HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Analiza sigurnosnih problema SUID programa na Unix operacijskim sustavima

CCERT-PUBDOC-2001-06-04

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

U ovom dokumentu opisani su sigurnosni problemi vezani za SUID programe. U normalnim uvjetima rada, UNIX programi i skripte ljuske izvode se pod istim pravima koja ima i korisnik koji ih je pokrenuo. Zbog toga tipično korisnici ne mogu promijeniti svoju zaporku editiranjem `/etc/passwd` datoteke, niti bilo koja komanda koju oni pokrenu ne može mijenjati integritet ove zaporke. SUID programi, međutim, mijenjaju normalna prava pristupa i pisanja te su uvijek pokrenuti sa punim pravima njihovog vlasnika. Upravo zbog toga `/usr/bin/passwd` komanda omogućava korisnicima sustava promjenu njihove zaporke. Ova komanda je SUID program i njen vlasnik je `root` korisnički račun pa je ona uvijek pokrenuta sa svim pravima `root` korisničkog računa.
Kada novi administratori otkriju SUID mogućnosti, obično u njima vide rješenje svih problema prilikom izrade neke skripte ili programa. Također, te se mogućnosti odmah i počinju koristiti što ponekad uvelike olakšava posao pisanja programa. U ovom dokumentu opisani su tipični programi skripta ljuske na koje treba obratiti pažnju i mogućnosti rješavanja istih.

2. Problematična skripta

Da bi ovaj sigurnosni problem bio bolje objašnjen, napisana je jednostavna skripta ljuske koja je nakon toga dorađivana kako bi se onemogućio pojedini sigurnosni problem. Ova skripta služi za promjenu zaporke korisnika i poziva `passwd` program da bi napravila isti posao.

```
% ls change-pass
-rwsr-x---   1 root helpdesk    37 Feb 26 16:35   change-pass
% cat change-pass

#!/bin/csh -b
set user = $1
passwd $user
```

Ova jednostavna skripta omogućava korisniku grupe `helpdesk` mijenjanje zaporke bilo kojeg korisnika. Ovaj primjer izabran je upravo zato što ovakva potreba najčešće postoji u različitim korporacijama kada `helpdesk` treba biti u mogućnosti mijenjati zaporku korisnika. Skripta je postavljena sa SUID bitom i mogu ju izvršavati samo `root` korisnički račun ili bilo koji korisnički račun koji pripada `helpdesk` grupi. No, ova naizgled jednostavna skripta ljuske ima puno sigurnosnih problema koji su analizirani u ovom dokumentu.

2.1. Problem csh ljuske

Prvi sigurnosni problem vezan je uz csh ljusku sustava koja ima loše napravljeno rukovanje varijablama okružja. Neovlašteni korisnici mogu vrlo jednostavno manipulirati varijablama korisnika, te se preko ove skripte može doći do ljuske sa administratorskim privilegijama na slijedeći način:

```
% env TERM='`cp /bin/sh /tmp/sh ; chown root /tmp/sh ; \
chmod 4755 /tmp/sh`' change-pass
```

Ovom komandom u jednoj je liniji promijenjena `TEMP` varijabla okružja i pokrenuta je skripta. U TEMP varijabli navedene su zapravo druge komande kojima je cilj napraviti kopiju sh ljuske sustava u `/tmp` direktoriju i postaviti SUID bit na nju. Kasnijim pokretanjem ove kopije ljuske sustava iz `/tmp` direktorija neovlašteni korisnik ima sva administratorska prava na računalu. Problem nastaje upravo zbog neispravnog rukovanja varijablama okružja od strane csh ljuske sustava. Zbog ovog razloga pisanje skripti se preporučuje u drugim ljuskama poput ksh, bash ili zsh.
Ista skripta napisana u ksh ljuski izgleda:

```
% cat change-pass
#!/bin/ksh
user=$1
passwd $user
```

Pisanje skripe u ksh ljusci eliminiralo je sigurnosni problem kod csh ljuske ali i dalje postoje drugi sigurnosni problemi. Ovakva skripta ima sigurnosni problem manipuliranja varijable puta direktorija (`PATH`). Budući da skripta koristi relativna imena komandi a ne apsolutna moguće je promijeniti `PATH` varijablu koja će ukazivati na drugu `passwd` komandu, a ne na onu u `/usr/bin` koja se želi izvršiti:

```
% export PATH='/tmp'
% echo "cp /bin/sh /tmp/sh ; chown root /tmp/sh ; chmod \
4755 /tmp/sh" > /tmp/passwd
% ./change-pass
```

Budući da je u ovom primjeru bila promijenjena `PATH` varijabla `change-pass` skripta pokrenuti će `/tmp/passwd` program umjesto željenog `/usr/bin/passwd`-a. Rezultati su očigledni, opet će biti napravljena kopija ljuske sustava u `/tmp` direktoriju na koju će biti postavljen SUID bit. Nakon toga neovlašteni korisnik pokretanjem te kopije ljuske sustava ima administratorska prava na računalu.

2.2. Postavljanje PATH varijable i korištenje apsolutnih imena

Ovi sigurnosni problemi mogu se eliminirati na slijedeći način:

```
% cat change-pass
#!/bin/ksh
PATH='/bin:/usr/bin'
user=$1
/usr/bin/passwd $user
```

Nakon ovih promjena put direktorija koji se koristi je siguran, kao i pokretanje pravog `passwd` programa. Međutim, ovakva skripta omogućava mijenjanje bilo čije zaporke, pa čak i administratorove.

2.3. Razumijevanje programa

Slijedeći logičan korak je dodavanje dijela u skriptu koji raspoznaje unesenog korisnika, kako bi se onemogućila promjena zaporke administratorskog korisničkog računa:

```
$ cat change-pass
#!/bin/ksh
PATH="/bin:/usr/bin"
user=$1
rm /tmp/.user
echo "$user" > /tmp/.user
isroot='/usr/bin/grep -c root /tmp/.user'
[ "$isroot" -gt 0 ] && echo "You can't change root's \ password!" &&
exit
/usr/bin/passwd $user
```

Nakon ovog dodavanja skripta će završiti sa radom ukoliko je pokrenuta sa argumentom korisnika administratorskog računa. No, razumijevanje programa je ovdje izrazito bitno. Ukoliko neovlašteni korisnik pokrene ovu skriptu bez unošenja bilo kojeg argumenta, i `/usr/bin/passwd` će biti pozvan bez argumenta. U ovom slučaju `passwd` će promijeniti zaporku za trenutnog korisnika – kako je riječ o SUID skripti trenutni korisnik je `root` korisnički račun što će opet omogućiti promjenu zaporke.

U ovom dijelu se vidi potreba za razumijevanjem rada programa kako bi se moglo shvatiti koje je promjene potrebno dodati u skriptu da se onemogući promjena zaporke `root` korisničkog računa. Skripta iz slijedećeg koraka će izgledati:

```
$ cat change-pass
#!/bin/ksh
PATH="/bin:/usr/bin"
user=$1
[ -z $user ] && echo "Usage: change-pass username" && exit
rm /tmp/.user
echo "$user" > /tmp/.user
isroot='/usr/bin/grep –c root /tmp/.user'
[ "$isroot" –gt 0 ] && echo "You can't change root's \ password!" &&
exit
/usr/bin/passwd $user
```

Iako je ovim postupkom riješen sigurnosni problem koji je omogućavao promjenu bilo čije zaporke, postoji još sigurnosnih rupa. Pažljivim pregledavanjem skripte može se vidjeti da se koristi privremena datoteka u `/tmp` direktoriju. Skripta prvo briše ovu datoteku, nakon toga je ponovno stvara, zapisuje u nju korisničko ime koje je navedeno kao argument i na kraju provjerava da li je to korisničko ime bilo ono `root` korisničkog računa.
Sigurnosni problem koji se javlja kod ovako napisanih skripti ljuske se obično naziva utrkom (eng. *racing*). Naime, neovlašteni korisnik u ovom slučaju može vrlo pažljivo tempirati vrijeme tako da točno nakon što skripta obriše `/tmp/.user` datoteku, ali i trenutak prije nego što ju je skripta ponovo napravila sam napravi istoimenu datoteku. No, u slučaju da skripta ne promijeni sadržaj ove datoteke neovlašteni korisnik može opet postaviti lažni argument i promijeniti zaporku administratorskog računa. Da bi se ovaj posao pojednostavio, neovlašteni korisnici obično rade specijalne programe koji im omogućavaju automatsku provjeru stvaranja i brisanja ovakvih datoteka, te postavljanje njihovih, proizvoljnih, datoteka.

2.4. Uklanjanje privremenih datoteka

Zbog navedenih sigurnosnih problema, preporuka za pisanje skripti ljuske je upravo da se uklone sve privremene datoteke. Navedenu skriptu moguće je prepisati tako da više ne koristi privremene datoteke:

```
% cat change-pass
#!/bin/ksh
PATH='/bin:/usr/bin'
user=$1
[ -z $user ] && echo "Usage: change-pass username" && exit
[ "$user" = root ] && echo "You can't change root's \ password!" &&
exit
/usr/bin/passwd $user
```

Kao što se vidi, ova skripta više ne koristi nikakve privremene datoteke, ali još uvijek nije sigurna. Neovlašteni korisnici mogu na njoj upotrijebiti poznati trik sa ";". ; omogućava pozivanje više komandi iz ljuske sustava u jednoj liniji. Koristeći ovu činjenicu, neovlašteni korisnik može skriptu iz primjera pokrenuti na slijedeći način:

```
% change-pass "user ; cp /bin/sh /tmp/sh ; chown root \ /tmp/sh ;
chmod 4755 /tmp/sh"
```

Ovakvo pozivanje skripte efektivno pokreće komandu:

```
/usr/bin/passwd user ; cp /bin/sh /tmp/sh ; chown root \ /tmp/sh ;
chmod 4755 /tmp/sh
```

Svaka od ovih komandi bila bi pokrenuta pod pravima administratora sustava. Dakle, da bi se uklonio ovaj sigurnosni problem potrebno je ukloniti sve znakove iz ulaznih argumenata koji mogu dovesti do ovakvog pokretanja komandi. Slijedeća skripta to radi:

```
% cat change-pass
#!/bin/ksh
PATH='/bin:/usr/bin'
user=${1##*[ \\$/;()|\>\<&    ]}
[ -z $user ] && echo "Usage: change-pass username" && exit
[ "$user" = root ] && echo "You can't change root's \ password!" &&
exit
/usr/bin/passwd $user
```

Sada su iz ulaznih argumenata uklonjeni svi znakovi iz slijedećeg niza: razmak, \, $, /, ;, (,), |, >, <, & i tabulator.

2.5. Analiza ulaznih argumenata

Ulazni argumenti predstavljaju veliki sigurnosni problem i potrebno ih je pažljivo analizirati i filtrirati. Još jedan poznati sigurnosni problem predstavlja interno dijeljenje polja koje koristi ljuska (eng. *Internal Field Separator – IFS*). IFS specificira koji znak odjeljuje komande. Prilikom normalnog korištenja ljuske ovaj znak je uvijek postavljen na razmak, tabulator ili novu liniju. No, postavljanjem nove vrijednosti za IFS neovlašteni korisnik može mijenjati komande koje će biti izvođenje. Kako skripta iz primjera poziva `/usr/bin/passwd` program za mijenjanje zaporke neovlašteni korisnik može promijeniti IFS u "/" jednostavnim izvođenjem komande:

```
% export IFS='/'
```

Nakon ovoga skripta više neće pokretati `/usr/bin/passwd` program već će probati pokrenuti tri programa, `usr`, `bin` i `passwd`. Sada je neovlaštenom korisniku dovoljno napraviti skriptu pod imenom `usr` koja će biti izvedena.
Zbog ovog sigurnosnog problema uvijek se preporučuje ručno postavljanje IFS vrijednosti u skriptama ljuske:

```
% cat change-pass
#!/bin/ksh
PATH='/bin:/usr/bin'
IFS=' '
user=${1##*[ \\$/;()|\>\<&    ]}
[ -z $user ] && echo "Usage: change-pass username" && exit
[ "$user" = root ] && echo "You can't change root's \ password!" &&
exit
/usr/bin/passwd $user
```

Na žalost, čak ni nakon svih ovih promjena skripta još uvijek nije sigurna. U svim skriptama ljuske postoji inherentna mogućnost utrke izvođenja programa koja se ne može nikako popraviti. Problem je u tome da izvođenje skripte zahtijeva proces u dva koraka. U prvom koraku sustav pokreće proces sa novom ljuskom. U drugom koraku nova ljuska čita sadržaj skripte i izvodi ga po redu. Neovlašteni korisnici mogu postaviti svoje programe koji će biti izvedeni točno u trenutku prije nego što je nova ljuska pročitala skripta. Ovo se obično izvodi simboličkim vezama koji se brišu i nanovo postavljaju u trenutku prije nego što su izvedeni kao što je prikazano na primjeru:

```
% cd /tmp
% ln -s change-pass rootme

% ./rootme &
```

```
% rm rootme && echo "cp /bin/sh /tmp/sh ; chown root /tmp/sh \ ;
chmod 4755 /tmp/sh" >> rootme
```

Na ovaj način postoji vrlo mala šansa za točnim izvođenjem ovih komandi ali potrebno je napomenuti da postoje druge metode koje povećavaju šansu neovlaštenim korisnicima i automatiziraju cijeli proces.

3. Zaključak

Nakon što su sagledani svi sigurnosni problemi skripti ljuske preporuka koja se može dati je ta da se u ljuskama pišu samo programi koji ne trebaju koristiti SUID bitove. Osim toga, potrebno je napomenuti da neki operacijski sustavi ipak nastoje zaobići pojedine sigurnosne probleme navedene u ovom dokumentu, kao što je problem sa utrkom za datotekama. Tako Solaris operacijski sustav predaje novoj ljusci otvorene pokazivače na datoteke koji onemogućavaju neovlaštene korisnike u promjeni ovih datoteka.
Za programe koji koriste SUID bit preporučuje se pisanje *wrapper* programa koji će biti pokrenuti prije njih, a oni sami su napisani u C programskom jeziku ili korištenje programa poput `sudo`.