Anna Plust, 36148

Zaufana Infrastruktura Obliczeniowa

Laboratorium 6, Sprawozdanie

SELinux - Polityki kontroli dostępu (Linux)

Wstęp.

Sprawdzanie i weryfikacja polityk dostępu w systemie SELinux jest ważnym krokiem w zapewnianiu bezpieczeństwa i ochrony danych. Dzięki temu można upewnić się, czy polityki są odpowiednio skonfigurowane i spełniają swoje zamierzone cele. Sprawdzanie polityk SELinux obejmuje analizę reguł, kontekstów bezpieczeństwa, uprawnień dostępu oraz audytowanie zdarzeń związanych z działaniem systemu.

Podczas sprawdzania polityk można stosować różne narzędzia, np audit2allow, które pomagają analizować i poprawiać błędy w politykach. Weryfikacja polega na analizie logów i komunikatów AVC oraz innych informacji diagnostycznych, które mogą wskazywać na nieprawidłowości lub potencjalne zagrożenia.

Krótki opis ćwiczenia.

Ćwiczenie dotyczy tworzenia i instalowania polityk dostępu w SELinux oraz obsługi SELinux i budowy bezpiecznych systemów operacyjnych.

Zadania praktyczne.

Na początku zmieniono tryb SELinux na Permissive.

```
[root@fedora home]# setenforce Permissive
[root@fedora home]# getenforce
Permissive
```

W drugiej konsoli zalogowano się na konto użytkownika bob. Wymagało to wystartowania serwera sshd:
`service sshd start`

Identyfikatory użytkownika bob to:

```
[bob@fedora ~]$ id -Z
user_u:user_r:user_t:s0
```

Sprawdzono czy zmienna logiczna user_ping jest ustawiona na off

```
[root@fedora home]# getsebool user_ping
selinuxuser_ping --> on
[root@fedora home]# setsebool -P user_ping off
[root@fedora home]# getsebool user_ping
selinuxuser_ping --> off
```

Sprawdzono ping z terminalu użytkownika bob. Wywołanie komendy powiodło się.

```
[bob@fedora ~]$ ping -c 4 google.com
PING google.com (142.250.203.142) 56(84) bytes of data.
64 bytes from waw07s06-in-f14.1e100.net (142.250.203.142): icmp_seq=1 ttl=113 time=31.5 ms
64 bytes from waw07s06-in-f14.1e100.net (142.250.203.142): icmp_seq=2 ttl=113 time=32.8 ms
64 bytes from waw07s06-in-f14.1e100.net (142.250.203.142): icmp_seq=3 ttl=113 time=30.4 ms
64 bytes from waw07s06-in-f14.1e100.net (142.250.203.142): icmp_seq=4 ttl=113 time=32.9 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3017ms
rtt min/avg/max/mdev = 30.363/31.916/32.924/1.052 ms
```

1. **Zinterpretuj wynik polecenia wywołanego w konsoli osboxes: sudo ausearch -c "ping".**

```
[root@fedora home]# sudo ausearch -c "ping"
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.578:418): avc:  denied  { create } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.619:419): avc:  denied  { setopt } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.619:420): avc:  denied  { getopt } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
```

Wyłączenie uprawnień SELinux dotyczących pingowania (user_ping) przez użytkownika "admin" nie będzie miało wpływu na działanie polecenia ping wykonywanego przez użytkownika "bob". Jednak niepożądane działanie zostało zalogowane.

Czy SELinux umożliwiłby użytkownikowi w domenie user_t wykonanie polecenia ping, gdyby był włączony tryb enforcing?
SELinux w trybie enforcing domyślnie nie umożliwiłby użytkownikowi w domenie user_t wykonania polecenia ping.

2. **Czy udało się wykonać polecenie ping?**

Nie, polecenie nie powiodło się.

```
[bob@fedora ~]$ ping -c 4 google.com
[bob@fedora ~]$
```

3. **Czy pojawiły się jakieś nowe logi?**

Tak, pojawiły się logi o zablokowaniu wykonania komendy ping.

```
[root@fedora home]# sudo ausearch -c "ping"
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.578:418): avc:  denied  { create } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.619:419): avc:  denied  { setopt } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
----
time->Sat Jun 17 15:54:54 2023
type=AVC msg=audit(1687010094.619:420): avc:  denied  { getopt } for  pid=4090 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=1
----
time->Sat Jun 17 15:58:28 2023
type=AVC msg=audit(1687010308.469:428): avc:  denied  { create } for  pid=4184 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=0
----
time->Sat Jun 17 15:58:28 2023
type=AVC msg=audit(1687010308.469:429): avc:  denied  { create } for  pid=4184 comm="ping" scontext=user_u:user_r:user
_t:s0 tcontext=user_u:user_r:user_t:s0 tclass=icmp_socket permissive=0
```

4. **O czym świadczy uzyskany wynik?**

Gdy enforcing jest ustawiony na permissive, zabronione akcje są możliwe do wykonania, ale są zapisywane w logach. Gdy enforcing jest ustawiony na enforcing, zabronione akcje są blokowane i logowane.

Ćwiczenie 2. Role użytkowników

Utworzono nowego użytkownika o nazwie alice i przyporządkowano go fo grupy użytkowników SELinux o nazwie staff_u.

5. **W jakim kontekście działa użytkownik alice (wynik polecenia id -Z)?**

```
[alice@fedora ~]$ id -Z
staff_u:staff_r:staff_t:s0-s0:c0.c1023
```

Następnie rola użytkownika alice została zmieniona na sysadm_r za pomocą komendy
`$ newrole -r sysadm_r.`

6. **W jakim kontekście działa użytkownik alice po zmianie roli?**

```
[alice@fedora ~]$ id -Z
staff_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

Użytkownik został przypisany do roli sysadm_r, ponieważ jego wcześniejsza rola była przechodnia.

W konsoli użytkownika bob dokonano próby przełączenia jego roli na sysadm_r.

7. **Jaki jest rezultat próby zmiany roli?**

```
[bob@fedora ~]$ newrole -r sysadm_r
user_u:sysadm_r:sysadm_t:s0 is not a valid context
```

Użytkownik bob nie został przypisany do roli sysadm_r, ponieważ jest przypisany do user_r. Byłoby to możliwe tylko, gdy użytkownik jest przypisany do roli staff_r, która jest rolą przechodnią.

Użytkownik alice został dodany do grupy użytkowników z dostępnym poleceniem sudo:
`$sudo usermod -aG wheel alice`

8. **W konsoli osboxes przejdź za pomocą sudo -i na użytkownika alice. Sprawdź jaki mamy kontekst ochrony (id -Z)? Za pomocą exit wróć do użytkownika osboxes.**

```
[alice@fedora root]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Dodano nowego użytkownika SELinux mystaff_u z wybranymi rolami użytkownika staff_u:
`$ sudo semanage user -a -R "staff_r sysadm_r system_r" -r "s0-s0:c0-c1023" mystaff_u`

Spróbuj zmienić mapowanie użytkownika *alice* ze staff_u na mystaff_u.

```
[root@fedora ~]# semanage login -a -s mystaff_u alice
[root@fedora ~]# semanage login -l

Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
alice                mystaff_u            s0-s0:c0             *
bob                  user_u               s0                   *
root                 unconfined_u         s0-s0:c0.c1023       *
```

Ćwiczenie 3. Tworzenie własnej polityki bezpieczeństwa

Utworzono plik, który definiuje nowy typ topsecret_t. Zgodnie z powyższą polityką, użytkownik o typie user_t będzie miał prawa do odczytywania atrybutów z plików opatrzonych typem topsecret_t. Został on skompilowany, a polityka zainstalowana.

```
[root@fedora bob]# seinfo --type=topsecret_t

Types: 1
   topsecret_t
```

Ćwiczenie 4. Aktualizacja polityki bezpieczeństwa

W katalogu domowym użytkownika bob utworzono plik o nazwie secret_file i zapisano do niego niepusty ciąg znaków.

```
[root@fedora bob]# cat secret_file
123456789
```

Właściciel pliku został zmieniony, a także zmieniona została jego etykieta.

9. **Jaką etykietę ma plik secret_file?**

Etykieta została utajniona.

```
[root@fedora bob]# ls -laZ /home/bob
ls: cannot access '/home/bob/secret_file': Permission denied
total 12
drwx------. 1 bob  bob  user_u:object_r:user_home_dir_t:s0 102 Jun 17 16:44 .
drwxr-xr-x. 1 root root system_u:object_r:home_root_t:s0    24 Jun 17 16:03 ..
-rw-r--r--. 1 bob  bob  user_u:object_r:user_home_t:s0      18 Sep 27  2022 .bash_logout
-rw-r--r--. 1 bob  bob  user_u:object_r:user_home_t:s0     141 Sep 27  2022 .bash_profile
-rw-r--r--. 1 bob  bob  user_u:object_r:user_home_t:s0     492 Sep 27  2022 .bashrc
drwxr-xr-x. 1 bob  bob  user_u:object_r:mozilla_home_t:s0   34 Nov  5  2022 .mozilla
-?????????? ? ?    ?    ?                                    ?            ? secret_file
```

Wynik polecenia:
`$ ls -lZ secret_file`

```
[bob@fedora ~]$ ls -lZ secret_file
-rw-r--r--. 1 bob root system_u:object_r:topsecret_t:s0 10 Jun 17 16:45 secret_file
```

Próba odczytania zawartości pliku, kończy się niepowodzeniem.

```
[bob@fedora ~]$ cat secret_file
cat: secret_file: Permission denied
```

10. **Zinterpretuj wynik polecenia szukając komunikatu dotyczącego ostatniego polecenia (konsola osboxes): $ sudo ausearch -m AVC,USER_AVC -ts recent**

```
time->Sat Jun 17 16:54:44 2023
type=AVC msg=audit(1687013684.414:721): avc:  denied  { read } for  pid=6872 comm="cat" name="secret_file" dev="sda3"
ino=1028 scontext=user_u:user_r:user_t:s0 tcontext=system_u:object_r:topsecret_t:s0 tclass=file permissive=0
```

Log informuje o odrzuceniu żądania odczytu o nazwie "secret_file". Zostało ono odrzucone zgodnie z zasadami bezpieczeństwa SELinux, ponieważ kontekst bezpieczeństwa pliku "topsecret_t" nie pasuje do kontekstu bezpieczeństwa procesu "user_t". Jako że SELinux działa w trybie Enforcing, to takie żądania są blokowane.

11. **Wypreparuj odnaleziony log (np. używając w ausearch przełącznika -c) i zapisz go w pliku error.txt.**

```
[root@fedora bob]# ausearch  -c cat > error.txt
[root@fedora bob]# cat error.txt
----
time->Sat Jun 17 16:54:44 2023
type=AVC msg=audit(1687013684.414:721): avc:  denied  { read } for  pid=6872 comm="cat" name="secret_file" dev="sda3"
ino=1028 scontext=user_u:user_r:user_t:s0 tcontext=system_u:object_r:topsecret_t:s0 tclass=file permissive=0
```

Za pomocą polecenia:
$ audit2allow -M topsecretread < error.txt
utworzono nowy plik z polityką, na bazie komunikatów o odmowie dostępu.

12. **Jaka nowa reguła została zdefiniowana w pliku polityki?**

Nowa reguła to `allow user_t topsecret_t:file read`. Pozwala ona użytkownikowi z etykietą user_t czytać pliki z etykietą topsecret_t.

```
[root@fedora bob]# cat topsecretread.te

module topsecretread 1.0;

require {
        type topsecret_t;
        type user_t;
        class file read;
}

#============= user_t ==============
allow user_t topsecret_t:file read;
```

Nowa polityka została zainstalowana.

```
[root@fedora bob]# semodule -l | grep topsecret
topsecret
topsecretread
```

Jednak to nadal nie pozwoliło użytkownikowi *bob* odczytać zawartości pliku. Należało rozszerzyć politykę nie tylko o komendę *read*, ale też i o komendę *open*.

```
[root@fedora bob]# cat topsecretread.te

module topsecretread 1.0;

require {
        type topsecret_t;
        type user_t;
        class file { open read };
}

#============= user_t ==============

#!!!! This avc is allowed in the current policy
allow user_t topsecret_t:file read;
allow user_t topsecret_t:file open;
```

Po zainstalowaniu nowej polityki, użytkownik bob jest wstanie odczytać zawartość pliku.

```
[bob@fedora ~]$ cat secret_file
cat: secret_file: Permission denied
[bob@fedora ~]$ cat secret_file
cat: secret_file: Permission denied
[bob@fedora ~]$ cat secret_file
123456789
```

13. **Czy w logach systemowych pojawiły się nowe błędy wynikające z braku praw odczytu do obiektów typu topsecret_t?**

Po zainstalowaniu kolejnej polityki, nowe błędy nie pojawiają się.

14. **Ponownie wywołaj polecenie, które sprawdzi czy użytkownik typu user_t ma prawa odczytu obiektów klasy plik o typie topsecret_t.**

Tak, użytkownik bob jest w stanie odczytać zawartość pliku.

Ćwiczenie 5. Tworzenie zaawansowanych polityk

15. **Jaki jest wynik polecenia: $ ls -lZ a.out**

Etykieta SELinux dla pliku "a.out" w powyższym wyniku to
`unconfined_u:object_r:topsecret_exec_t:s0`

16. **W jakiej domenie działa proces a.out? Co się stało?**

Proces a.out działa w domenie topsecret_t, która jest zdefiniowana w poprzednio zainstalowanej polityce topsecret.

Proces a.out uruchomiony przez użytkownika bob działa w domenie topsecret_t według polityki SELinux. To zapewnia mu dostęp jedynie do uprzywilejowanych operacji i obiektów oznaczonych etykietą topsecret_t, co zwiększa bezpieczeństwo i ochronę danych w systemie.

Podsumowanie i wnioski.

SELinux jest ważnym narzędziem do kontroli dostępu w systemach operacyjnych, umożliwiające tworzenie i zarządzanie politykami bezpieczeństwa. Zrozumienie modułów bezpieczeństwa LSM i architektury SELinux jest ważne dla skutecznego tworzenia polityk dostępu. Dzięki kompilacji polityki, zmianie etykiet plików i uruchamianiu procesów można wprowadzać spersonalizowane polityki bezpieczeństwa w SELinux. Analiza logów systemowych i komunikatów AVC jest kluczowa do zinterpretowania logów SELinux i rozwiązywania problemów z dostępem.