Tesztelés és felügyelet

Számítástechnika tanár szak
Rendszertechnika II.
előadás

Főbb területek

- A szerverek egyszeri tökéletes beállítása nem elég
- Folyamatos felügyelet szükséges
  - Biztonsági lyukak keresése
  - Szokatlan viselkedések figyelése
  - Gyanús felhasználói tevékenység észlelése
  - Támadási kísérletek naplózása
- Főbb témáik
  - Bejelentkezések és jelszavak
  - Fájlrendszerek
  - Hálózatkezelés
  - Naplózás

Bejelentkezési jelszavak ellenőrzése John the Ripper

- SuSE disztribúcióknak általában része
- Egyszerű használat (minden fiókra)
  - `unshadow /etc/passwd /etc/shadow > jelszavak`
  - `john jelszavak`
- Feltört jelszavak megmutatása
  - `john -show jelszavak`
- Csak bizonyos jelszavak teszteléses
  - `john -users:bela,geza,rita jelszavak`
  - `john -groups:oktatok,hallgatok jelszavak`

Bejelentkezési jelszavak ellenőrzése John the Ripper

- `unshadow` utasítás
  - egybegyűjti a `passwd` és `shadow` állományok információit
  - minden jelszó változtatás után le kell futtatni az utasítást
  - fontos az újraintegrált állomány biztonsága
  - éles környezetben az ilyen műveletet célszerű a hálózatról leválasztva végezni
- Szótárak
  - gyakori szavakat tartalmazó szótárak használata
  - szótári szavak és azok permutációi, kis és nagybetűk, stb.
  - érdemes a felhasználók nyelvéhez tartozó szótárt letölteni
    - `ftp://ftp.ox.ac.uk/pub/wordlists/`
    - `ftp://ftp.cerias.purdue.edu/pub/dict/wordlists/`
  - a környezetnek megfelelő szavak hozzáadása a szótárhoz

Bejelentkezési jelszavak ellenőrzése a CrackLib használata

- A Crack jelszótörő program egy része
- Más programokba történő beágyazásra készült
- `FascistCheck` függvénnyel használható
  - első paramétere: jelszó
  - második paramétere: szótár
- Készíthető saját program is a tesztelésre
- PAM modulokba ágyazható
- SuSE esetén a Yast-ban is beállítható

Bejelentkezési jelszavak ellenőrzése a CrackLib használata

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <crack.h>
#define DICTIONARY "/usr/lib/cracklib_dict"
int main(int argc, char *argv[]) {
 char *password;
 char *problem;
 int status = 0;
 printf("A kilépéshez adjon meg egy üres jelszót.\n");
 while ((password = getpass("\nJelszó: ")) != NULL && *password) {
  if ((problem = FascistCheck(password, DICTIONARY)) != NULL) {
   printf("Rossz jelszó: %s.\n", problem);
   status = 1;
  } else {
   printf("Jelszó rendben\n");
  }
 }
 exit(status);
}
```

Jelszóval nem rendelkező fiókok szűrése

- A legveszélyesebb jelszó, ha nincs jelszó
- A jelszavak a /etc/shadow állományban találhatók
  - Csak a rendszergazda férhet hozzá az állományhoz
  - Kódolt jelszavak
  - A jelszavak a második mezőben találhatók
    - pl.: `c-ta:E5x9NtV/3uWxw:13269:0:99999:7:::`
- A jelszóval nem rendelkező fiókok listázása
  - `awk -F: '$2 == "" {print $1, "Üres jelszó" }' /etc/shadow`
- A jelszó nélküli és az "üres" jelszavas fiókok nem azonosak!

Rendszeradminisztrátori fiókok szűrése

- A rendszergazda felhasználók azonosítója: 0
- Az azonosítók a /etc/passwd állományban találhatók
  - Csak a rendszergazda férhet hozzá az állományhoz
  - Az azonosítók a harmadik mezőben találhatók
    - pl.: c-ta:x:1000:100:Rébay Viktor:/home/c-ta:/bin/bash
- A rendszeradminisztrátori fiókok listázása
  - `awk -F: '$3 == 0 {print $1, "Rendszergazda" }' /etc/passwd`
- A nem 0-s azonosítóval rendelkező felhasználóknak is lehetnek kiemelt jogaik

Gyanús fiókhasználat - az utolsó bejelentkezés adatai

- Veszélyt jelentő fiókok
  - Szunnyadó felhasználói fiókok
  - A számos sikertelen bejelentkezéssel rendelkező fiókok
  - Gyenge jelszóval rendelkező fiókok
- A felhasználó(k) utolsó bejelentkezése
  - `lastlog [-u felhasználónév]`
- A felhasználó(k) utolsó bejelentkezésének tárolása
  - /var/log/lastlog
    - adatbázis formátum (nem napló állomány)
    - látszólagos mérete nem változik
    - néhány disztribúció csak a rendszergazdáknak engedi olvasni
- A felhasználó(k) teljes login előzménye
  - `last [-u felhasználónév]`

Gyanús fiókhasználat - sikertelen bejelentkezések

- Ki- és bejelentkezések, kikapcsolás, újraindítás, futási szint változás tárolása:
  - `/var/log/wtmp` állományban
- A sikertelen bejelentkezések naplózásának bekapcsolása
  - touch /var/log/btmp
  - chown --reference=/var/log/wtmp /var/log/btmp
  - chgrp --reference=/var/log/wtmp /var/log/btmp
- A btmp és wtmp állományok mérete folyamatosan növekszik, gondoskodni kell a forgatásukról

Keresési útvonalak tesztelése

- A keresési útvonal megváltoztatásával elérhető hogy egy ismert program helyett valami más induljon el
- A keresési útvonalban ne szerepeljen relatív elérés
- `perl -e 'print "a PATH relatív könyvtárat tartalmaz\"$_\"\n" foreach grep ! m[^/], split /:/, $ENV{"PATH"}, -1;'`
- Például a `cat` parancs kiadása a /tmp könyvtárban
  - Normál esetben a `/bin/cat` indul
  - Ha a `/tmp` könyvtárban is van egy `cat` állomány
    - Ha a "." előrébb van a $PATH-ban mint "/bin", a /tmp-ben lévő indul
    - Ha a "." hátrébb van akkor a szokott `cat` parancs kerül végrehajtásra
- Veszélyek
  - Gyakran használt fájlnevek mögé bújtatott ártó programok
  - Hatásuk nem mindig vehető észre, mert elvégzik a várt funkciót is
  - Minél előbb szerepel a "." annál nagyobb a veszély
  - Az utolsó pozícióban sem veszélytelen, kihasználhatók az elírások

A setuid vagy setgid programok

- A setuid bit segítségével megoldható, hogy a tulajdonos felhasználó nevében fusson a program
- Például a jelszó változtatásához a `passwd`
  - Szükséges, hogy minden user hozzáférjen a `/etc/shadow` állományhoz
  - Mindenki csak az engedélyezett műveletet hajthassa végre
  - `ls -l /usr/bin/passwd` kimenete
  - `-rwsr-xr-x  1 root shadow 75144 Sep  9 2005 /usr/bin/passwd`
- Az ilyen lehetőséggel megfontoltan kell bánni mert komoly biztonsági problémákat okozhatnak

A setuid vagy setgid programok

- Érdemes tehát áttekinteni ezen programok listáját
  - `find / -xdev -type f -perm +ug=s -print`
  - Adjunk meg kezdőkönyvtárat, mert így minden csatolt fájlrendszerben keresni fog
    - NFS fájlrendszerekben ez elég lassú lehet
    - Bizonyos fájlrendszerekben pedig nincs is értelme
- Keresés a felhasználók könyvtáraiban
  - `find /home -xdev -type f -perm +ug=s -print`
- A setuid vagy setgid bitek eltávolítása
  - `chmod u-s állomány` *a setuid bit eltávolítása*
  - `chmod g-s állomány` *a setgid bit eltávolítása*

Speciális eszközállományok

- Olyan állományok amelyek lehetővé teszik eszközök közvetlen elérését a fájlrendszeren keresztül
- A biztonság érdekében ezek hozzáférése gondos felügyeletet igényel
- Ezek másolatai kiindulópontok lehetnek a memória, a lemezmeghajtók és egyéb fontos eszközök olvasásához
- A speciális eszközállományok listázása
  - `find /dir -xdev \( -type b -o -type c \) -ls`
    - `-type b`: blokkos állományok
    - `-type c`: karakteres állományok
    - nem csak a `/dev` könyvtárban lehetnek ilyen állományok
- A /dev összes szabályos állományának listázása (kivéve MAKEDEV)
  - `find /dev -type f ! -name MAKEDEV -ls`
    - szabályos állományok a speciális állományok helyettesítésére
    - rejtőzködő setuid vagy setgid állományok
    - MAKEDEV kivétel, mivel ennek segítségével hozhatók létre új bejegyzések
- Fájlrendszerek csatlakoztatása a `nodev` opcióval
  - megakadályozza a eszközállományok felismerését és használatát

Rootkitek keresése

- Rootkitek, férgek, trójai programok és egyéb támadásokra utaló jelek után kutat
- Általában az első lépés lehet ha támadásra gyanakszunk
- Érdemes a használt verziót folyamatosan frissíteni
- Telepítése
  - Része a disztribúciónak
  - Letölthető a http://www.chkrootkit.org címről
    - Ellenőrizzük a letöltött állomány ellenőrző összegét
- Futtatása
  - root jogosultságokkal
  - A chkrootkit számos linux parancsot használ, ha ezek már kompromittálódtak, akkor hamis lehet a kimenet
  - `chkrootkit -p /mnt/cdrecorder`
    - futtatás megbízható bináris állományok felhasználásával egy nem írható médiumon

Nyitott portok ellenőrzése

- Általában a támadások első lépése, előzzük meg a támadókat!
- Kiszolgáltatottságunk függhet:
  - A támadás kiindulási pontjától
    - Külső és belső támadások
    - Forrás IP címe, portja (címhamisítás)
  - A kommunikáció során érintett tűzfalak
    - Saját tűzfal(ak)
    - Szolgáltató tűzfala
  - A védett rendszer hálózati konfigurációja
    - Beérkező kapcsolatok belépési pontja
    - Engedélyezett beérkező kapcsolattípusok
- A hálózat tesztelése kívülről
  - Saját távoli felhasználói fiók felhasználásával
  - Egy tesztkörnyezet kialakításával és használatával

Az `nmap` parancs

- Hatékony eszköz a hálózati biztonság tesztelésére
- A tesztelést körültekintően kell végezni
  - A rendszergazdák támadásnak értékelhetik
  - célszerű mindenkit értesíteni aki az adott kiszolgálóval dolgozik
  - Az `nmap` megsérti a hálózati protokollokat ez megzavarhatja az éles rendszereket
- Az információgyűjtés fázisai
  - Kiszolgálók felkutatása
    - Egy vagy több kiszolgáló letapogatása a hálózatban
  - Portletapogatás
    - A kapcsolatot fogadó, nyitott portok feltérképezése
  - Az operációs rendszer ujjlenyomatának vizsgálata
    - A hálózati viselkedés egyedi jellemzői alapján

`nmap` alapműveletek

- TCP portok scannelése
  - nmap -v kiszolgáló
- UDP portok scannelése
  - nmap -v -sU kiszolgáló
- Operációs rendszer ujjlenyomat azonosítás
  - nmap -v -O kiszolgáló
- Kiszolgáló felkutatása egy tartományban
  - nmap -v -sP 192.168.1.100-150
- Kiszolgálók felkutatása egy ("C") hálózati osztályban
  - nmap -v kiszolgáló /24
  - nmap -v 192.168.1.0/24
  - nmap -v 192.168.1.0-255
  - nmap -v "10.12.104.*"

Kiszolgálók felkutatása

- Csak a bekapcsolt gépekhez tartozó portokat kell letapogatni
  - A kiszolgálók felkutatása során az aktív állapot lekérdezése történhet
    - ICMP ping üzenetek segítségével
    - TCP ping üzenetek segítségével
  - Ha tűzfalak tiltják a fenti ping-eket, kikapcsoltnak tűnhet a gép
  - Ha biztosak vagyunk a bekapcsolt állapotban
    - nmap -P0 opcióval tilthatjuk a kiszolgáló felkutatását

A `telnet` és `nc` utasítások

- Portok ellenőrzése telnettel
  - telnet c-ta-linux.ttk.pte.hu ssh
    - nyitott port
    - zárt port
    - tűzfallal védett port
- nc (netcat)
  - nc -z -vv kiszolgáló portok
    - nyitott port
    - zárt port
    - tűzfallal védett port
- nc6
  - nc6 --recv-only -vv kiszolgáló port

A `netstat` program

- Összefoglaló információ a hálózatkezelés állapotáról
- Aktív hálózati kapcsolatok megjelenítése:
  - `netstat --inet`
  - gyanús lehet a sok `SYN_RECV` (portletapogatás)
- Az aktív kapcsolatok fogadására kész szerver socket-ek megjelenítése
  - `netstat --inet --listening`
  - Igyekezni kell minden socket feladatát tisztázni
  - A felesleges socketeket kikapcsolni
- Minden listázása
  - `netstat --all`

Az `lsof` parancs

- Processzekhez tartozó nyitott állományok listázása
- Kapcsoló nélkül mindent listáz
- Nyitott hálózati kapcsolatok listázása
  - `lsof -i [TCP|UDP][@server][:port]`
  - Például
    - `lsof -i :ssh`
    - `lsof -i TCP`
    - `lsof -i TCP@iatt.ttk.pte.hu:22`
- IP címek és portszámok automatikus konverziója
  - Sok nyitott hálózati kapcsolat esetén lassú lehet
    - `-n` IP címek használata a kiszolgálónevek helyett
    - `-P` portszámok használata a szolgáltatás neve helyett
    - `-l` user ID-k user felhasználói nevek helyett

Rendszerhívások követése

- Egy processz rendszerhívásainak követése
  - `strace -p PID`
    - megjeleníti a rendszerhívások paramétereit, visszaadott értékei és az esetleges hibákat
    - a processz és a kernel között átadott információkat
  - minden rendszerhívás követése nagyon sok információt jelent
- Rendszerhívások egy csoportjának figyelése
  - hálózati tevékenység figyelése
    - `strace -e trace=network`
  - Mivel a hálózati socketek gyakran végeznek írási és olvasási műveleteket:
    - `strace -e trace=network,read,write`
- Például
  - `strace -e trace=network,read,write -p 12345`

Hálózati forgalom figyelése

- A hálózati interfészen megjelenő csomagok
  - unicast: egycímes csomagok, az adott gépnek címezve
  - multicast: többcímes csomagok, például videó vagy hanganyagok esetén
  - broadcast: csoportos csomagok, a hálózat minden gépe számára fontos információ vagy ha ismeretlen a cél
- Az egyes interfészekre nem csak a nekik szóló csomagok érkeznek
  - Normál módba másnak szóló csomagokat figyelmen kívül hagyja
  - Promiscuous (lehallgató) mód: a hálózat összes csomagját fogadja
    - rendszergazdaként kapcsolható mód
    - az átkapcsolás naplózásra kerül
- Kapcsolók és jelosztók (hub) …
- Útválasztók és átjárók …

A `tcpdump` program

- Promiscuous mód be- és kikapcsolása
  - `ifconfig interfész promisc`
  - `ifconfig interfész -promisc`
- Forgalomfigyelés a tcpdump programmal
  - `tcpdump -w dumpfile [-c csomagok száma] [-i interfész] [-s hossz] [kifejezés]`
    - az interfész lehet `all` is
    - alapesetben a csomagoknak csak az első 68 bájtját menti el, a -s opcióval ez növelhető, 0-val a teljes csomag rögzíthető
- Rögzítendő csomagok szűkítése
  - capture filter (rögzítési szűrő) alkalmazása
  - Például
    - `tcpdump -w proba.dump host freemail.hu`
    - `tcpdump -w proba.dump tcp port telnet and host freemail.hu`

A `tcpdump` program

- Elmentett nyomkövetési adatok megjelenítése
  - nem igényel root jogosultságot
    - normál állomány
    - a tartalma miatt érdemes lehet egyéb módon védeni
  - `tcpdump -r dumpfile`
- Elfogott csomagok megjelenítése közvetlenül
  - `-w` és `-r` nélkül
- Az elfogott csomagokat javasolt állományba menteni:
  - az elemzés gyakran az adatok különböző formában történő többszöri megjelenítését igényli
  - Régebbi adatok elemzése és összehasonlítása is szükséges lehet
  - Nem mindig tudhatjuk előre, hogy mi érdekel (szűrési feltételek)
  - A megjelenítés további erőforrásokat használ fel
  - Elkerülhető a további felesleges forgalom generálása
    - ssh-n keresztüli futtatáskor a megjelenítés további ssh forgalmat generál
    - IP címek DNS nevekké konvertálása további "felesleges" DNS kéréseket generál

Az `ngrep` program

- karakterláncok keresésére a hálózati forgalomban
- Lehetőségei:
  - `ngrep [grep opciók] reguláris-kifejezés [szűrő]`
  - `ngrep -X hexa számok [szűrő]`
  - `ngrep -O állomány [-n számláló] [-d interfész] [-s hossz] reguláris-kifejezés [szűrő]`
  - `ngrep -I állománynév reguláris-kifejezés [szűrő]`
- Csak önálló csomagokban figyeli a minták illeszkedését
  - FTP esetén például jól használható
  - Telnet esetén nem igazán használható
- Egy lehetséges megoldás
  - forgalom rögzítése a tcpdump segítségével
  - a rögzített forgalom szűrése az ngrep használatával
  - szükség esetén a teljes rögzített tartalom átnézése az ethereallal