2019 oplossingen labo 1 - Lars Lemmens

Met dank aan de Github van Martijn en natuurlijk Lars Lemmens

LABO 1

In order to connect to a remote SSH server leia.uclllabs.be in this example, execute the following command

```
'user:~$' • ssh r1234567@leia.uclllabs.be -p 22345
```

The client configuration file can make your life easier because default command line options could be specified in this configuration file

Normally you would use the following command to log in to leia

```
'user:~$' • ssh -p 22345 r1234567@leia.uclllabs.be

"FILE: ~/.ssh/config"
```

```
Host leia
HostName leia.uclllabs.be
Port 22345
User r1234567
```

The following command creates an ssh tunnel. Local tcp port CPORT (on your laptop) will be tunneled through GATEWAY to SERVER port

```
'user:~$' • ssh -f GWUSERNAME@GATEWAY -L CPORT:SERVER:SPORT -N
```

- The -f argument instructs the ssh instance to go into the background, and -N instructs it to not launch a shell.
- The -L argument specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.

Try it yourself: There is a website running - https://darthvader.uclllabs.be/nw2/lab1 which is only accessible from 193.191.177.1. That's the IP address of leia.uclllabs.be.

```
'user:~$' • ifconfig | grep -B1 inet
```

- ifconfig configures a network interface
- The grep command grep searches the named input FILEs
- -B NUM prints NUM lines of leading context before matching lines

With this command, you can create an ssh tunnel between server leia.uclllabs.be and

the webserver darthvader.uclllabs.be.

```
'user:~$' • ssh -p 22345 -f r0123456@leia.uclllabs.be -L 10000:darthvader.uclllabs.be:443 -N
```

- The -p argument shows which port to connect to on the remote host
- The -f argument instructs the ssh instance to go into the background, and -N instructs it to not launch a shell.
- The -L argument specifies that the given port on the local (client) host is to be forwarded to the given host and port on the remote side.

First find the ssh process to close:

```
'user:~$' • ps fauxw | grep darthvader
```

- The ps command reports a snapshot of the current processes
- The f argument does a full-format listing
- The a argument selects all processes except both session leaders and processes not associated with a terminal.
- The u argument selects by effective user ID (EUID) or name.
- The x arugment Lift the BSD-style "must have a tty" restriction, which is imposed upon the set of all processes when some BSD-style (without "-") options are used or when the ps personality setting is BSD-like.
- The w argument gives wide output
- grep searches the named input FILEs

Next send a sigterm signal to the correct pid (process identifier) = second column in ps fauxw output

```
'user:~$' • kill 12345
```

And if you need to close all running ssh sessions/tunnels use the following command:

```
'user:~$' • killall ssh
```

By default kill sends a SIGTERM signal (similar to clicking on the X to close a window (Chrome, Notepad, Gnome-terminal,..). If you need to force close an application, send the SIGKILL signal:

```
'user:~$' • kill -9 12345
'user:~$' • killall -9 ssh
```

Use a text based browser on leia to verify the remote website is accessible through the ssh tunnel.

```
'user:~$' • lynx https://localhost:10000/nw2/lab1-all-but-leia
'user:~$' • curl -k -s https://localhost:10000/nw2/lab1-all-but-leia/ | html2text
```

- The argument -k instructs curl not to verify the server certificate.
- The argument -s makes curl mute make sure that it doesn't show progress meter or error messages

OpenSSH has built-in support for dynamic port forwarding via the SOCKS proxy protocol. This command turns your SSH client (on your laptop) into a SOCKS proxy server: "

```
'user:~$' • ssh -f user@GATEWAY -D 10000 -N -p port
```

- The -f argument instructs the ssh instance to go into the background, and -N instructs it to not launch a shell.
- The -p argument shows which port to connect to on the remote host
- The -D argument specifies a local “dynamic” application-level port forwarding

There are 3 different levels of debug modes that can help troubleshooting issues. With the -v option SSH prints debugging messages about its progress.

This is helpful in debugging connection, authentication, and configuration problems. Multiple -v options increase the verbosity. Maximum verbosity is three levels deep.

```
'user:~$' • ssh leia -v
'user:~$' • ssh leia -vv
'user:~$' • ssh leia -vvv
```

This command will aid you in finding an available tcp port: "

```
'user:~$' • netstat -lnt | grep -oP ':\K[^:][0-9]+' | sort -un
```

- The command netstat prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
- The -l shows only listening sockets.

- The -n shows numerical addresses instead of trying to determine symbolic host, port or user names.
- The -t argument shows only TCP ports
- The command grep searches for PATTERNS in each FILE.
- The -o argument prints only the matched (non-empty) parts of a matching line, with each such part on a separate output line.
- The -p argument Interpret I as Perl-compatible regular expressions (PCREs).
- : matches the character : literally (case sensitive)
- \K resets the starting point of the reported match.
- [^:] matches a single character not present in the list below
- [0-9] matches a single character present in the list below
- '+' matches the previous token between one and unlimited times, as many times as possible, giving back as needed (greedy)
- The command sort sorts lines of text files
- The -u argument output only the first of an equal run
- The -n argument compares according to string numerical value

For instance, we can scan all ports up to 1000 by issuing this command:

```
'user:~$' • nc -vz leia.uclllabs.be 1-1000
```

- The command nc is a TCP/IP swiss army knife
- The -v argument is for verbose
- The -z argument is for zero-I/O mode [used for scanning]

"Messages returned by Netcat are sent to standard error. You can send the standard error messages to standard out, which will allow filtering the results. Stderr can be redirected to stdout via the 2>&1 bash syntax. Then use grep to filter

Alternatively you could use |& as a shorthand for 2>&1 |. "

```
'user:~$' • nc -z -n -v 193.191.177.1 1-1000 2>&1 | grep succeeded
'user:~$' • nc -z -n -v 193.191.177.1 1-1000 &| grep succeeded
```

- The command nc is a TCP/IP swiss army knife
- The -z argument is for zero-I/O mode [used for scanning]
- The -v argument is for verbose
- The -v argument is used for short version

Netcat is a very slow TCP port scanner as it will wait for a TCP timeout for every port it tries. Add option -w 1 to increase scanning speed, and execute this command on leia to circumvent slowdown due to firewalled ports.

Netcat isn't designed to scan for open ports, it's functionality is very limited. If you need to scan for open ports, use a real portscanner like Nmap"

```
'user:~$' • nmap -p1-1000 193.191.177.1
```

- The command nmap is a network exploration tool and security / port scanner
- The -p argument specifies which ports you want to scan

Once again, you need to choose one end of the connection to listen for connections.

However, instead of printing information onto the screen, you will place all information straight into a file

```
'user:~$' • nc -l -p 10000 > received_file
```

- The command nc is a TCP/IP swiss army knife
- The -l argument is used as listening mode
- The -p argument specifies which ports you want to scan
- The > overwrites a file

On the second computer, create a simple text file by typing: "

```
'user:~$' • echo "Hello, this is a file" > original_file
```

You can now use this file as an input for the Netcat connection you will establish to the remote listening computer.

The file will be transmitted just as if you had typed it interactively: "

```
'user:~$' • nc -l -p 10000 < original_file
```

On the receiving end, you can anticipate a file coming over that will need to be unzipped and extracted by typing: "

```
'user:~$' • nc -l -p 10000 | tar xzvf -
```

- The command nc is a TCP/IP swiss army knife
- The -l argument is used as listening mode
- The -p argument specifies which ports you want to scan
- The tar command is a GNU version of the tar archiving utility
- The -x argument extracts files from an archive
- The -z argument unzips the file
- The -v argument is for verbose
- The -f argument uses archive file
- The ending dash (-) means that tar will operate on standard input, which is being piped from Netcat across the network when a connection is made.

On the side with the directory contents you want to transfer, you can pack them into a tarball and then send them to the remote computer through Netcat:

```
'user:~$' • tar -czf - * | nc leia.uclllabs.be 10000
```

- The tar command is a GNU version of the tar archiving utility
- The -c argument creates a new archive
- The -z argument unzips the file
- The -f argument uses archive file
- The command nc is a TCP/IP swiss army knife

In the following example, we create a full backup of our home directory on leia (all

files and folders) using tar - Netcat method.

Execute this on leia: "

```
'user:~$' • tar -cz ~ | nc -l -p 10000
```

- The tar command is a GNU version of the tar archiving utility
- The -c argument creates a new archive
- The -z argument unzips the file
- The command nc is a TCP/IP swiss army knife
- The -l argument is used as listening mode
- The -p argument specifies which ports you want to scan

And this on your laptop (the machine receiving backups):

```
'user:~$' • cd /path/to/where_I_want_to_store_my_backup
'user:~$' • nc leia.uclllabs.be 10000 | tar -xzf -
```

- The command nc is a TCP/IP swiss army knife
- The tar command is a GNU version of the tar archiving utility
- The -x argument extracts files from an archive
- The -z argument unzips the file
- The -f argument uses archive file w

This is just one example of transferring more complex data from one computer to another. Another common idea is to use the dd command to image a disk on one side and transfer it to a remote computer.

```
'user@leia:~$' • tar -cz ~ | pv | nc -l -p 10000

'user@laptop:~$' • nc leia.uclllabs.be 10000 | pv > backup.tar.gz
```

- The tar command is a GNU version of the tar archiving utility
- The command nc is a TCP/IP swiss army knife
- The -c argument creates a new archive
- The pv command monitors the progress of data through a pipe
- The -z argument unzips the file
- The > argument overwrites the file
- The pv command monitors the progress of data through a pipe
- The command nc is a TCP/IP swiss army knife
- The -l argument is used as listening mode
- The -p argument specifies which ports you want to scan

```
'user@leia:~$' • tar -cz ~ | pv | nc '-N' -l -p 10000

'user@laptop:~$' • nc -d leia.uclllabs.be 10000 | pv > backup.tar.gz
```

- The tar command is a GNU version of the tar archiving utility
- The command nc is a TCP/IP swiss army knife
- The -c argument creates a new archive
- The -d argument makes sure that it does not attempt to read from stdin.
- The -z argument unzips the file

```
'user@laptop:~$' • nc leia.uclllabs.be 10000 </dev/null | pv | tar xzf -
```

- The pv command monitors the progress of data through a pipe
- The -N argument shuts the network socket down after EOF on the input
- The -l argument is used as listening mode
- The -p argument specifies which ports you want to scan
- /dev/null is for error log

Exercise 1:

How do you verify if the packages openssh-server and openssh-client are installed on your Debian system? (Hint:

dpkg -l ...)"

```
'user:~$' • dpkg -l openssh-server
'user:~$' • dpkg -l openssh-client
```

- The argument -l is used for list

Exercise 2:

How do you verify if the openssh-server is running?

```
'user:~$' • systemctl status ssh
```

Exercise 3:

How to check which port is used by the SSH server daemon? (Hint: netstat, ss or lsof)

```
'root #' • netstat -lntp | grep sshd
'root #' • ss -lntp | grep sshd
'root #' • lsof -i tcp | awk '$1=="sshd" && $10=="(LISTEN)" {print $9}' | cut -d ':' -f2 | sort -u
```

- The netstat command prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
- The -l argument is used as listening mode
- The -n argument shows numerical addresses instead of trying to determine symbolic host, port or user names.
- The -p argument shows the PID and name of the program to which each socket belongs
- The grep command prints lines matching a pattern
- The ss command is another utility to investigate sockets
- The lsof command lists open files

- The -i argument selects the listing of files any of whose Internet address matches the address specified in i.
- The awk command is used for pattern scanning and processing language
- The cut command removes sections from each line of files
- The -d argument use DELIM instead of TAB for field delimiter
- The -f argument selects only these fields
- The sort command sort lines of text files
- The -u argument outputs only the first of an equal run

Exercise 4:

How do you disconnect from a remote SSH session?

```
'user:~$' • exit
'user:~$' • logout
'user:~$' • <CTRL>+d
```

Exercise 5:

Is there a config file for the OpenSSH server? If so, where is it located in the file system tree?

```
Yes: '/etc/ssh/sshd_config'
```

Exercise 6:

Create and demonstrate a simple web server with Netcat.

```
'user:~$' • while true; do { echo -e "HTTP/1.1 200 OK\r\n$(date)\r\n\r\n<h1>hello world from $(hostname) on $(date)</h1>" | nc -vl -p 10000; } done
```

- The while command The while command continuously executes the list list-2 as long as the last command in the list list-1 returns an exit status of zero.
- The echo command displays a line of text
- The -e argument enables interpretation of backslash escapes
- The command nc is a TCP/IP swiss army knife
- The -v argument uses verbose
- The -l argument is used for listen mode, for inbound connects
- The -p argument is used for local port number

Revision #1
Created 17 June 2021 14:12:29 by Jasper G.
Updated 3 December 2021 22:13:09 by Jasper G.