

Ethical Hacker Workshop

exercises

Henrik Lund Kramshøj

hlk@solidonetworks.com

May 24, 2010



Contents

1	Putty installation - Secure Shell login	6
2	WinSCP installation - Secure Copy	8
3	Login to Unix server	9
4	Get to know some Unix	10
5	Access the root on Unix	11
6	Unix boot CD	12
7	Wireshark installation	14
8	Sniffing network packets	15
9	Discovery using ping and traceroute	16
10	ICMP tool - icmpush	17
11	Lookup Whois data	18
12	Discover using DNS	19
13	Try the bind-version shell script	21
14	Try the dns-timecheck Perl program	22
15	Research arpspoof and dsniff	23
16	Discover active systems ping sweep	24
17	Execute nmap TCP and UDP port scan	25
18	Perform nmap OS detection	26
19	Perform nmap service scan	27
20	Find systems with SNMP	28
21	Try Hydra brute force	29
22	Try Cain brute force	30
23	Network scripting using netcat	31
24	OpenSSL forbindelser	32

CONTENTS

25	OpenVAS scanning	33
26	Discover wireless networks	34
27	Aircrack-ng	35
A	Host information	36

Preface

This material is prepared for use in *ethical hacker workshop* and was prepared by Henrik Lund Kramshøj, <http://www.security6.net>

This material is expected to describe networking setup and applications for trainings and workshops where hands-on exercises are needed.

Further a presentation is used which is handed out and some other documents that can assist during exercises.

Have fun and learn

Overview

This material has some degree of freedom with regards to setup of the environment.

The purpose is to give participants a feel for practical setups. The suggested configurations and applications are close to real life scenarios but have been designed to fit in with existing infrastructures used for training.

Prerequisites

This material expects that participants have a working knowledge of TCP/IP from a user perspective. Basic concepts such as web site addresses and email should be known as well as IP-addresses and common protocols like DHCP.

Tools used

These exercises are expected to be performed in a training setting with network connected workstations.

The exercises use a number of tools which can be copied and reused after training.

Tools used are mostly:

- Unix - such as Linux, OpenBSD, NetBSD, FreeBSD or Mac OS X
- Microsoft Windows - primary use is for workstations
- The requirements for the workstations are a browser and Secure Shell Access
- In most trainings a Linux based security tool is distributed which is called BackTrack. This tool can be used as a live CD or installed to hard disk.

Introduction to networking

IP - Internet protocol suite

It is extremely important to have a working knowledge about IP to implement secure and robust infrastructures. Knowing about the alternatives while doing implementation will allow the selection of the best features.

ISO/OSI reference model

A very famous model used for describing networking is the ISO/OSI model of networking which describes layering of network protocols in stacks.

This model divides the problem of communicating into layers which can then solve the problem as smaller individual problems and the solution later combined to provide networking.

Having layering has proven also in real life to be helpful, for instance replacing older hardware technologies with new and more efficient technologies without changing the upper layers.

In the picture the OSI reference model is shown along side with the Internet Protocol suite model which can also be considered to have different layers.

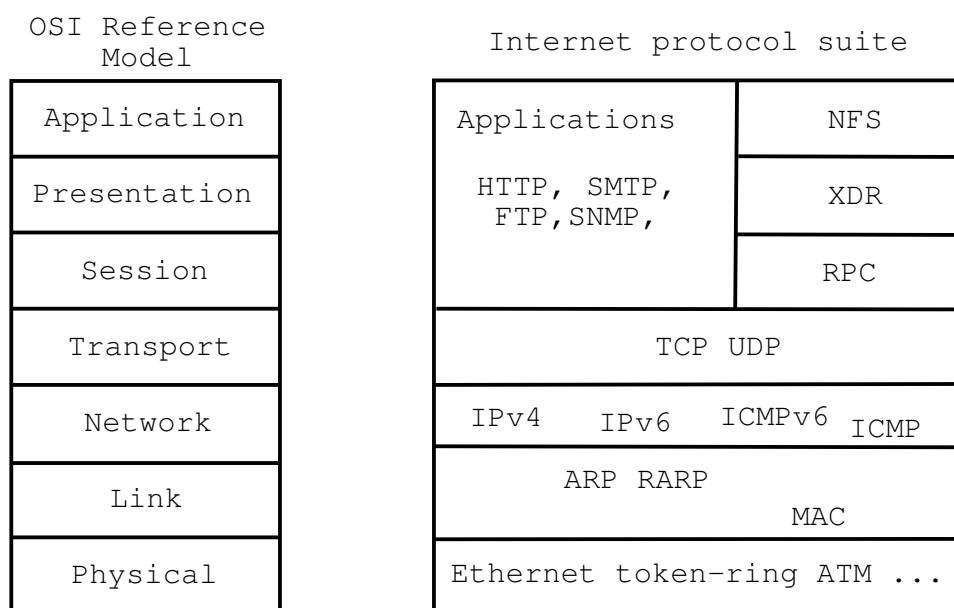


Figure 1: OSI og Internet Protocol suite

Standards and RFC

The internet has a number of working groups which are tasked with describing new features and protocols considered for use on the internet. These working groups primarily function across the internet using open mailing lists in which anyone can contribute to discussion.

When consensus is reached the features are described in document which are named Request For Comments, or RFC for short. These documents can be obtained free of charge from their web site <http://www.rfc-editor.org/>.

Some RFC documents describe actual standards or specific uses and are noted in various indexing documents called standards (STD), For Your Information (FYI) and Best Current Practice (BCP).

Whenever a standard is to be updated a new RFC is published and the old version is not changed allowing the RFC series to also document the development of the internet standards from the oldest documents in the 1969.

One example is the IP specification itself (IPv4) from 1981:

0791 Internet Protocol. J. Postel. Sep-01-1981. (Format: TXT=97779 bytes) (Obsoletes RFC0760) (Updated by RFC1349) (Also STD0005) (Status: STANDARD)

As specified the document RFC-0791 is a standard and it was superseded by the new version which is RFC-1349 - which was in fact also updated by other documents.

Addressing in the network

The network is expected to use private IP-addresses, which are specified in RFC-1918 *Address Allocation for Private Internets*

The default subnet to use is:

- 10.0.45.0/24 - which is about 250 addresses in a subnet with 24 mask bits

If internet connectivity is needed and available it will be connected through a router leaving us with an isolated subnet which can be used for various experiments.

Hardware and networking used

This chapter describes the required hardware and software used for doing exercises.

The requisites should be similar to what is found in a normal setting with PCs running Microsoft Windows clients and having basic network connectivity.

Parts of the exercises are using Unix, specifically OpenBSD and Linux. Unix is provided and no prior knowledge of Unix is expected.

A number of programs to be used on Microsoft Windows are provided using a web server:

- Putty - SSH access from Microsoft Windows
- Winscp - easy access to the filesystem on the Unix server using SSH and also has a built-in editor
- Wireshark - an open source network protocol analyzer

Exercise content

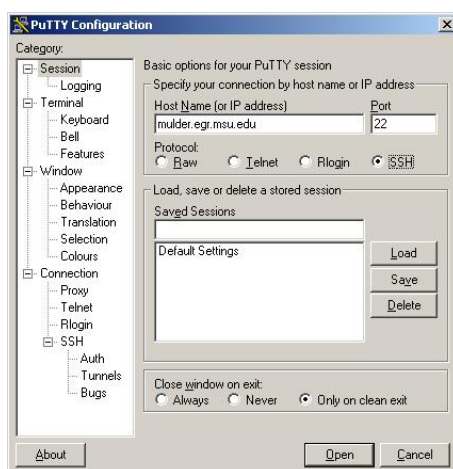
Most exercises follow the same procedure and has the following content:

- **Objective:** What is the exercise about, the objective
- **Purpose:** What is to be the expected outcome and goal of doing this exercise
- **Suggested method:** suggest a way to get started
- **Hints:** one or more hints and tips or even description how to do the actual exercises
- **Solution:** one possible solution is specified
- **Discussion:** Further things to note about the exercises, things to remember and discuss

Please note that the method and contents are similar to real life scenarios and does not detail every step of doing the exercises. Entering commands directly from a book only teaches typing, while the exercises are designed to help you become able to learn and actually research solutions.

Exercise 1

Putty installation - Secure Shell login



Objective:

Install the program Putty locally on your workstation

Purpose:

Installing Putty will make sure you have administrative access and allow us to use Secure Shell for connecting to Unix systems and networking devices.

Suggested method:

Download and install the program, either download from web server locally or from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Hints:

Putty is a terminal emulator and replaces the telnet program in Windows. It is often the preferred way of connecting to Unix systems and is also available in network devices such as switches, routers and firewalls.

Further Putty will enable serial connections which can be used for configuring equipment through console connections. Remember to select the method when using Putty.

It is suggested to save profiles for future use, and remember to change a profile you should load the profile, make changes and **remember to go back and save the profile** before opening a connection. Otherwise the profiles changes will only be active in the current connection.

Solution:

Do a normal installation with default settings.

If you know Putty already you can investigate the Puttygen program and research the use of public and private keys.

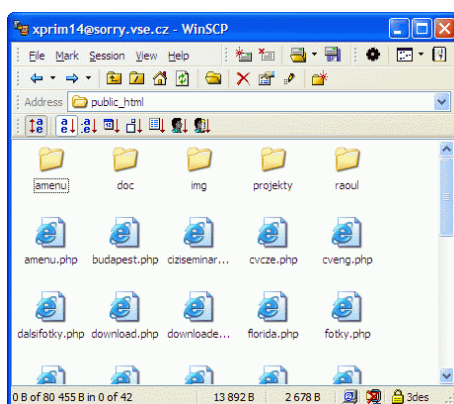
Discussion:

The Secure Shell protocol is an internet standard for secure terminal connections and the same protocol allows file transfer and forwarding of network packets.

Note: the protocol version 2 is the one recommended

Exercise 2

WinSCP installation - Secure Copy

**Objective:**

Install the program WinSCP locally on your workstation

Purpose:

Get required programs ready for doing exercises.

Suggested method:

Installing WinSCP will make sure you have access to transferring files from Unix systems and networking devices.

Hints:

WinSCP is very helpful allowing easy access to files using Secure Shell protocol and also when working with text files it is possible to use the built-in editor of WinSCP.

Solution:

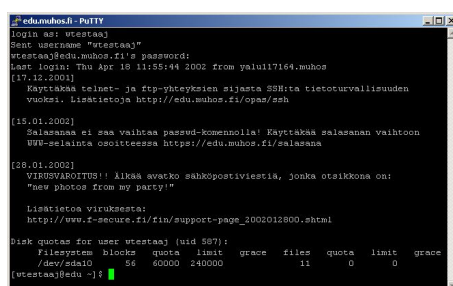
Download and install the program

Discussion:

WinSCP can also be used instead of FTP, why is that helpful?

Exercise 3

Login to Unix server



```
edu-muhos.fi - PuTTY
login as: wtestaa1
Sent username "wtestaa1"
wtestaa1@edu-muhos.fi's password:
Last login: Thu Apr 18 11:55:44 2002 from yalul17169.muhos
[17.12.2001]
  Käyttäkää telnet- ja ftp-yhteyksien sijasta SSH:tä tietoturvallisuuden
  vuoksi. Lisätietoja http://edu.muhos.fi/opas/ssh

[15.01.2002]
  Salasanat ei saa vaihtaa passwd-komenolla! Käyttäkää salasanan vaihtoon
  WWW-selainta osoitteessa https://edu.muhos.fi/salasana

[18.01.2002]
  VIRUSVAROITUS!! Älkää avakko sähköpostiviestiä, jonka otsikkona on:
  "new photos from my party!"

  Lisätietoa viruksesta:
  http://www.f-secure.fi/fi/support-page_2002012800.shtml

Disk quotas for user wtestaa1 (uid 597):
  Filesystem blocks quota limit grace files quota limit grace
  /dev/sda10  56  60000  240000      11  0      0      0
wtestaa1@edu- ~$
```

Objective:

Do a remote login from your workstation to the servers provided

Purpose:

Make sure the network is working and allow you to use the Unix system for exercises.

Suggested method:

You will use Putty or another Secure Shell program and login to the servers provided

Hints:

Use the Putty program or boot the Linux Live CD and run ssh from the command line.

Using the Linux Live CD the OpenSSH programs are already installed and available and are used with commands like this::

```
ssh username@server -p port
```

 which for the actual server is:

```
ssh team1@10.0.45.45 -p 22
```

NB: the server may have another IP-address due to the use of DHCP

The users defined all have the password **team**

Solution:

Start Putty or boot using the Linux Live CD

Discussion:

The Linux Live CD is based on Open Source and may be copied freely.

The BackTrack security distribution contain more than 300 security programs and is being updated actively.

Exercise 4

Get to know some Unix

Objective:

Try a few Unix commands and see that help is available

Answer the following questions:

- What does the command `cal` do? What happened in September 1752?
- What does the commands `date`, `clear` and `echo` do?

Purpose:

Learn enough Unix to be able to run simple commands from the command line

Suggested method:

Log into the Unix system and try executing the commands

After trying the commands use the manual pages with the following commands:

`man cal`, `man date`, `man clear`, `man echo`

```
$ date
...
$ cal
...
$ cal 2009
...
$ cal 1752
...
output is not shown on purpose, try it for yourselves :-)
```

Hints:

The manual system is always available on Unix and usually you can do searches when displaying a manual page using the operators `/` (forward search) and `?` (backward search).

Solution:

Type `man cal` and do a search by entering `/`, the year 1752 and press enter

Discussion:

Searching using `/` and `?` are very common on Unix

Exercise 5

Access the root on Unix

Objective:

Learn to use the `sudo` command to gain root access.

Purpose:

Know a way to gain access as root user - to run hacker programs later

Suggested method:

Run the command and use the manuals of the two commands `su` and `sudo` to answer the following questions:

- What is the goal of the programs?
- What are the similarities and differences?
- Can the `su` command be configured not to use a password? can `sudo`?
- What password needs to be entered when using the programs, your password or the superuser password?

Hints:

`Switch user` is the old command used to gain root access - and requires the knowledge of the password for the root user or the other user your are switching to. `Su` always give complete access by switching to the user id. `Sudo` is a more modern way to control access.

Solution:

Use the command `sudo -s` to get root access and then `exit` to exit superuser.

Discussion:

Unix systems have traditionally used the switch user `su` - but the superuser do `sudo` is much more modern and flexible by allowing you to specify specific commands and permissions on a fine grained permission model.

`Sudo` is used almost exclusively and is considered the de facto way of gaining root on Unix systems.

An example use of `sudo` might be the restarting of a web server with apache control:

```
hlk@bigfoot:hlk$ sudo apachectl configtest
Syntax OK
hlk@bigfoot:hlk$ sudo apachectl restart
hlk@bigfoot:hlk$
```

(Note: when things succeed Unix wont say much, only if something unexpected happens there will be output)

Exercise 6

Unix boot CD

**Objective:**

Boot a Live CD on the workstation

Purpose:

Learn to use Live CD's - specifically the BackTrack Live CD

Suggested method:

Insert the DVD and boot from it

Hints:

There is a large number of Live CDs built on the Linux operating system specifically designed for various purposes. Some of the well known CDs are:

- Knoppix which include a lot of productivity tools, like web browser, office suite, mail programs etc.
- BackTrack which include more than 300 security tools and a premade Linux kernel with a lot of security related patches.
- Damn Vulnerable Linux which is also a security CD but the focus is on providing a learning environment for security training. Some tools help work with buffer overflows and others provide an opportunity to do reverse engineering

Solution:

When booted use the commands shown below

Discussion:

The Live CDs are designed to be used on most computer, but some models require more work - typically the graphic card or wireless network card can cause trouble.

If that should happen it is recommended to search on the internet, to see if others have tried using Linux on the specific brand and model of computer.

In case of the wireless card not working it is recommended to research and buy a wireless network card that is known to work.

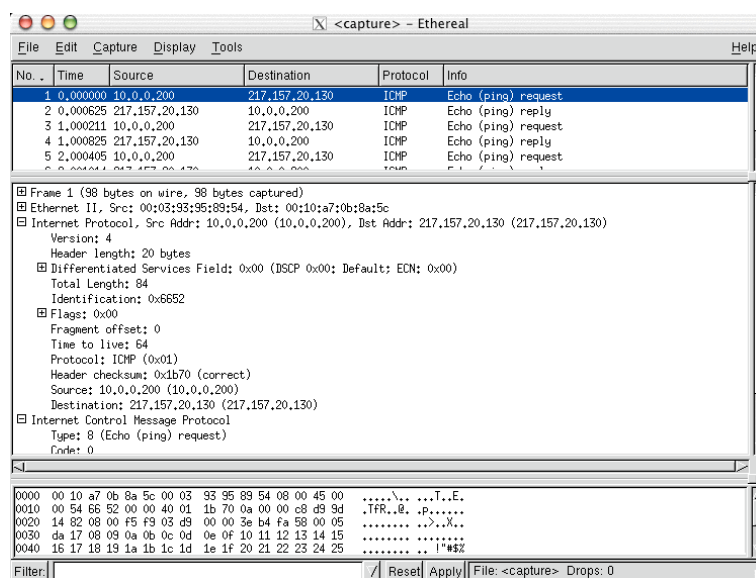
Note: When working with the BackTrack CD the following commands are usefull:

- `startx` will enter the graphical environment
- `/etc/init.d/networking start` will try configuring the network on all interfaces with DHCP
- `dhclient eth0` start a single DHCP client using a specific network card, like eth0
- `wicd` followed by `wicd-client` will start a wireless client program to allow you to join wireless networks
- `apt-get update` and `apt-get upgrade - upgrade` when installed in hard disk
- `apt-get update` and `apt-get dist-upgrade - upgrade` with major upgrades

The individual tools on the BackTrack are described in detail on the internet and some of the tools, like Wireshark and nmap will have excellent documentation available.

Exercise 7

Wireshark installation



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 from <http://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

Exercise 8

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?

Exercise 9

Discovery using ping and traceroute

Objective:

Learn how to use the ping and traceroute programs.

Purpose:

Doing network discovery is an important part of doing security testing.

Suggested method:

Use `ping` and `traceroute` testing your network connection.

Can be performed from both Windows and Unix/Linux

Remember though that traceroute is named `tracert` on Windows.

Hints:

ICMP is the Internet Control Message Protocol which is used for reporting problems back to a source on the internet. It can also be used for diagnosing problems using ICMP ECHO request packets. ICMP is very important when doing security testing for network discovery and making sure connections are alive.

The following protocols are being used:

- Ping uses ICMP packets with request and expect responses
- Tracert on Windows uses ICMP packets
- Traceroute on Unix by default uses UDP packets, but can also use ICMP

Solution:

Run the commands - not all are available on Windows, so perhaps use Unix:

- `traceroute` (Unix) or `tracert` (Windows)
- `traceroute -I`

Discussion: A lot of people just try to block any ICMP, but that will actually hurt a lot of functionality within your network.

Other trace programs exist, for example TCP traceroute programs - find them on the BackTrack!

Exercise 10

ICMP tool - icmpush

Objective:

See a sample program that allows you to send ICMP packets without doing actual programming

Purpose:

Know that a lot of hacker programs exist on any level of IP

Suggested method:

Login to the Unix server - see the manual and use timestamp request packets

Alternative install icmpush on BackTrack using the command `apt-get`, try running icmpush and then follow on screen instructions.

Hints:**Solution:**

Use the command `icmpush -v -tstamp 10.0.45.45` and also try echo, mask from the icmpush program

Discussion:

Other toolboxes for creating network packets are:

- Nemesis - which is on the BackTrack
- Scapy - which allow you to do Python programs that can send packets
- Hping - which is on the BackTrack

Exercise 11

Lookup Whois data

Objective:

Learn to use Whois databases

Purpose:

Knowing who to contact in case of problems on the internet is important, and also verifying before starting scanning is required.

Suggested method:

- Login to the UNIX server and use `whois` or use the web interfaces like <http://www.ripe.net>

Hints:

Whois databases are distributed to Regional Internet Registries such as ARIN, AfriNIC, RIPE, LACNIC and APNIC.

Solution:

Use the specified command above with an IP address, `whois 91.102.91.17`.

Discussion:

The whois system was implemented after the Morris Worm affected the internet in November 1988, because it was realized that the internet had grown to a size that required more management.

Exercise 12

Discover using DNS

Objective:

Try some programs for doing Domain Name System (DNS) lookups

Purpose:

Learning to do network discovery includes looking into public information such as DNS

Suggested method:

Try these commands:

- nslookup - available on both Unix and Windows, but not recommended anymore
- Try `nslookup -q=txt -class=CHAOS version.bind. 0`
- Try `dig @ns1.gratisdns.dk www.security6.net A`
- Try `host -a security6.net` and `host -a www.security6.net` any difference?
- The host program uses the syntax `host host server` while dig uses `dig @server host`

Hints:

Host is available by default on OpenBSD, so use the Unix server provided

There are a lot of Graphical User Interface programs available both for Unix and Windows

Solution:

Run the commands above, output would be like this:

```
$ host -t ns security6.net
security6.net name server ns1.gratisdns.dk.
security6.net name server ns2.gratisdns.dk.
security6.net name server ns3.gratisdns.dk.
security6.net name server ns4.gratisdns.dk.
security6.net name server ns5.gratisdns.dk.
$ host -t ns security6.net 217.157.20.131
...
```

Discussion:

Previously it was possible to do Zone Transfers, but today most DNS system administrators do not allow that. If possible a zone transfer will reveal all names for a domain.

Make sure that you know the difference between forward and reverse lookups. Forward is from name to IP address lookup, while reverse does a lookup from IP address to name.

Exercise 13

Try the bind-version shell script

Objective: Try to use a shell script to automate lookups

Purpose:

When doing actual security testing you should automate as much as possible.

Suggested method: Login to the Unix server provided and run the bind-version script

Hints: Unix files with # ! as the first line will be executed using the command specified.

Unix shell scripting is very usefull and the book *Classic shellscripting* is recommended when doing shell scripting.

Unix also typically include scripting languages like Perl, Python, Ruby, Groovy, ...

Solution:

Run the script provided

Discussion: The script only does a few DNS lookups, but more elaborate scripts are being used daily by administrators, security consultants and hackers.

The script available on the system is:

```
#!/bin/sh
# Try to get version info from BIND server
# many ways to do it
# nslookup -q=txt -class=CHAOS version.bind. 0
# dig @$* version.bind chaos txt
PROGRAM=`basename $0`
TARGET=$1

if [ $# -ne 1 ]; then
    echo "get name server version, need a target! "
    echo "Usage: $0 target"
    echo "example $0 10.1.2.3"
    exit 0
fi

# using dig
dig @$1 hostname.bind chaos txt
dig @$1 ID.SERVER chaos txt
dig @$1 version.bind chaos txt
dig @$1 authors.bind chaos txt
```


Exercise 14

Try the dns-timecheck Perl program

Objective: Try to use a Perl script to communicate with a binary protocol

Purpose:

See that programming languages such as Perl often include a lot of libraries which allow efficient implementation of ideas.

Suggested method: Login to the Unix server provided and run the dns-timecheck script

Hints: Perl can be a bit difficult to read, but a lot of tutorials exist

Solution:

Discussion: While Perl has been around for lots of years it seems that security tools are often implemented using these languages:

- Perl, of course :-)
- Python - like Scapy
- Ruby - like Metasploit

The script available on the system is:

```
#!/usr/bin/perl
# modified from original by Henrik Kramshøj, hlk@kramse.dk
# 2004-08-19
#
# Original from:
# http://www.rfc.se/fpdns/timecheck.html

use Net::DNS;

my $resolver = Net::DNS::Resolver->new;
$resolver->nameservers($ARGV[0]);

my $query = Net::DNS::Packet->new;
$query->sign_tsig("n","test");

my $response = $resolver->send($query);
foreach my $rr ($response->additional) {
    print "localtime vs nameserver $ARGV[0] time difference: ";
    print $rr->time_signed - time() if $rr->type eq "TSIG";
    print "\n";
}
```

Exercise 15

Research arpspoof and dsniff

Objective:

Read about arpspoof and dsniff

Purpose:

Realize that having a switch does not prevent sniffing, but makes it a bit more difficult.

Suggested method:

Log onto the Unix server and look at manual pages

Hints:

ARP spoofing is about sending false information to systems trying to communicate. If it happens the systems will send their packets to the wrong destination, the hacker who can then sniff data and forward.

Dsniff is a program that can decode a lot of older protocols.

Solution:

To read manual pages use: `man arpspoof` and `man dsniff`

Discussion:

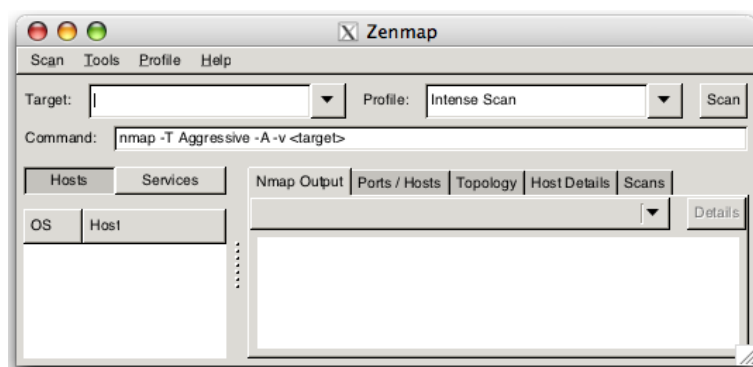
What can be done using these programs?

Please notice that it can make the network a bit unstable if you are not careful. Luckily the network will recover by itself in 5-10 minutes.

A graphical tool is available on the BackTrack named Ettercap.

Exercise 16

Discover active systems ping sweep



Objective:

Use nmap to discover active systems

Purpose:

Know how to use nmap 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.

Exercise 17

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.

Exercise 18

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 it 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.

Exercise 19

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.

Exercise 20

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.

Exercise 21

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.

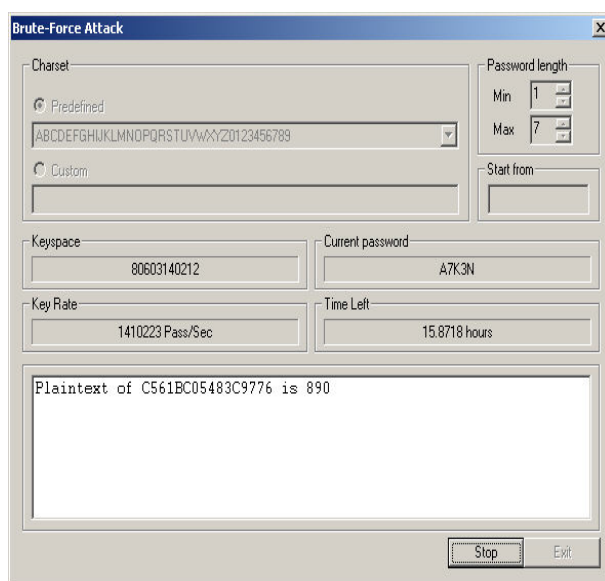
Discussion:

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.

Exercise 22

Try Cain brute force

**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.

Exercise 23

Network scripting using netcat

Objective:

Learn how to use the netcat program for scripting

Purpose:

Learn that a lot of protocols on the internet are easy read and create tools for.

Suggested method:

Login to the Unix server - look at the manual `man nc`. Then create a textfile named `headh.sh` using this content

```
#!/bin/sh
# get HEAD from Webserver
cat | nc $1 $2 << EOF
HEAD / HTTP/1.0

EOF
```

Then use the command `chmod +x head.sh` to make it executable and run it

Hints:

The netcat program is a swiss army-knife for network data, and allows you to forward data to various ports and connect programs.

Solution:

Run the program: `./head.sh www.pentest.dk 80`

Discussion:

Sometime the program will seem to hang, use `ctrl-c` to break it.

Exercise 24

OpenSSL forbindelser

Objective: Learn how to use the OpenSSL programs to do scripting protocols wrapped in SSL/TLS

Purpose:

Learn that even if protocols are being wrapped in encryption you can write test programs.

Suggested method:

Login to the Unix server - look at the manual `man openssl`. Note the possibility of using `openssl s_client`. Then create a textfile named `headssl.sh` using this content

```
#!/bin/sh
# get HEAD from Webserver SSL port
openssl s_client -host $1 -port $2 << EOF
HEAD / HTTP/1.0

EOF
```

Then use the command `chmod x headssl.sh` to make it executable and run it

Hints: `Openssl` programmet kan fungere som en wrapper til forbindelser til webservere og andre protokoller som benytter SSL/TLS

Solution:

Run the program: `./headssl.sh server 443`

Discussion:

Another program for SSL is `sslsn` available on the BackTrack to allow you to know the allowed algorithms on a web server running SSL/TLS.

Exercise 25

OpenVAS scanning

Objective:

Use the OpenVAS system to do a more complex test.

Purpose:

See that more user friendly applications exist, but that these tools still require you to know the details.

Suggested method:

Create a certificate for the OpenVAS server, create a user, then start the server and client.

Hints:

There are a number of programs in the OpenVAS environment, but typing `openvas` and then pressing TAB twice will show you:

- `openvas-mkcert` make a certificate for the server
- `openvas-adduser` add a user
- `openvasd` start the OpenVAS server
- `OpenVAS-Client` client program that connects to the server

If you have installed BackTrack on a server make sure that you run these command as the superuser, like `sudo openvasd`

Solution:

Run the programs shown above in that order

Discussion:

Note that OpenVAS is based on the source code from Nessus. Nessus has for many years been the tool of choice for a lot of companies when doing security testing.

Unlike commercial tools which are often Windows tools that require you to bring a laptop to a specific network to allow testing this OpenVAS is based on a client-server model.

The client can be anywhere and the server only needs to be close to the network being tested.

Exercise 26

Discover wireless networks

Objective:

Install wardriving tool on a laptop and run the program.

Purpose:

See how to discover wireless networks, even ones that are not broadcasting.

Suggested method:

Using various tools it is possible to see all the networks in use at a specific place.

Some tools used for this are: inSSIDer (Windows), netstumbler(Windows), Kismet(Linux), Airodump-ng (Linux) and Kismac

Hints:

You need a network card that supports monitor mode, and the driver.

Some vendor keep programming information secret, making it hard to use for wardriving - in that case you might need to go buy another :-)

Solution:

See the programming running.

Discussion:

Is it ethical to look for wireless networks?

Is it ethical to publish results on the internet?

Exercise 27

Aircrack-ng

Objective:

See the program aircrack-ng being used for cracking WEP and WPA-PSK keys.

Purpose:

Some methods previously used to protect wireless networks should not be used anymore.

Suggested method:

Get access to an encrypted dump of wireless network traffic and break encryption.

Hints:

BackTrack includes the aircrack-ng program and some test data in
`/pentest/wireless/aircrack-ng/test`

Solution:**Discussion:**

There is a lot of information available about aircrack-ng at the web site:

<http://www.aircrack-ng.org/>

Another tool on the BackTrack is pyrit and cpyrit which can break WPA-PSK using CUDA enabled graphic cards - instead of 100s of keys/second this may allow 10000s keys/second.

Appendix A

Host information

- You should note the IP-addresses used for servers and devices
- The web server for installing programs:
http:// . . . /public/windows/
- Server used for team login: . . .
Available usernames: team1, team2, ... team10 password: team
- You can obtain root access using: `sudo -s`

Available servers and devices:

- IP: . . . -
- IP: . . . -
- IP: . . . - OpenBSD
- IP: . . . - OpenBSD server
- IP: . . . - Your workstation with Windows/Linux