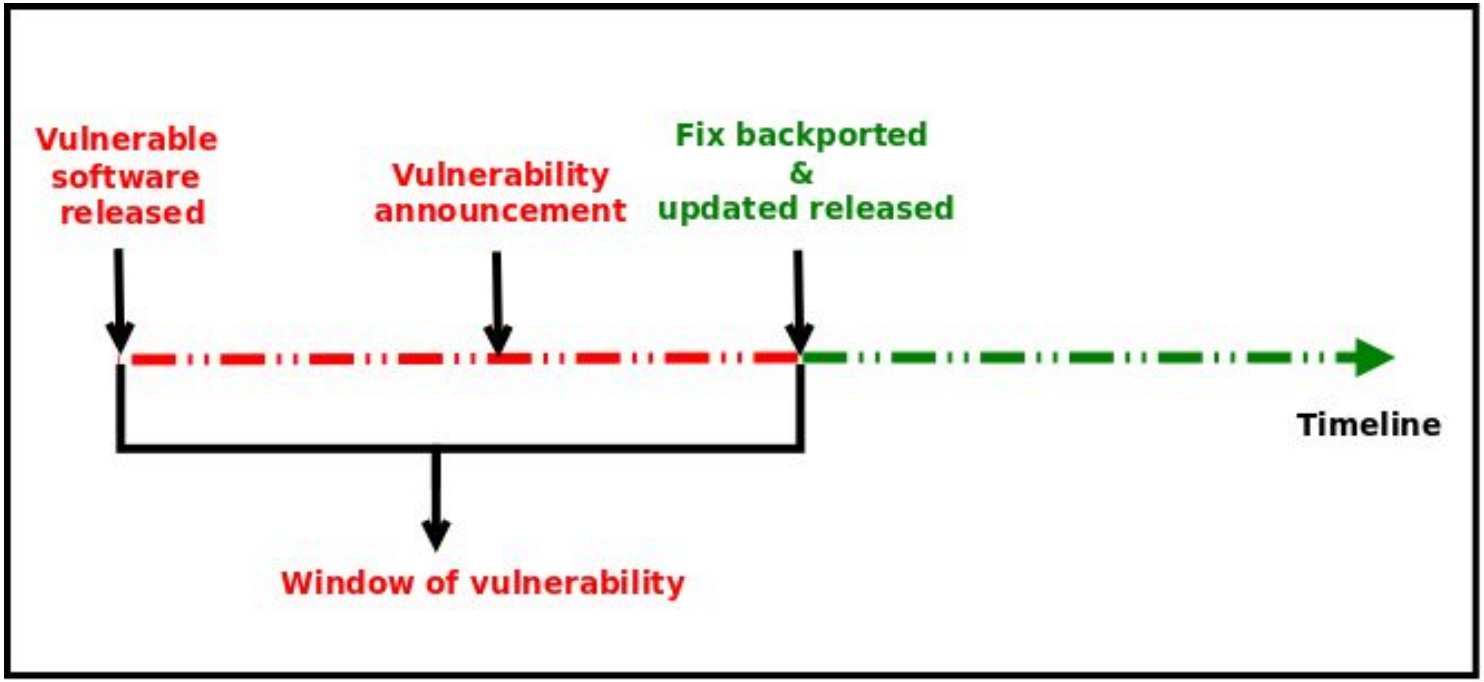

This Hack Will
Destroy Your Phone!

WHERE DO SECURITY ISSUES COME FROM?


I HAVE NO IDEA
WHAT I'M DOING

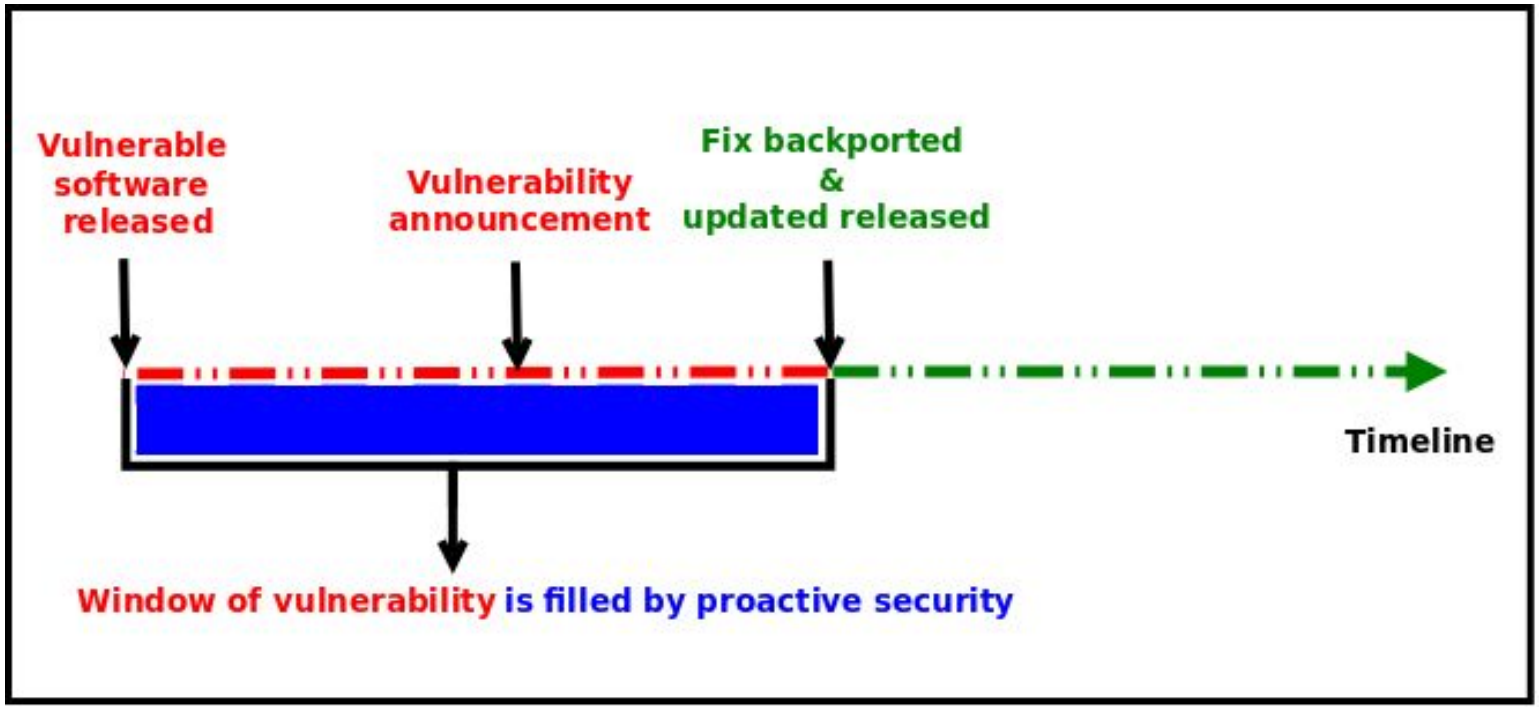
HOW ARE THEY FIXED?

REACTIVE SECURITY


Vulnerable
software
released
Vulnerability
announcement
Fix backported
&
updated released
Timeline
Window of vulnerability

YOUR SYSTEM **IS NOT PROTECTED** DURING THE WINDOW OF VULNERABILITY!

PROACTIVE SECURITY


Vulnerable software released
Vulnerability announcement
Fix backported & updated released
Timeline
Window of vulnerability is filled by proactive security

PROACTIVE SECURITY HELPS TO **PROTECT** YOUR SYSTEM DURING THE WINDOW OF VULNERABILITY!

SECURITY ENHANCED LINUX IS A SECURITY MECHANISM BRINGING PROACTIVE SECURITY FOR YOUR SYSTEM.

TECHNOLOGY FOR **PROCESS ISOLATION** TO MITIGATE ATTACKS VIA PRIVILEGE ESCALATION

EXPLOIT EXAMPLES WHERE SELINUX HELPED TO PROTECT YOUR SYSTEM

VENOM

VENOM

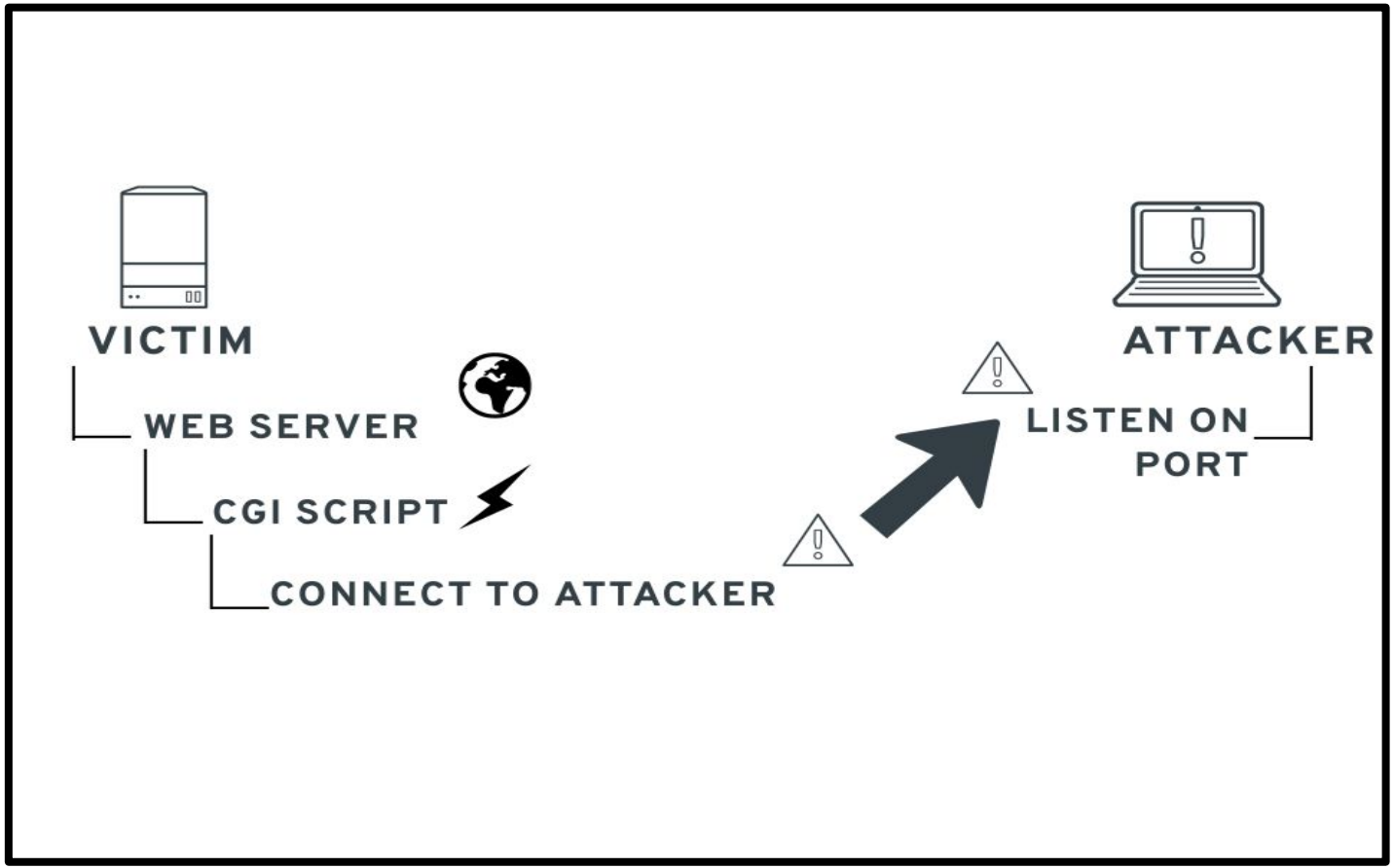
DOCKER CVE-2016-9962

VENOM

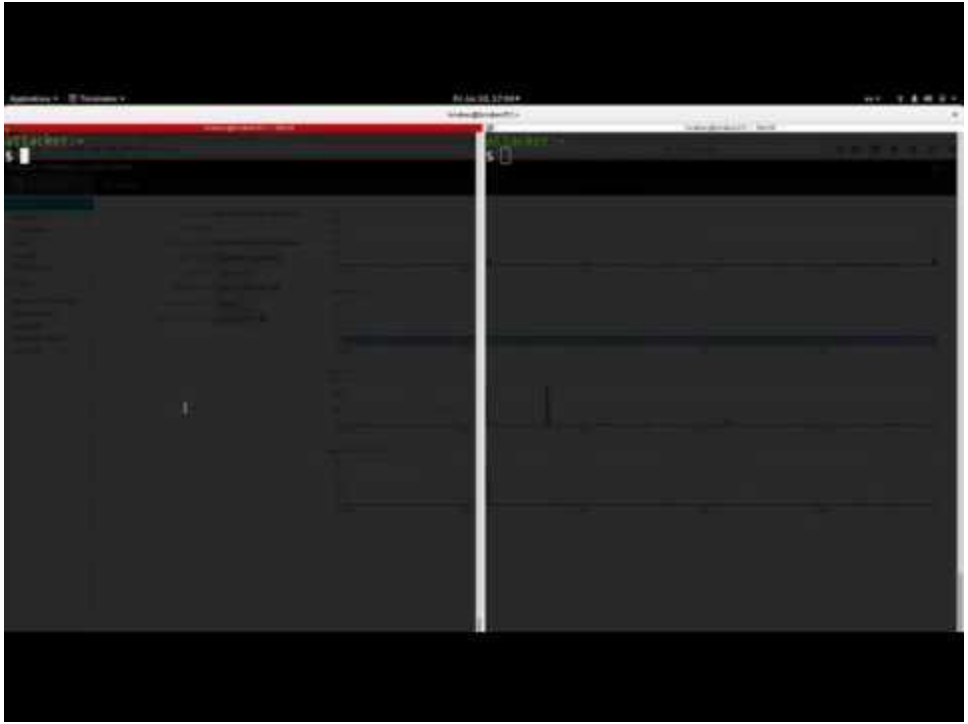
DOCKER CVE-2016-9962

SHELLSHOCK

HACKING TIME!


VICTIM
WEB SERVER
CGI SCRIPT
CONNECT TO ATTACKER
ATTACKER
LISTEN ON PORT

DEMO TIME!



https://www.youtube.com/watch?v=Ysshrh4aGOs

CONCLUSION?


IF YOU RUN LINUX WITH SELINUX DISABLED
INSTRUCTOR
YOUR GONNA HAVE A BAD TIME
memegenerator.net

Traditional Linux Security

$ ls -dl /var/www/html/

drwx r-x r-x. 2 root root /var/www/html/

USER GROUP ALL

```
$ ps -ef | grep NetworkManager

root   11781    1  0 Feb27     00:01:24
/usr/sbin/NetworkManager --no-daemon
```

PROBLEMS

ROOT BYPASSING THIS SECURITY

SETUID BIT

SELinux Security Policy

CORE COMPONENT OF SELINUX

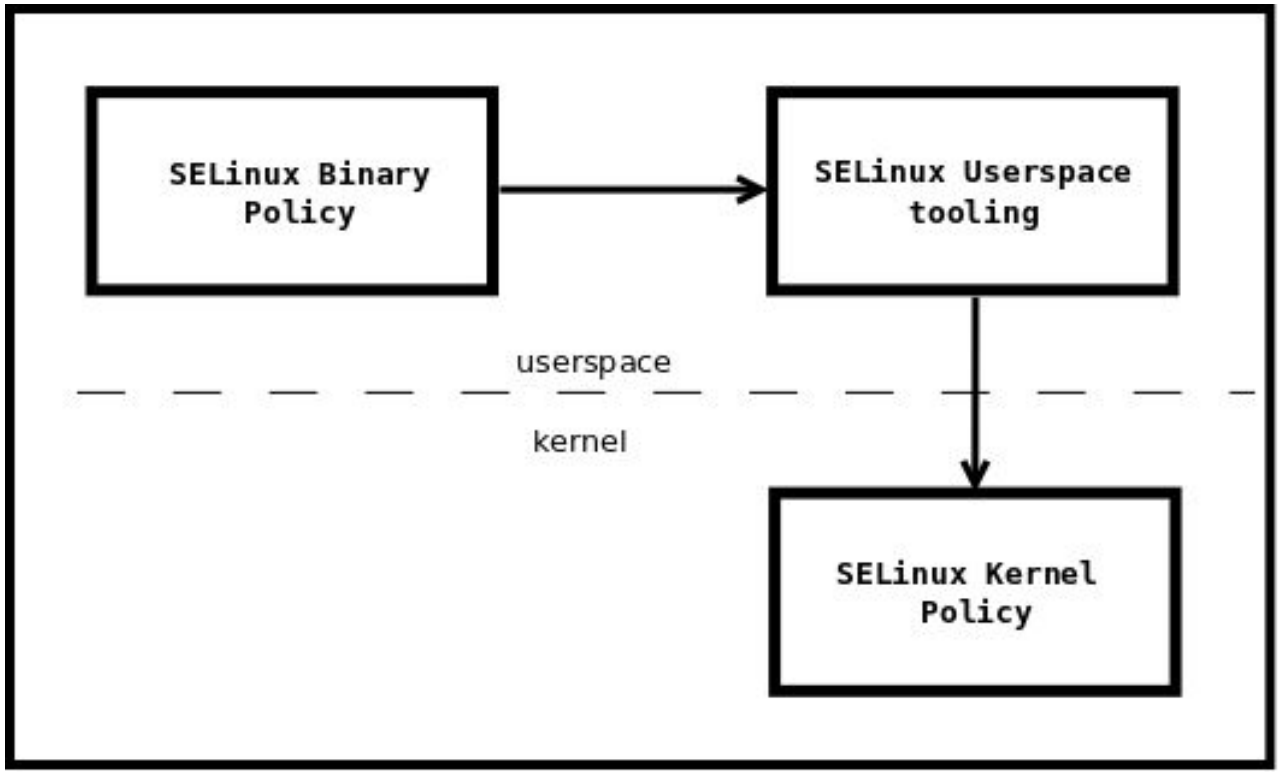
CORE COMPONENT OF SELINUX

COLLECTION OF SELINUX POLICY RULES

CORE COMPONENT OF SELINUX

COLLECTION OF SELINUX POLICY RULES

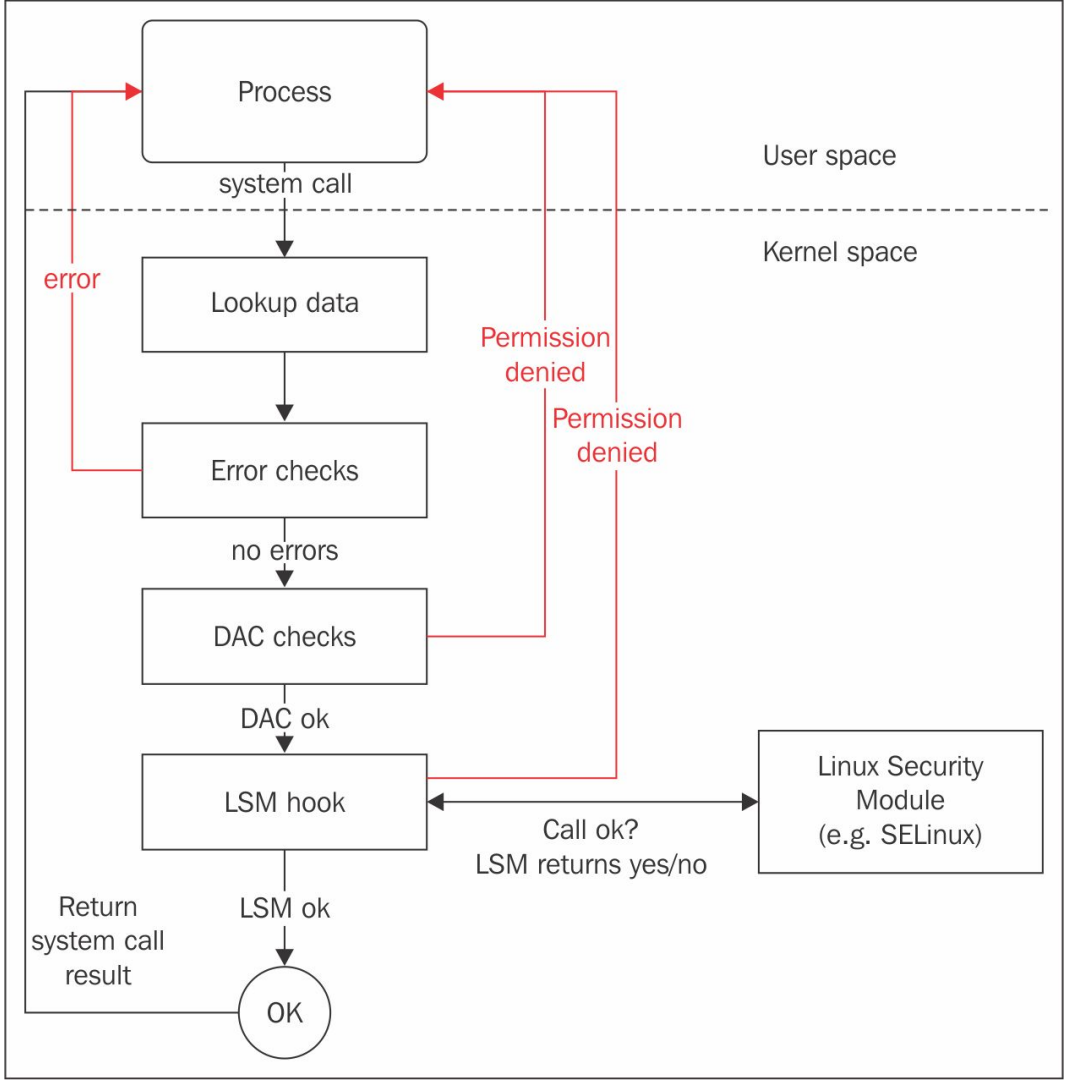
LOADED INTO THE KERNEL BY SELINUX USERSPACE TOOLS


SELinux Binary Policy
SELinux Userspace tooling
userspace
kernel
SELinux Kernel Policy

ENFORCED BY THE KERNEL

ENFORCED BY THE KERNEL

USED TO AUTHORIZE ACCESS REQUESTS ON THE SYSTEM


Process
User space
system call
Kernel space
error
Lookup data
Permission
denied
Permission
denied
Error checks
no errors
DAC checks
DAC ok
LSM hook
Linux Security
Module
(e.g. SELinux)
Call ok?
LSM returns yes/no
Return
system call
result
LSM ok
OK

BY DEFAULT **EVERYTHING** IS DENIED AND YOU DEFINE POLICY RULES TO ALLOW CERTAIN REQUESTS.

SELINUX POLICY RULES

DESCRIBE AN **INTERACTION** BETWEEN PROCESSES AND SYSTEM RESOURCES

SELINUX POLICY RULE IN HUMAN LANGUAGE

"APACHE process can READ its LOGGING FILE"

SELINUX VIEW OF THAT INTERACTION

```
ALLOW apache_process apache_log:FILE
READ;
```

`apache_process` `apache_log`

ARE **LABELS**

LABELS

ASSIGNED TO PROCESSES

ASSIGNED TO PROCESSES

ASSIGNED TO SYSTEM RESOURCES

ASSIGNED TO PROCESSES

ASSIGNED TO SYSTEM RESOURCES

BY SELINUX SECURITY POLICY

ASSIGNED TO PROCESSES

ASSIGNED TO SYSTEM RESOURCES

BY SELINUX SECURITY POLICY

MAP REAL SYSTEM ENTITIES INTO THE SELINUX WORLD

LABELS IN REALITY

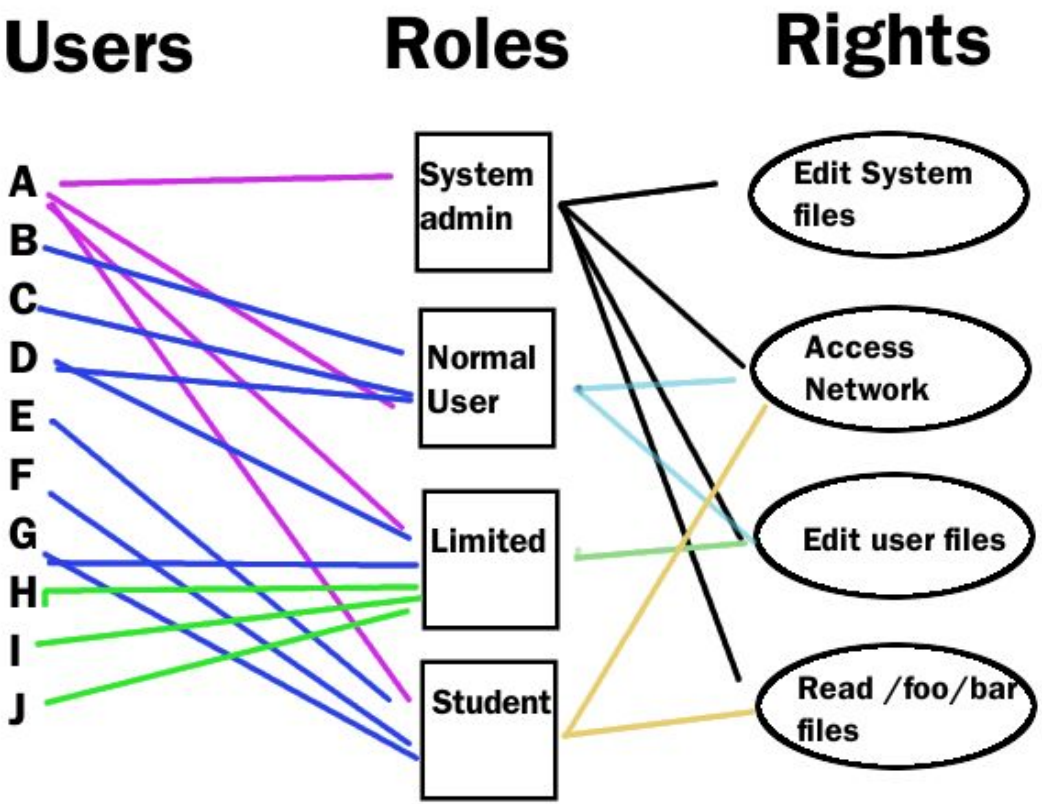
STORED IN EXTENDED ATTRIBUTES OF FILE SYSTEMS - EXT2,EXT3, EXT4 ...

```
# getfattr -n security.selinux /etc/passwd
getfattr: Removing leading '/' from absolute path
names
# file: etc/passwd
security.selinux="system_u:object_r:passwd_file_t:s0"

# ls -Z /etc/passwd
system_u:object_r:passwd_file_t:s0 /etc/passwd
```

SELINUX LABELS CONSIST OF **FOUR** PARTS

<user>:<role>:<type>:<MLS/MCS>


Users
Roles
Rights
A
B
C
D
E
F
G
H
I
J
System admin
Normal User
Limited
Student
Edit System files
Access Network
Edit user files
Read /foo/bar files

<user>:<role>:<type>:<MLS/MCS>

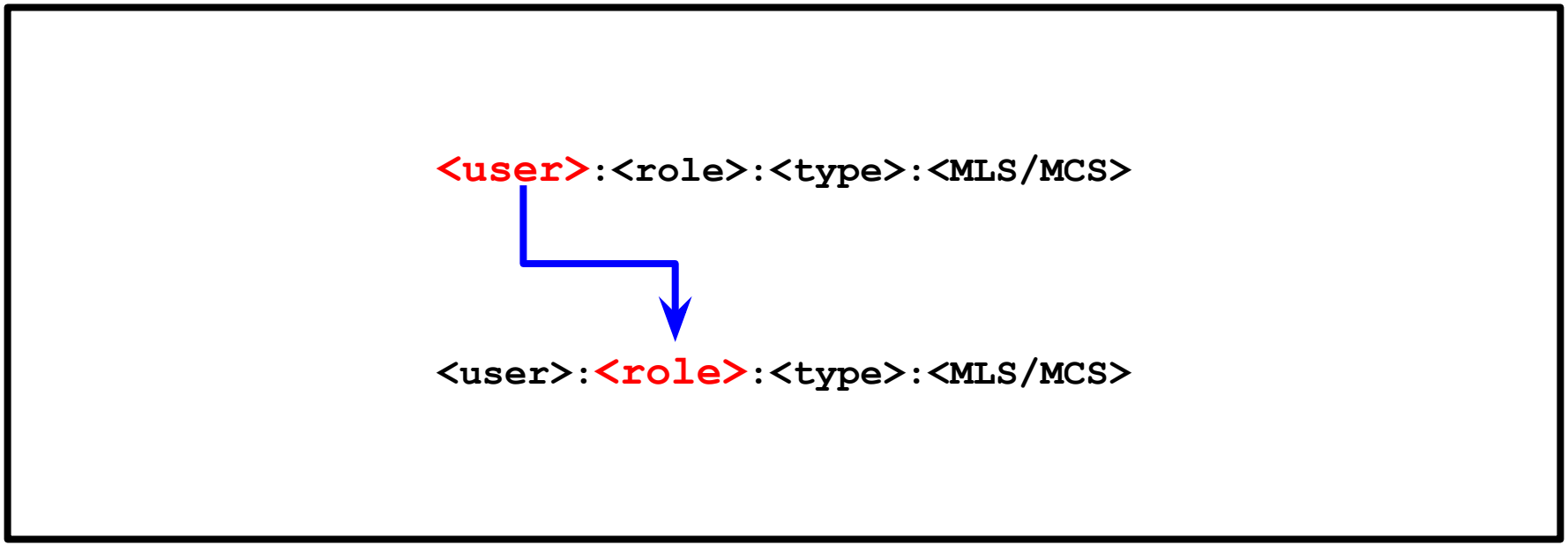
Not the same as Linux users

Several Linux users can be mapped to a single SELinux user

object_u is a placeholder for Linux system resources

system_u is a placeholder for Linux processes

Can be limited to a set of SELinux roles


<user>:<role>:<type>:<MLS/MCS>
<user>:<role>:<type>:<MLS/MCS>

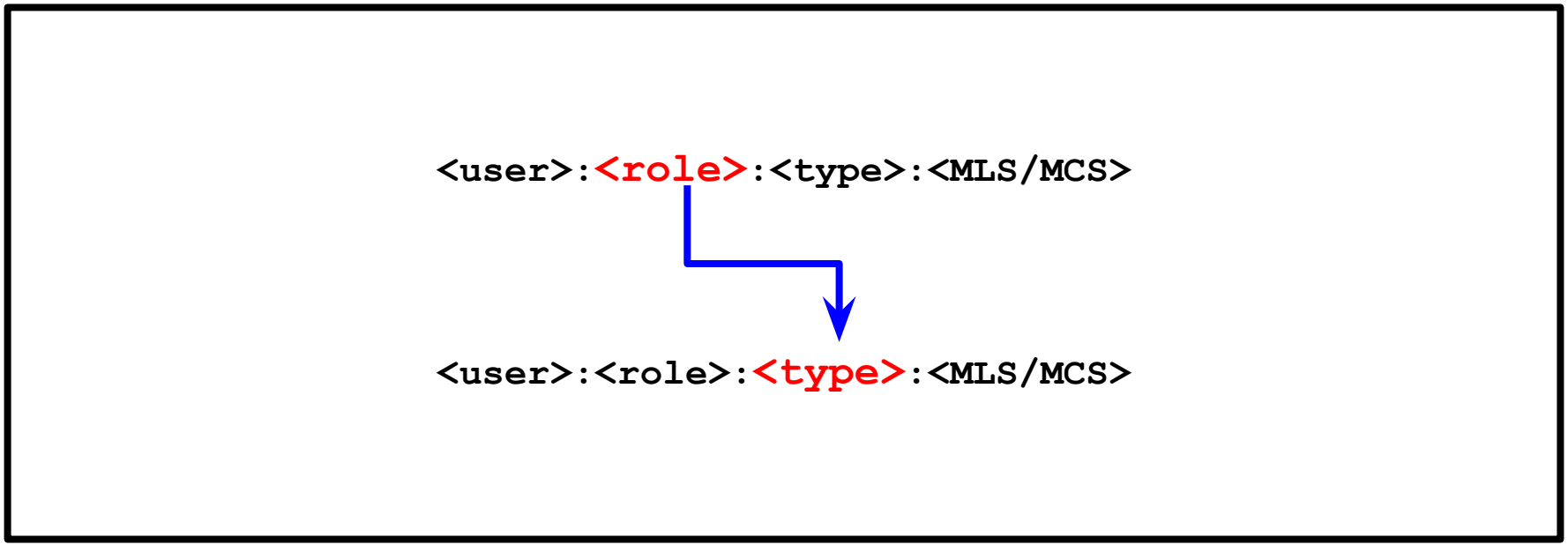
<user>:<role>:<type>:<MLS/MCS>

SELinux users can have multiple roles but only one can be active

object_r is a placeholder for Linux system resources

system_r is a placeholder for system processes

Can be limited to a set of SELinux types


<user>:<role>:<type>:<MLS/MCS>
<user>:<role>:<type>:<MLS/MCS>

<user>:<role>:<type>:<MLS/MCS>

Security model known as **TYPE ENFORCEMENT**

In 99% you care only about TYPES

policy rules and interactions between types

<user>:<role>:<type>:<MLS/MCS>

Multi Level Security

Only the MCS part is used in Targeted Policy with the default *s0* level

Allow users to mark resources with compartment tags *(MCS1*, *MCS2*)

Used for RHEL virtualization and for container security

s0:c1 can not access *s0:c2*

User	Role	Domain	X Window System	su or sudo	Execute in home directory and /tmp/ (default)	Networking
sysadm_u	sysadm_r	sysadm_t	yes	**su** and **sudo**	yes	yes
staff_u	staff_r	staff_t	yes	only **sudo**	yes	yes
user_u	user_r	user_t	yes	no	yes	yes
guest_u	guest_r	guest_t	no	no	no	no
xguest_u	xguest_r	xguest_t	yes	no	no	Firefox only

IN RHEL8 WE SHIP THE **TARGETED** SELINUX POLICY BY DEFAULT

WE MOSTLY CARE ONLY ABOUT **TYPES**

SELINUX **ALLOW** RULE SYNTAX WITH **TYPES**

```
ALLOW TYPE1 TYPE2:OBJECT_CLASS
PERMISSION;
```

```
ALLOW APACHE_T APACHE_LOG_T:FILE READ;
```

DOMAIN TRANSITION RULES

```
TYPE_TRANSITION TYPE1 TYPE2:PROCESS
                NEW_DOMAIN;
```

```
TYPE_TRANSITION INIT_T
HTTPD_EXEC_T:PROCESS HTTPD_T;
```

FILE TRANSITION RULES

```
TYPE_TRANSITION TYPE1 TYPE2:OBJECT_CLASS
NEW_TYPE;
```

```
TYPE_TRANSITION HTTPD_T VAR_LOG_T:FILE
HTTPD_LOG_T;
```

SELINUX MODES

ENFORCING

ENFORCING

SELINUX SECURITY POLICY IS ENFORCED BY KERNEL

PERMISSIVE

PERMISSIVE

SELINUX SECURITY POLICY IS NOT ENFORCED BY KERNEL

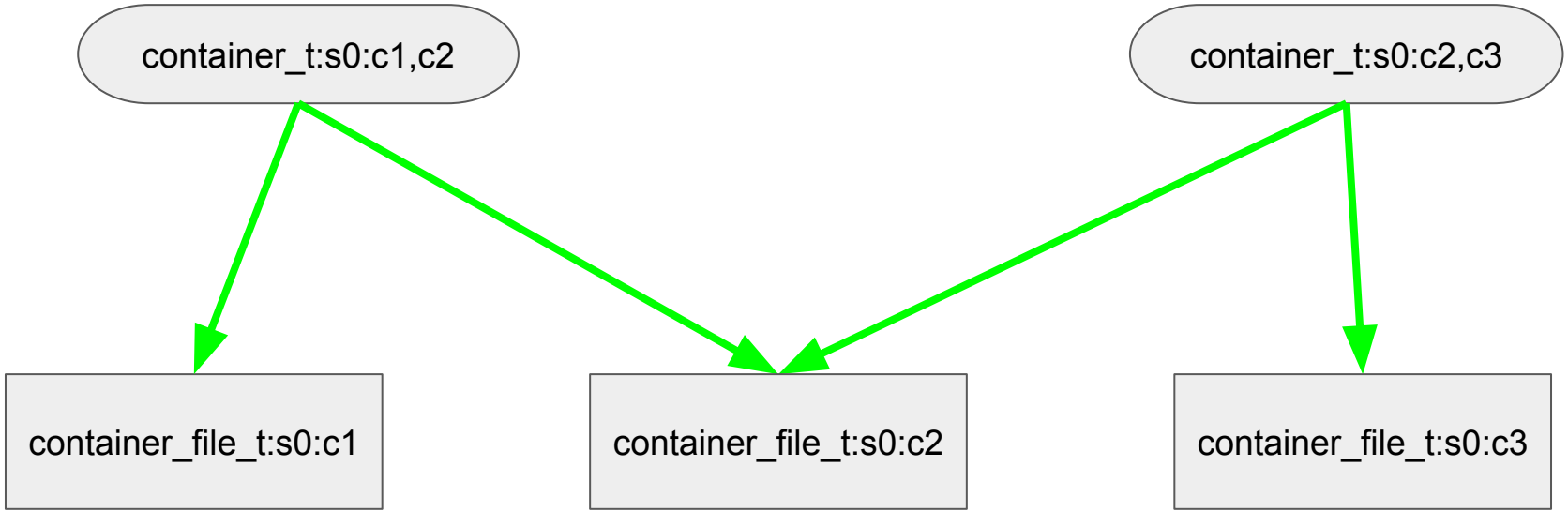
PERMISSIVE

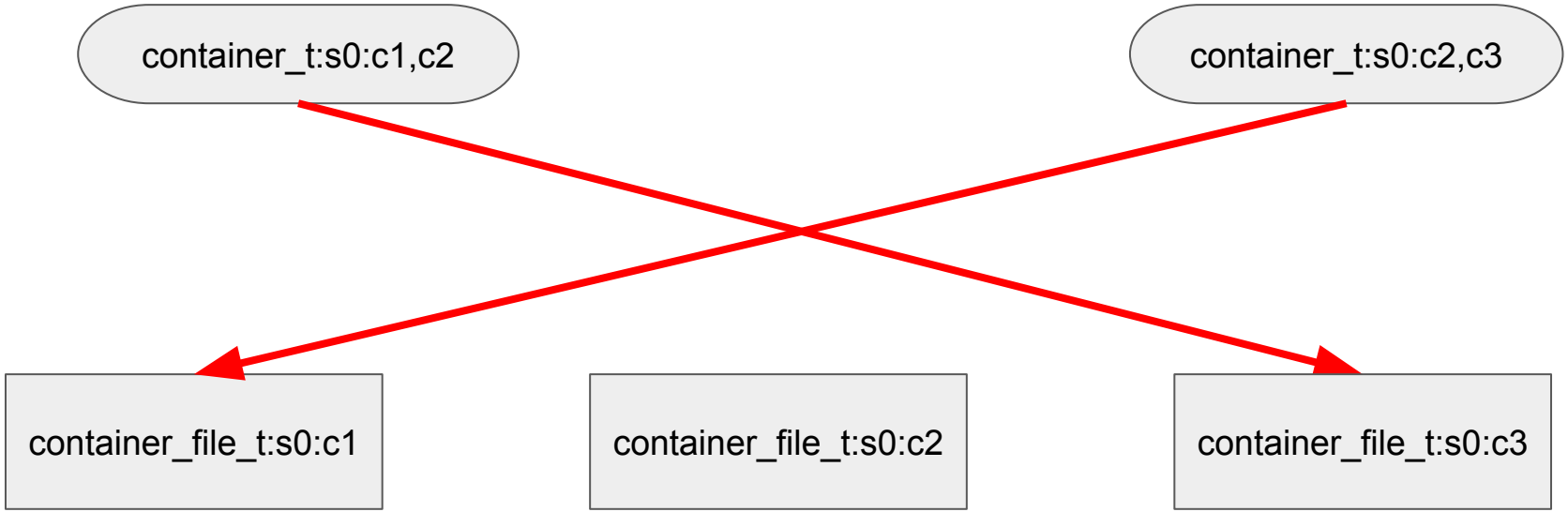
SELINUX SECURITY POLICY IS NOT ENFORCED BY KERNEL

ACCESSES ARE LOGGED

SELINUX VS. CONTAINERS

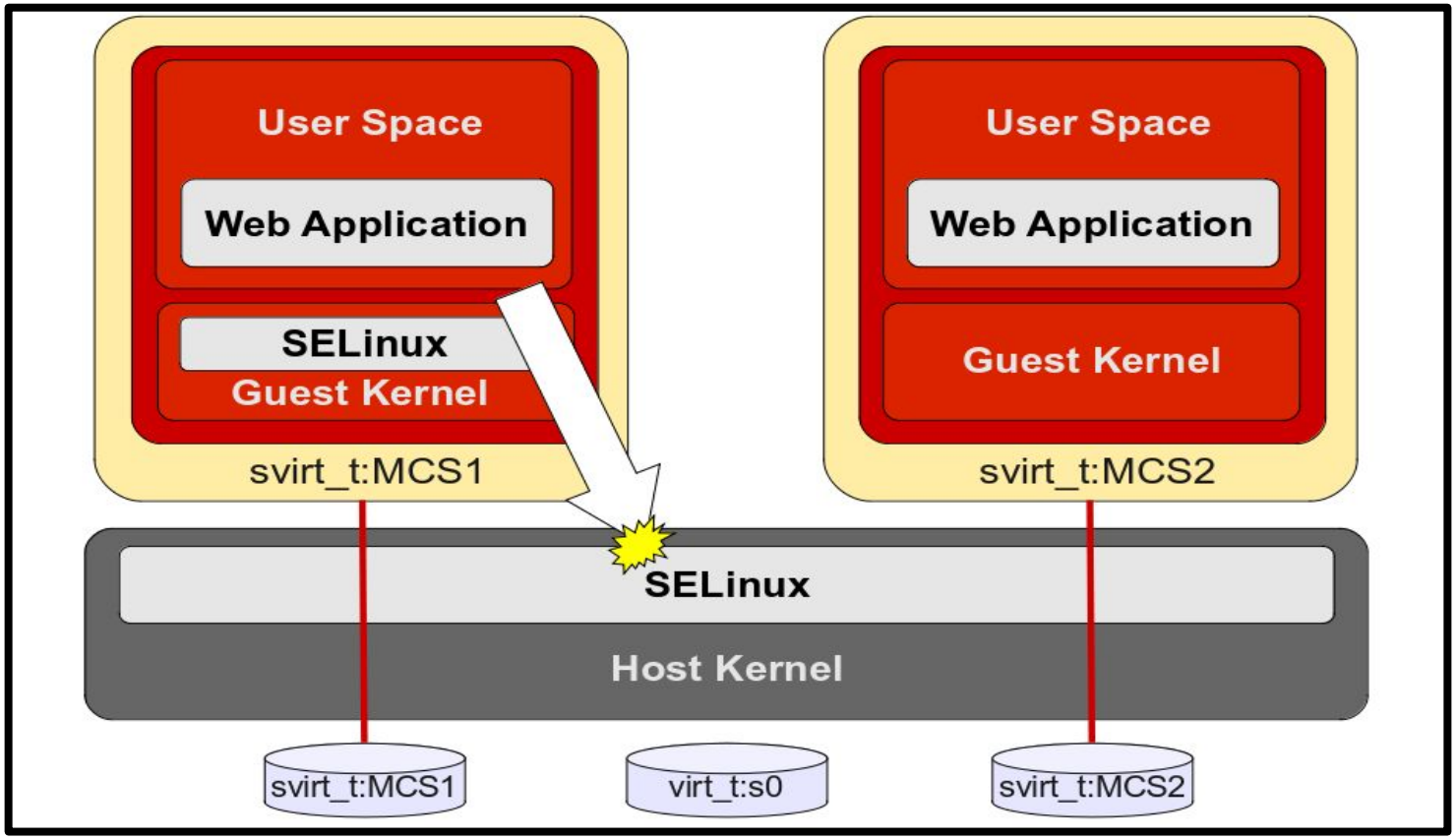
APPLIES MAC TO IMPROVE SECURITY WHEN USING CONTAINERS OR VIRTUAL MACHINES


container_t:s0:c1,c2
container_t:s0:c2,c3
container_file_t:s0:c1
container_file_t:s0:c2
container_file_t:s0:c3


container_t:s0:c1,c2
container_t:s0:c2,c3
container_file_t:s0:c1
container_file_t:s0:c2
container_file_t:s0:c3

Granted access:

- container_t:s0:c1,c2
  - container_file_t:s0
  - container_file_t:s0:c1
  - container_file_t:s0:c2
  - container_file_t:s0:c1,c2
- container_t:s0:c2,c3
  - container_file_t:s0
  - container_file_t:s0:c2
  - container_file_t:s0:c3
  - container_file_t:s0:c2,c3


User Space
Web Application
SELinux
Guest Kernel
svirt_t:MCS1
User Space
Web Application
Guest Kernel
svirt_t:MCS2
SELinux
Host Kernel
svirt_t:MCS1
virt_t:s0
svirt_t:MCS2

SELinux user:SELinux role:SELinux type:SELinux category

SELinux user:SELinux role:SELinux type:SELinux category

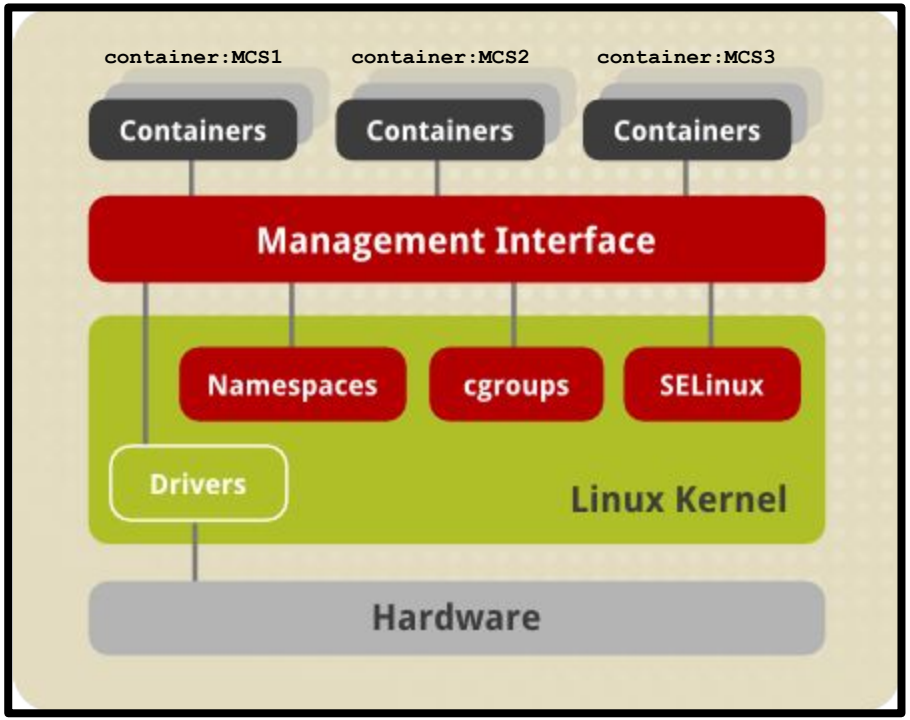
system_u:object_r:svirt_t:c306,c536

SELinux user:SELinux role:SELinux type:SELinux category

system_u:object_r:svirt_t:c306,c536

system_u:object_r:svirt_t:c206,c636

SELINUX KEEPS YOUR CONTAINER IN ITS OWN SPACE


container:MCS1
container:MCS2
container:MCS3
Containers
Containers
Containers
Management Interface
Namespaces
cgroups
SELinux
Drivers
Linux Kernel
Hardware

SELinux user:SELinux role:SELinux type:SELinux category

SELinux user:SELinux role:SELinux type:SELinux category

system_u:object_r:container_t:c306,c536

system_u:object_r:container_t:c206,c636

system_u:object_r:container_t:c406,c736

AVC MESSAGES

WHERE CAN WE FIND LOGS?

```
# cat /var/log/audit/audit.log
```

```
# cat /var/log/audit/audit.log
```

```
# ausearch -m AVC
```

type=AVC **msg**=audit(1226882925.714:136): **avc: denied**
{ **read** } for **pid**=2512 **comm**="httpd" **name**="file1"
dev=dm-0 **ino**=284133
scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:shadow_t:s0
tclass=file

HOW TO PARSE AVC MESSAGES?

```
# ausearch
```

```
# ausearch
# audit2allow
```

```
# ausearch -m AVC -ts recent

type=AVC msg=audit(1226882925.714:136): avc:  denied  { read } for
pid=2512 comm="httpd" name="shadow" dev=dm-0 ino=284133
scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:shadow_t:s0 tclass=file

# ausearch -m AVC -ts recent | audit2allow

#============= httpd_t ==============

allow httpd_t shadow_t:file read;
```

- # semanage fcontext -> manage SELinux contexts
- # semanage boolean -> manage SELinux booleans
- # semanage port -> manage SELinux ports
- # semanage permissive -> put SELinux domain to permissive mode
- # sesearch -> search for present SELinux rules
- # ausearch -> search for SELinux denials
- # sealert -> SELinux troubleshooter
- # audit2allow -> Parse SELinux denials / create local SELinux module
- # semodule -DB / # semodule -B -> SELinux policy rebuild

ARE YOU USING SELINUX IN ENFORCING?

BLOGS

Lukas Vrabec's blog	https://lukas-vrabec.com/
Dan Walsh's blog	http://danwalsh.livejournal.com/
Miroslav Grepl's blog	https://mgrepl.wordpress.com/
Paul Moore's blog	http://www.paul-moore.com/
Petr Lautrbach's blog	https://plautrba.fedorapeople.org/