Proactive Security in Linux

Lukas Vrabec
MFF UK Praha 2017
Agenda

- Proactive Security
- Traditional Linux Security
- SELinux Security Policy
- Updated Userspace with Easier Policy Customization
- SELinux Awareness
- Troubleshooting Existing Policy
Proactive Security
WHEN DO PEOPLE CARE ABOUT SECURITY?
This Hack Will Destroy Your Phone!
HOW DO SECURITY ISSUES AFFECT ME?
LOST/UNWITTINGLY SHARED/COMPROMISED
PERSONAL DATA
LOST/UNWITTINGLY SHARED/COMPROMISED
PERSONAL DATA
ONLINE MONEY-RELATED THEFTS
LOST/UNWITTINGLY SHARED/COMPROMISED PERSONAL DATA
ONLINE MONEY-RELATED THEFTS
MOBILE DEVICE PROTECTION
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

Window of vulnerability
YOUR SYSTEM IS NOT PROTECTED DURING THE WINDOW OF VULNERABILITY!
PROACTIVE SECURITY
Vulnerable software released

Vulnerability announcement

Fix backported & updated released

Window of vulnerability is filled by proactive security

Timeline
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.
EXPLOIT EXAMPLES WHERE SELINUX HELPED TO PROTECT YOUR SYSTEM
VENOM
VENOM

DOCKER CVE-2016-9962
VENOM

DOCKER CVE-2016-9962

SHELLSHOCK
HACKING TIME!
DEMO TIME!
CONCLUSION?
IF YOU RUN LINUX WITH SELINUX DISABLED
YOUR GONNA HAVE A BAD TIME
Traditional Linux Security
```bash
$ ls -dl /var/www/html/
```

```
drwx  r-x  r-x. 2 root root /var/www/html/
  
  USER GROUP ALL
```
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 → SELinux Kernel Policy

Diagram:
- SELinux Binary Policy
- SELinux Userspace tooling
- SELinux Kernel Policy

Relationships:
- SELinux Binary Policy to SELinux Userspace tooling
- SELinux Userspace tooling to SELinux Kernel Policy

Annotations:
- userspace
- kernel
ENFORCED BY THE KERNEL
ENFORCED BY THE KERNEL
USED TO AUTHORIZE ACCESS REQUESTS ON THE SYSTEM
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
Not the same as Linux users
Several Linux users can be mapped to a single SELinux user
\textit{object\textunderscore u} is a placeholder for Linux system resources
\textit{system\textunderscore u} is a placeholder for Linux processes
Can be limited to a set of SELinux roles
<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
Security model known as **TYPE ENFORCEMENT**

In 99% you care only about TYPES

policy rules and interactions between types
<table>
<thead>
<tr>
<th>User</th>
<th>Role</th>
<th>Domain</th>
<th>X Window System</th>
<th>su or sudo</th>
<th>Execute in home directory and /tmp/ (default)</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysadm_u</td>
<td>sysadm_r</td>
<td>sysadm_t</td>
<td>yes</td>
<td>su and sudo</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>staff_u</td>
<td>staff_r</td>
<td>staff_t</td>
<td>yes</td>
<td>only sudo</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>user_u</td>
<td>user_r</td>
<td>user_t</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>guest_u</td>
<td>guest_r</td>
<td>guest_t</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>xguest_u</td>
<td>xguest_r</td>
<td>xguest_t</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>Firefox only</td>
</tr>
</tbody>
</table>
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
IN RHEL7 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;
Updated Userspace with Easier Policy Customization
FRIENDLY SELINUX?
NEW COMMON INTERMEDIATE LANGUAGE - CIL
"M4+COMPILATION" VS. CIL
PERFORMANCE IMPROVEMENTS
PERFORMANCE IMPROVEMENTS

NEW POSSIBILITY FOR HLL
PERFORMANCE IMPROVEMENTS
NEW POSSIBILITY FOR HLL
USABILITY
LOCAL POLICY IN TWO STEPS
“I have an apache process that needs to access its log file. I would like to add SELinux policy rule reflecting the following interaction so that I am able to read important info from the apache logging file.”
# cat myapache.cil

(allow httpd_t httpd_log_t (file (open read getattr)))
# semodule -i myapache.cil
HOW DO WE DO IT WITH M4 + COMPILATION?
# cat myapache.te

require {
  type httpd_t;
  type httpd_log_t;
}

allow httpd_t httpd_log_t:file { open read getattr };
# make -f /usr/share/selinux/devel/Makefile
# semodule -i myapache.pp
SELinux Awareness
SELINUX ENHANCING TECHNOLOGIES
SYSTEMD
SYSTEMD WORKS AS AN SELINUX ACCESS MANAGER
DO YOU REMEMBER SELINUX ALLOW RULE SYNTAX WITH TYPE ENFORCEMENT?
ALLOW TYPE1 TYPE2:OBJECT_CLASS
PERMISSION;
ALLOW HTTPD_T HTTPD_UNIT_FILE_T:SERVICE START;
SYSTEMD

SVIRT
APPLIES MAC TO IMPROVE SECURITY WHEN USING VIRTUAL MACHINES
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
SYSTEMD
SVIRT
CONTAINERS
SELINUX KEEPS YOUR CONTAINER IN ITS OWN SPACE
SELinux user:SELinux role:SELinux type:SELinux category
SELinux user:SELinux role:SELinux type:SELinux category
system_u:object_r:container_t:c306,c536
SELinux user:SELinux role:SELinux type:SELinux category

system_u:object_r:container_t:c306,c536
system_u:object_r:container_t:c206,c636
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
SELinux Modes
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
ACCESSSES ARE LOGGED
Troubleshooting Existing Policy
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?
# sesearch
# sesearch

# 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;
MISLABELED SYSTEM I.E LABELS ON OBJECTS ARE WRONG
# restorecon -Rv /

or

# fixfiles onboot

# reboot
SELINUX MODIFICATIONS OF THE DISTRO POLICY
VIA SELINUX USERSPACE TOOLING
SOMETIMES IT’S NOT NECESSARY TO CREATE CUSTOM SELINUX POLICY, LOCAL MODIFICATION CAN FIX IT.
APACHE HTTP SERVER WITH CHANGES IN THE DEFAULT CONFIGURATION
httpd service configured to listen on tcp port 3131 instead of port 80
httpd service configured to listen on tcp port 3131 instead of port 80

document root will be /var/test-www/ instead of /var/www/
Change in /etc/httpd/conf/httpd.conf

```bash
# sed -i 's_Listen 80_Listen 3131_' /etc/httpd/conf/httpd.conf


```
# systemctl restart httpd

# ausearch -m AVC -ts recent

....
httpd service trying to bind on port 3131 instead of 80, this should be changed in SELinux policy

```
type=AVC msg=audit(1491948261.488:599): avc: denied { name_bind } for pid=5920 comm="httpd" src=3131 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:unreserved_port_t:s0 tclass=tcp_socket
```
# sesearch -A -s httpd_t -t http_port_t -c tcp_socket -p name_bind
Found 1 semantic av rules:
   allow httpd_t http_port_t : tcp_socket name_bind;

# semanage port -a -t http_port_t -p tcp 3131
# semanage port -l | grep http_port_t
http_port_t
   tcp    3131, 80, 81, 443, 488, 8008, 8009, 8443, 9000
httpd service DocumentRoot is in /var/test_www/html and this directory has wrong label

type=AVC msg=audit(1491949594.146:622): avc: denied { read }
  for  pid=6094 comm="httpd" name="index.html" dev="dm-0"
    ino=13485999 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:var_t:s0 tclass=file
# matchpathcon /var/test_www/html/index.html
/var/test_www/html/index/html  system_u:object_r:var_t:s0

# matchpathcon /var/www/html/index.html
/var/www/html/index.html  system_u:object_r:httpd_sys_content_t:s0

# semanage fcontext -a -t httpd_sys_content_t "/var/test_www(/.*)?"
# restorecon -Rv /var/
# semanage fcontext -l | grep httpd_sys_content_t | grep www
/var/www(/.*)? all files
system_u:object_r:httpd_sys_content_t:s0
/var/test_www(/.*)? all files
system_u:object_r:httpd_sys_content_t:s0
....
....
HTTPD SERVICE SELINUX DENIALS ARE FIXED WITHOUT WRITING CUSTOM POLICY!
QUESTIONS?

Miroslav Grepl’s blog  https://mgrepl.wordpress.com/
Paul Moore’s blog     http://www.paul-moore.com/
Lukas Vrabec’s blog   https://lvrabec-selinux.rhcloud.com/
Dan Walsh’s blog      http://danwalsh.livejournal.com/
THANK YOU