Paranoia and government hacking workshop

Henrik Lund Kramshøj

hlk@solido.net

11. december 2013

Indhold

Forord

Dette kursusmateriale er beregnet til brug på kurset *Paranoia and government hacking workshop*. Materialet er lavet af Henrik Lund Kramshøj, http://www.solido.net
Materialet skal opfattes som øvelseshæfte til kurset, og indeholder derfor ikke en fuldstændig beskrivelse af emnet. Der henvises istedet til andet materiale om emnet som nævnt i litteraturlisten.
Til workshoppen hører desuden en præsentation som udleveres.
God fornøjelse

Oversigt

Materialet er inddelt i et antal øvelser som er beregnet til at give kursusdeltagerne et indblik i hvordan Paranoia and government hacking praksis ser ud og opfører sig.

Formålet med workshoppen er at give deltagerne en praktisk erfaring med emnet og information til at implementere i egne miljøer.

Forudsætninger og ordliste

Dette kursusmateriale forudsætter at deltageren har kendskab til internet og e-mail på brugerniveau. Det betyder at begreber som http://www.solido.net, hlk@solido.net ikke bør være ukendte.

Værktøjer

Dette materiale er udarbejdet ved hjælp af en masse værktøjer, og er beregnet på at kunne udføres i et almindeligt kursuslokale med netværksopkoblede pc'er. De praktiske øvelser benytter i vid udstrækning Open Source og kan derfor afvikles på blandt andet følgende platforme:

- UNIX - herunder Linux, OpenBSD, NetBSD, FreeBSD og Mac OS X
- Microsoft Windows 7 - primært som klientoperativsystem

Det anbefales at benytte virtualiseringsplatforme til hackerværktøjer, herunder BackTrack Linux. Der findes flere alternativer som:

- VMware Player https://www.vmware.com/products/player/
- VirtualBox https://www.virtualbox.org/
- Xen http://www.xen.org/

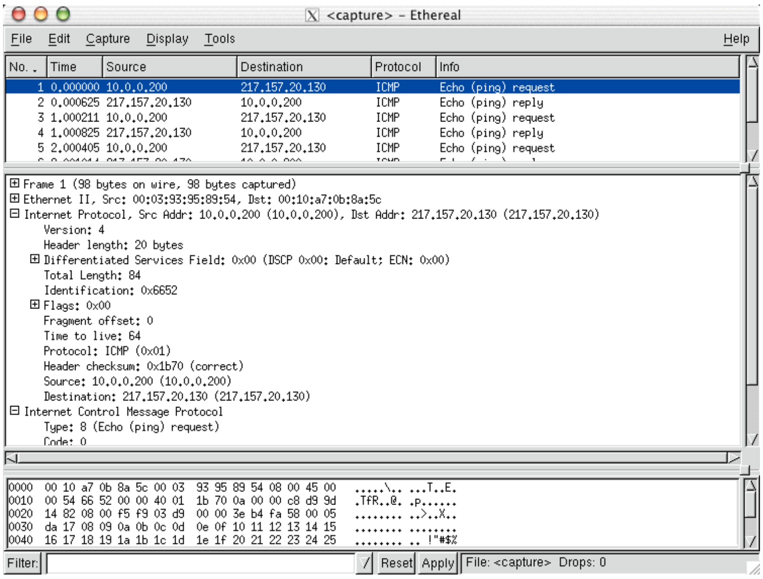
Indholdet i øvelserne

De fleste af øvelserne har følgende indhold:

- **Opgave:** Hvad går øvelsen ud på
- **Formål:** Hvad forventes det at man lærer ved at løse opgaven
- **Forslag til fremgangsmåde:** er en hjælp til at komme igang
- **Hjælp:** er flere tips eller beskrivelser af hvordan man kan løse opgaven
- **Forslag til løsning:** en mulig løsning til opgaven
- **Diskussion:** er oplæg til diskussion efter løsning af opgaven. Der er mulighed for at sammenligne og diskutere de valgte løsninger.

Øvelse 1

Wireshark installation



```
No. . Time     Source          Destination     Protocol Info
    1 0.000000 10.0.0.200      217.157.20.130  ICMP     Echo (ping) request
    2 0.000625 217.157.20.130  10.0.0.200      ICMP     Echo (ping) reply
    3 1.000211 10.0.0.200      217.157.20.130  ICMP     Echo (ping) request
    4 1.000825 217.157.20.130  10.0.0.200      ICMP     Echo (ping) reply
    5 2.000405 10.0.0.200      217.157.20.130  ICMP     Echo (ping) request
```

```
⊞ Frame 1 (98 bytes on wire, 98 bytes captured)
⊞ Ethernet II, Src: 00:03:93:95:89:54, Dst: 00:10:a7:0b:8a:5c
⊟ Internet Protocol, Src Addr: 10.0.0.200 (10.0.0.200), Dst Addr: 217.157.20.130 (217.157.20.130)
     Version: 4
     Header length: 20 bytes
   ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
     Total Length: 84
     Identification: 0x6652
   ⊞ Flags: 0x00
     Fragment offset: 0
     Time to live: 64
     Protocol: ICMP (0x01)
     Header checksum: 0x1b70 (correct)
     Source: 10.0.0.200 (10.0.0.200)
     Destination: 217.157.20.130 (217.157.20.130)
⊟ Internet Control Message Protocol
     Type: 8 (Echo (ping) request)
     Code: 0
```

```
0000  00 10 a7 0b 8a 5c 00 03  93 95 89 54 08 00 45 00   .....\.. ...T..E.
0010  00 54 66 52 00 00 40 01  1b 70 0a 00 00 c8 d9 9d   .TfR..@. .p......
0020  14 82 08 00 f5 f9 03 d9  00 00 3e b4 fa 58 00 05   ........ ..>..X..
0030  da 17 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
```

File: <capture> Drops: 0

Objective:
Install the program Wireshark locally on the Windows workstation

Purpose:
Installing Wireshark will allow you to analyse packets and protocols

Suggested method:
Download and install the program, either download from web server locally or from http://www.wireshark.org Wireshark requires a Windows Capture library to be installed, which is included in the Wireshark installation, but can none the less be downloaded fromhttp://www.winpcap.org/

Hints:
PCAP is a packet capture library allowing you to read packets from the network. Wireshark is a graphical application to allow you to browse through traffic, packets and protocols.

Solution:
When Wireshark is installed sniff some packets, also see next exercise.

Discussion:
Wireshark is just an example other packet analyzers exist, some commercial and some open source like Wireshark

Øvelse 2

Sniffing network packets

Objective:
Sniff packets and dissect them using Wireshark

Purpose:
See real network traffic, also know that a lot of information is available and not encrypted.

Suggested method:
Open Wireshark and start a capture - either from Windows or BackTrack
Then in another window execute the ping program while sniffing

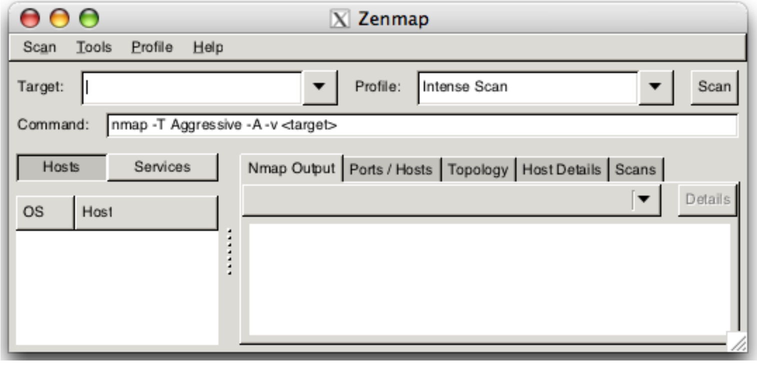
Hints:
When running on Linux the network cards are named eth0 for the first Ethernet and wlan0 for the first Wireless network card. In Windows the names of the network cards are long and if you cannot see which cards to use then try them one by one.

Solution:
When you have collected some packets you are done.

Discussion: Is it ethical to collect packets from an open wireless network?

Øvelse 3

Discover active systems ping sweep



Profile: Intense Scan

Command: nmap -T Aggressive -A -v <target>

Objective:
Use Nmap to discover active systems

Purpose:
Know how to use Zenmap from the Nmap suite to scan networks for active systems.

Suggested method:
Try different scans,

- Ping sweep to find active systems
- Port sweeps to find active systems with specific ports

Hints:
Try nmap in sweep mode

Solution:
Use the command below as examples:

- Ping sweep `nmap -sP 10.0.45.*`
- Port sweeps `nmap -p 80 10.0.45.*`

Discussion:

You can also use the graphical interface to nmap called Zenmap.

Øvelse 4

Execute nmap TCP and UDP port scan

Objective:
Use nmap to discover open ports on active systems

Purpose:
Finding open ports will allow you to find vulnerabilities on these ports.

Suggested method:
Use `nmap -p 1-1024 server` to scan the first 1024 TCP ports

Try to use `nmap -sU` to scan using UDP ports, not really possible if a firewall is in place.

If a firewall blocks ICMP you might need to add `-P0` or even `-PN` to make nmap scan even if there are no Ping responses

Hints:
Sample command: `nmap -P0 -sU -p1-1024 server` UDP port scanning 1024 ports without doing a Ping first

Solution:
Discover some active systems and you are done.

Discussion:
There is a lot of documentation about the nmap portscanner, even a book by the author of nmap. Make sure to visit http://www.nmap.org

TCP and UDP is very different when scanning. TCP is connection/flow oriented and requires a handshake which is very easy to identify. UDP does not have a handshake and most applications will not respond to probes from nmap. If there is no firewall the operating system will respond to UDP probes on closed ports - and the ones that do not respond must be open.

When doing UDP scan on the internet you will almost never get a response, so you cannot tell open (not responding services) from blocked ports (firewall drop packets). Instead try using specific service programs for the services, sample program could be `nsping` which sends DNS packets, and will often get a response from a DNS server running on UDP port 53.

Øvelse 5

Perform nmap OS detection

Objective:
Use nmap OS detection and see if you can guess the devices on the network

Purpose:
Getting the operating system of a system will allow you to focus your next attacks.

Suggested method:
Look at the list of active systems, or do a ping sweep.

Then add the OS detection using the option `-O`

Hints:
Use the manual page

The nmap can send a lot of packets that will get different responses, depending on the operating system.

Solution:
Use a command like `nmap -O -p1-100 10.0.45.45`

Discussion:
nmap OS detection is not a full proof way of knowing the actual operating system, but in most cases in can detect the family and in some cases it can identify the exact patch level of the system.

Another tool which does the same is Xprobe.

Øvelse 6

Perform nmap service scan

Objective:
Use more advanced features in nmap to discover services.

Purpose:
Getting more intimate with the system will allow more precise discovery of the vulnerabilities and also allow you to select the next tools to run.

Suggested method:
Use `nmap -A` option for enabling service detection

Hints:
Look into the manual page of nmap or the web site book about nmap scanning

Solution:
Run nmap and get results.

Discussion:

Some services will show software versions allowing an attacker easy lookup at web sites to known vulnerabilities and often exploits that will have a high probability of success.

Make sure you know the difference between a vulnerability which is discovered, but not really there, a false positive, and a vulnerability not found due to limitations in the testing tool/method, a false negative.

A sample false positive might be reporting that a Windows server has a vulnerability that you know only to exist in Unix systems.

Øvelse 7

Find systems with SNMP

Objective:
Use snmpwalk to research SNMP systems

Purpose:
Learn that gathering information can help an attacker.

Suggested method:
Log into the Unix server provided and run snmpwalk which is using UDP port 161.

Hints:
We are running in a LAN environment with less firewalls, so doing nmap UDP scan is possible.

When discovering an IP then use the `snmpwalk` program to show a lot of information.

Solution:

- Use the command `snmpwalk -v 2c -c public 10.0.45.34 | less`

The command less will show output one screen at a time.

Discussion:
In real networks SNMP is being used a lot, but new equipment is starting NOT to allow access using the community string public.

Øvelse 8

Try Hydra brute force

Objective:
Try a brute force program named hydra/Xhydra

Purpose:
Learn that some protocols allow brute forcing.

Suggested method:
Log into the Unix server or use the BackTrack.

Make a short list of usernames and a short list of passwords and use hydra to brute force your way into a system. Use the editor `kate`, using `kate users.txt` and `kate pass.txt` followed by a command similar to this:

```
$ hydra -V -t 1 -L users.txt -P pass.txt 10.0.45.2 ssh
```

Hints:
When learning tools create a nice environment and check that things are working before trying to hack. So with brute forcing an account, create and test it!

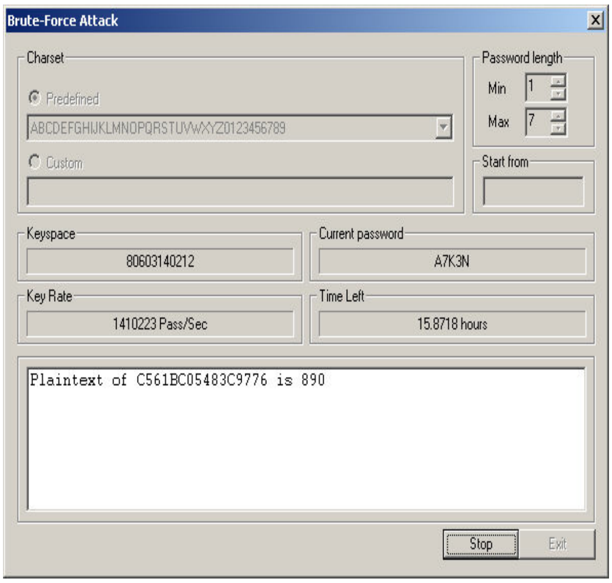
Solution:
There is an FTP server with an easy to guess administrator password.

Discussion:
The hydra program can brute force a lot of different protocols and also allow a lot of tuning.

The hydra program does an online brute force attack, in some cases you can get access to data like password databases, or hash values that can be cracked in off-line brute force attacks.

Øvelse 9

Try Cain brute force



Charset

● Predefined

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

○ Custom

Password length

Min 1

Max 7

Start from

Keyspace 80603140212

Current password A7K3N

Key Rate 1410223 Pass/Sec

Time Left 15.8718 hours

```
Plaintext of C561BC05483C9776 is 890
```

Objective:
Try a brute force program named Cain

Purpose:
Learn that some algorithms allow for easier brute forcing.

Suggested method:
Download and install the Windows program Cain

Then try cracking some local accounts, access to hash is only allowed if you are administrator.

Hints:
When learning tools create a nice environment and check that things are working before trying to hack. So with Cain use a system where you are administrator and crack local accounts.

Then later get hash values from real systems, or by doing google searches.

Solution:
See that some algorithm can do 100.000s keys/second and others only allow 100s keys/second.

Discussion:
Cain is built for cracking passwords in off-line brute force attacks, but also includes other features like sniffing.