SOWID – UŜYTKOWNICY I PRAWA DOSTęPU

PAWEŁ JÓZIAK
WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH
POLITECHNIKI WARSZAWSKIEJ

- Nigdy nie wykonuj poleceń metodą *copy&paste*. Przepisz je ręcznie!
- Nie redukuj pracy nad zadaniem do przeczytania go. Wykonaj je i upewnij się, że rozumiesz jak i dlaczego wynik jest taki, jak na ekranie. Zajrzyj do prezentacji bądź zapytaj prowadzącego, jeśli coś nie będzie jasne.
- Linia zaczynająca się od $ reprezentuje prompt zwykłego użytkownika (nie przepisuj go!). Komendą jest to, co następuje po nim.
- Ostrożnie ze spacjami: nie dodawaj ich więcej ani nie usuwaj takowych, jeśli są!

1. Użytkownicy i bazy tożsamości

(a) Sprawdź swoje UID, GID i przynależność do grup innych niż podstawowa:
`$ id`

(b) Zajrzyj do lokalnej bazy tożsamości:
`$ less /etc/passwd`
Jaki jest katalog domowy `roota`?
Jakich powłók używa się dla kont systemowych?

(c) Przejrzyj wszystkie prawidłowe powłoki na swoim systemie:
`$ cat /etc/shells`

(d) Sprawdź syntaks `'/etc/shadow'`:
`$ man 5 shadow`
Co dzieje się, jeśli zaszyfrowane hasło zaczyna się od `'!'` albo `'*'`?
Jakie polityki zmiany haseł można ustawić w Linuksie?

(e) To samo hasło w postaci stringowej nie musi oznaczać tej samej postaci zaszyfrowanej. Dlaczego? Poczytaj o parametrze *salt*:
`$ man 3 crypt`

(f) Zajrzyj na stronę projektu `fail2ban`: `https://www.fail2ban.org/`. Dziś jest to standardowe rozwiązanie.

(g) Wygeneruj zestaw silnych losowych haseł:
`$ pwgen -s`
albo jedno takie hasło:
`$ pwgen -sN1`
Zajrzyj do strony podręcznika:
`$ man pwgen`

(h) Zajrzyj do dokumentacji komendy pozwalającej na interaktywną zmianę hasła:
`$ man passwd`
Nie używaj go do zmiany **swojego** hasła! Tożsamość Twojego konta jest uzyskiwana z zewnętrznej bazy danych o innym algorytmie szyfrowania, więc zmiana będzie nieskuteczne (a jeśli – odpukać – skuteczna, może to skutkować wyprodukowaniem niewłaściwego hasła).

(i) Zajrzyj do dokumentacji komendy pozwalającej na batchową zmianę haseł:
`$ man chpasswd`
Bardzo wygodne dla administratorów!

(j) Zajrzyj do swojej lokalnej bazy grup:
`$ less /etc/group`
Wśród tych istotnych warto zwrócić uwagę na `wheel`, `wireshark` oraz `uucp`.

(k) Zajrzyj do dokumentacji PAM: `$ man 8 pam`
Jakie są możliwe zadania autentyfikacyjne?

(l) Zajrzyj do paru plików konfiguracyjnych PAM, leżących w `'/etc/pam.d/'` (większość zawiera `'/etc/pam.d/system-auth'`):
```
$ ls /etc/pam.d/
$ less /etc/pam.d/system-auth
```
Zwróć uwagę, że nasze komputery są ustawione na korzystanie z `'pam_ldap'` oraz `'pam_group'` do większości zadań. Pierwsze zadanie oznacza możliwość autentyfikacji względem zewnętrznego serwera za pomocą protokołu LDAP. Ten drugi oznacza możliwość dynamicznego przypisu do grup podczas logowania.
Lightweigth Directory Access Protocol (LDAP) wykracza poza ramy tego kursu. Konfiguracja `'pam_group'` jest jednak prosta:
```
$ less /etc/security/group.conf
```
albo, dla lepszej czytelności:
```
$ grep -vP '^[ \t]*(#|$)' /etc/security/group.conf
```
(uwaga: wiele modułów PAM ma pliki konfiguracyjne `'/etc/security/*.conf'`)

(m) Wykonaj
```
$ id
```
oraz
```
$ id twoj_sasiad
```
Skąd taka różnica? Moduł `'pam_group'` jest aktywny tylko dla zalogowanych użytkowników.

(n) Sprawdź zawartość pliku `'/etc/nsswitch.conf'`:
```
$ less /etc/nsswitch.conf
```
i zwróć uwagę, że dwa źródła tożsamości są zdefiniowane dla użytkowników i grup – lokalne pliki oraz zewnętrzne katalogi LDAP. Aby listować użytkowników i grupy z obu źródeł łącznie, użyj:
```
$ getent passwd
$ getent group
```

(o) Sprawdź swój zestaw zmiennych środowiskowych:
```
$ env
```
Przeloguj się jako inny użytkownik (*uszatekm*): `$ su uszatekm`
```
$ id
$ env
$ exit
```
Czym różnią się zmienne środowiskowe?
Powtórz to z przeładowaniem środowiska przy logowaniu:
```
$ su - uszatekm
$ id
$ env
$ exit
```

(p) Wspomnieliśmy, że grupa `'wheel'` może być ważna dla niektórych zadań. Jakich? Zajrzyj do:
```
$ less /etc/pam.d/su
```

(q) Sprawdź wartości domyślne dla komendy `'useradd'`:
```
$ cat /etc/default/useradd
```
W sumie, przy tej okazji warto też zajrzeć do dokumentacji i trochę poczytać:
```
$ man useradd
```

2. Prawa dostępu do plików

Stacje robocze naszych laboratoriów opierają się o zdalnie montowany system plików. Dwukierunkowa synchronizacja może nie być natychmiastowa, więc jakiekolwiek zmiany mogą być widoczne u sąsiada po paru sekundach (w złych przypadkach: nawet 20-30!).

(a) Utwórz pusty plik:
`$ touch plik2a`
Wylistuj wszystkie grupy do których należysz:
`$ id -Gn`
Zmień grupę tego pliku na jakąkolwiek inną, do której należysz (wymień *nowa_grupa* na właściwą nazwę):
`$ chgrp nowa_grupa plik2a`
albo
`$ chown :nowa_grupa plik2a`
Spróbuj to samo z inną grupą, do której **NIE** należysz.

(b) Sprawdź uprawnienia publicznie czytalnego oraz chronionego pliku używanego w pierwszym ćwiczeniu:
`$ ls -l /etc/passwd`
`$ ls -l /etc/shadow`

(c) Wylistuj zawartość katalogu domowego:
`$ ls -l ~`
Pamiętaj, że katalogi muszą mieć prawo `'x'`, aby można je było listować (por. slajdy).

(d) Utwórz plik tekstowy zawierający komendy Basha (skorzystaj z dowolnego edytora tekstu w tym celu):
`$ cat plik_exe`
```
date
uname
id -un
```
Daj temu plikowi prawo do wykonywania:
`$ chmod +x plik_exe`
i wywołaj go:
`$ ./plik_exe`
(uwaga: katalog bieżący zwykle nie należy do `$PATH`, więc musisz nazwę ścieżkę programu pełną ścieżką; można używać jokerów i reprezentować ją przez `'./'`).
Gratulacje! Właśnie napisałeś/-aś swój pierwszy skrypt bashowy. Nie jest on zbyt poważny, więc nie dodawaj jeszcze sekcji *skryptowanie w Bashu* do swojego CV.

(e) Spraw, by Twój skrypt był odczytywalny publicznie:
`$ chmod ugo+r plik_exec # albo wręcz a+r`
Czy Twoi sądziedzi mogą go wyświetlić? `$ cat ~/../twoj_uzytkownik/plik_exe`
Dodaj prawa do listowania katalodu domowego wszystkim użytkownikom: `$ chmod +x ~` i poproś sąsiadów o ponowienie.
Czy mogą oni wylistować zawartość Twojego katalogu domowego i innych plików tamże?
W końcu, zmień uprawnienia dostępu do Twojego katalogu domowego na możliwie bardzo restrykcyjne:
`$ chmod 700 ~`
albo
`$ chmod go-x ~`

(f) Zmień uprawnienia pliku `'plik_exe'` na zerowe:
`$ chmod 000 plik_exe`
Czy możesz zmienić nazwę pliku?
Czy możesz zmienić jego grupę? Czy możesz go usunąć?

(g) Utwórz plik z uprawnieniami ustawionymi na 640:
```
$ touch plik2g
$ chmod 640 plik2g
$ ls -l plik2g
```
Zmień uprawnienia tego pliku na 777:
```
$ chmod 777 plik2g
$ ls -l plik2g
```

i w końcu wycofaj je znów do 640 używając ‘chmod’ w trybie symbolicznym:
`$ chmod u=rw,g=r,o-rwx plik2g`

(h) Wylistuj wszystkie pliki wykonywalne w ‘/usr/bin’ z *setuid*: `$ ls -l /usr/bin/ | grep -E '^.{3}s'`
albo alternatywnie:
`$ find /usr/bin/ -perm -u=s | sort`

(i) Zmodyfikuj poprzednie przykłady: wylistuj wszystkie pliki z *setgid*, w potem mające zarówno *setuid* jak i *setgid*.

(j) Sprawdź czy ‘/usr/bin/su’ ma ustawioną flagę *setuid*: `$ ls -l /usr/bin/su`
Skopiuj ten plik do swojego katalodu roboczego:
`$ cp /usr/bin/su .`
i sprawdź, czy kopiowanie zachowuje tę flagę:
`$ ls -l`
Niestety, nie! Sprzątanie:
`$ rm su`

(k) Utwórz plik w katalogu ‘/tmp’: `$ echo cokolwiek > /tmp/$(id -un)`
Pozwól swojemu sąsiadowi połączyć się z Twoją maszyną via SSH (zakładając, że nazwa stacji roboczej to ‘p21801’ – sprawdź format prompta):
`$ ssh p21801`
Czy może on/ona usunąć ten plik? Nie – to efekt działania *sticky bit* dla ‘/tmp’.

(l) Sprawdź swoje ‘umask’:
`$ umask`
Zmień je na 000, utwórz nowy plik i sprawdź jego uprawnienia:
`$ umask 000; touch plik2l1; ls -l plik2l1`
Później zmień ‘umask’ na 777, utwórz nowy plik i sprawdź jego uprawnienia:
`$ umask 777; touch plik2l2; ls -l plik2l2`
Zresetuj ‘umask’ do domyślnej wartości:
`$ umask 022`

(m) Sprzątanie:
`$ rm plik2*`

3. Listy kontroli dostępu

(a) Znajdź pliki w katalogu ‘/dev’, w których jest ustawiony ACL:
`$ ls -l /dev/ | grep -P '^[^ ]*\+'`
Te ACL pozwalają użytkownikom środowiska graficznego (DE) mieć dostęp do pewnych ustawień urządzeń multimedialnych.

(b) Wyświetl uprawnienia ACL (załóżmy, że ‘/dev/video0’ był wylistowany w poprzednim ćwiczeniu):
`$ getfacl /dev/video0`
Poniższe ćwiczenia mogą nie być wykonywalne ze względu na specyfikę NFS (net file system) używanym w naszych laboratoriach – skonsultuj się z prowadzącym.

(c) Udostępnij swój katalog domowy wszystkim użytkownikom:
`$ chmod +x ~`
Utwórz w nim plik i pozwóc sąsiadowi/sąsiadce go odczytawać i do niego pisać:
`$ touch plik3c`
`$ setfacl -m u:`*`twoj_sasiad`*`:rw plik3c`
Wyświetl jego uprawnienia (zwróć uwagę na symbol ‘+’) oraz jego ACL: `$ ls -l`
`$ getfacl plik3c`
Poproś sąsiada/sąsiadkę, by zmodyfikowali jego zawartość:
`$ echo "jakiś tekst" > ~/../`*`twoj_uzytkownik`*`/plik3c`
Sprawdź, czy zawartość faktycznie się zmieniła:
`$ cat plik3c`
Usuń uprawnienia ACL dla pliku:

```
$ setfacl --remove-all plik3c
```

i wycofaj zmiany uprawnień katalogu domowego do możliwie najbardziej restrykcyjnych (por. 2(e)).