Operativsystem Labb: Säkerhet i Linux

Assistent:
Stefan Grönroos
stefan.gronroos(ät)abo.fi

Labb: Säkerhet i Linux

- Första delen: En virtuell masking ”Vulnerable Debian” för VirtualBox används.
- Tyngdpunkt på knäckande av lösenord och buffer overflow attacker.
- Vulnerable Debian har flera nivåer i form av användarkonton (guest, level10, level20...)
- Sätt upp virtuella maskinen
  - Två nätverkskort, ett konfigurerat som ”NAT”, och ett som ”Host-only”.
- Börja med att logga in som guest
  - Lösenord: *bemyguest*
  - *För ssh tillgång: /sbin/ifconfig ger ip address*

Nivå 1: Lösenord

- Linux förvarar info om användarkonton i filen /etc/passwd
- Logga in som guest/bemyguest på Vuln. Debian
- Använd virtuella maskinen ”Backtrack 5” och undersök verktygen ”*john the ripper*” samt kanske ”*crunch*”.
- Tips: Ni vill knäcka lösenordet för användaren *level10*. Lösenordet har exakt tre bokstäver.
- Tips2: T.ex. Crunch kan generera en lista med alla lösenord innehållande tre bokstäver, som John sedan kan använda sig av för att knäcka lösenordet.

Nivå 2: SUID program

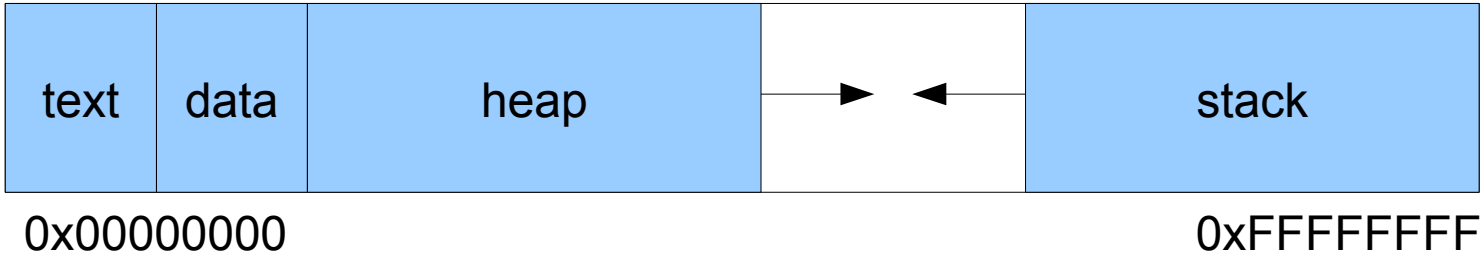
- En exekverbar fil i Linux kan vara konfigurerad med en flagga som gör att programmet körs som den användare som äger filen istället för den som kör den.
- Kan vara nyttigt, men också farligt.
- När ni loggat in till level10 finns en exekverbar fil *lshome* som exekverar som level20.
  - Källkoden för *lshome* finns i lshome.c
  - ”ls -l lshome” ger *-rw**s**r-**x**r--* där s flaggan betyder att den körs som ägaren. X flaggan visar att den är körbar för användare i gruppen group10, till vilken level10 också hör.
- Tips: undersök koden
- Tips2: Vad gör environment variables i linux, speciellt PATH?

Nivå 3: Buffer overflow

- Buffer overflow är en mycket vanlig typ av attack
- Overflow sker när man försöker skriva in mera data än vad som ryms i en buffer, t.ex. en teckenarray.
  - char buffer[128];
  - Rymmer 128 tecken
  - Om vi skriver fler än 128 tecken in i buffern, kommer vi antagligen att skriva över annan data
  - Detta kan utnyttjas för att ändra programflödet!
- Aleph One skrev ”Smashing the stack for fun and profit” år 1996, och där beskrivs hur buffer overflow kan utnyttjas.
  - http://www.phrack.org/issues.html?issue=49&id=14#article

Buffer overflow forts.

- Minnet (Intel x86 32-bit):



- Lokala variabler, t.ex. *i* och *buffer* i exemplet, lagras på stacken.

```
void function(int a, int b, int c) {
   int i;
   char buffer2[12];
}
void main() {
  function(1,2,3);
}
```