# Network Security Essentials

INTRODUCTION TO DEBIAN DISTRIBUTION AND NETWORK TOOLS

dkoutras@unipi.gr

# Debian Based OS

Directory Usage

Basic Commands

Useful Paths

Permissions

Install packages

Script files

# Directory Usage

| | |
|---|---|
| / | root of the virtual directory, where normally, no fi les are placed |
| /bin | binary directory, where many GNU user-level utilities are stored |
| /boot | boot directory, where boot fi les are stored |
| /dev | device directory, where Linux creates device nodes |
| /etc | system configuration fi les directory |
| /home | home directory, where Linux creates user directories |
| /lib | library directory, where system and application library fi les are stored |
| /media | media directory, a common place for mount points used for removable media |

# Directory Usage(2)

/mnt mount directory, another common place for mount points used for

removable media

/opt optional directory, often used to store third-party software packages

and data fi les

/proc process directory, where current hardware and process information is stored

/root root home directory

/sbin system binary directory, where many GNU admin-level utilities are stored

/run run directory, where runtime data is held during system operation

# Directory Usage(3)

/srv service directory, where local services store their fi les

/sys system directory, where system hardware information fi les are stored

/tmp temporary directory, where temporary work fi les can be created and destroyed

/usr user binary directory, where the bulk of GNU user-level utilities and data fi les are stored

/var variable directory, for fi les that change frequently, such as log files

# Basic Commands

**$ file my_file:** Viewing the file type

**$ cat test1:** Viewing the whole file

**$ tail -n 5 log_file:** The file's last 10 lines

**$ head Log_file:** The first 10 lines of text

**$ ps:** **Processes**

- **-A Shows all processes**
- **-N Shows the opposite of the specified parameters**
- **-a Shows all processes except session headers and processes without a**
- **terminal**
- **-d Shows all processes except session headers**

# Basic Commands(2)

- -e Shows all processes
- -C cmslist Shows processes contained in the list cmdlist
- -G grplist Shows processes with a group ID listed in grplist
- -U userlist Shows processes owned by a userid listed in userlist
- -g grplist Shows processes by session or by groupid contained in grplist
- -p pidlist Shows processes with PIDs in the list pidlist
- -s sesslist Shows processes with session ID in the list sesslist
- -t ttylist Shows processes with terminal ID in the list ttylist
- -u userlist Shows processes by effective userid in the list userlist
- -F Uses extra full output

# Basic Commands(3)

**-O format Displays specifi c columns in the list format, along with the default columns**

**-M Displays security information about the process**

**-c Shows additional scheduler information about the process**

**-f Displays a full format listing**

**-j Shows job information**

**-l Displays a long listing**

**-o format Displays only specifi c columns listed in format**

# Basic Commands(4)

**-y Prevents display of process flags**

**-Z Displays the security context information**

**-H Displays processes in a hierarchical format (showing parent processes)**

**-n namelist Defines the values to display in the WCHAN column**

**-w Uses wide output format, for unlimited width displays**

**-L Shows process threads**

**-V Displays the version of ps**

# Basic Commands - Signals

**1 HUP Hangs up**

**2 INT Interrupts**

**3 QUIT Stops running**

**9 KILL Unconditionally terminates**

**11 SEGV Produces segment violation**

**15 TERM Terminates if possible**

**17 STOP Stops unconditionally, but doesn't terminate**

**18 TSTP Stops or pauses, but continues to run in background**

**19 CONT Resumes execution after STOP or TSTP**

*those signals work with the *pid* of the *ps command*.

# Basic commands - Tools

$ sudo killall apt apt-get : A useful trick for downloading

$ grep katipouthelo myfile

| | |
|---|---|
| **-v** | **reverse the search (output lines that don't match the pattern)** |
| **-n** | find the line numbers where the matching patterns are found |
| **grep -e t -e f file1** | Specify each individual pattern |
| **[at]** | |

$ git clone https://github.com/libgit2/libgit2

# Basic commands – Tools(2)

**Extract (tar)**

- Tar -zxvf
- z means (un)zip.
- x means extract files from the archive.
- v means print the filenames verbosely.
- f means the following argument is a filename.

$ tar –zxf filename

# Useful Paths

**/etc/passwd file :** The Linux system uses a special file to match the login name to a corresponding UID value. This file is the /etc/passwd file. The /etc/passwd fi le contains several pieces of information about the user.

- The login username
- The password for the user
- The numerical UID of the user account
- The numerical group ID (GID) of the user account
- A text description of the user account (called the comment field)
- The location of the HOME directory for the user
- The default shell for the user

# Useful Paths(2)

/etc/shadow file : The /etc/shadow file provides more control over how the Linux system manages passwords. Only the root user has access to the /etc/shadow file, making it more secure than the /etc/passwd file.

- The login name corresponding to the login name in the /etc/passwd file
- The encrypted password
- The number of days since January 1, 2000, that the password was last changed
- The minimum number of days before the password can be changed
- The number of days before the password must be changed
- The number of days before password expiration that the user is warned to change the password
- The number of days after a password expires before the account will be disabled
- The date (stored as the number of days since January 1, 2000) since the user account was disabled
- A field reserved for future use

# Permissions

-rw-rw-r-- 1 rich 50 2010-09-13 07:49 file1.gz

The first field in the output listing is a code that describes the permissions for the fi les and directories. The first character in the field defines the type of the object:

- - for files
- d for directories
- l for links
- c for character devices
- b for block devices
- n for network devices

# Permissions(2)

After that, you see three sets of three characters. Each set of three characters defines an

access permission triplet:

- r for read permission for the object
- w for write permission for the object
- x for execute permission for the object

If a permission is denied, a dash appears in the location. The three sets relate the three

levels of security for the object:

- The owner of the object
- The group that owns the object
- Everyone else on the system

# Permissions(3)

**Why $ chmod *777* myfile?**

Permissions Binary Octal Description

- --- 000 0 No permissions
- --x 001 1 Execute-only permission
- -w- 010 2 Write-only permission
- -wx 011 3 Write and execute permissions
- r-- 100 4 Read-only permission
- r-x 101 5 Read and execute permissions
- rw- 110 6 Read and write permissions
- rwx 111 7 Read, write, and execute permissions

$ chown owner.group myfile : change the owner

# Install packages

- apt-get install aptitude
- aptitude search packageName
- aptitude full-upgrade
- aptitude dist-upgrade
- aptitude safe-upgrade
- aptitude install package_name

# Creating a Script File

1. Create a script

```
#!/bin/bash
# This script displays the date and who's logged on
date
who
```

2. Δημιουργούμε το αρχείο dokimi.sh
3. Chmod *777* dokimi.sh
4. ./dokimi.sh

Μπορεί να χρειαστεί να δηλώσουμε το path : export PATH="$PATH:~/scripts"

# WIRESHARK

Το Wireshark αποτελεί τον πιο διάσημο και ευρέως χρησιμοποιούμενο αναλυτή πρωτοκόλλων δικτύου. Είναι δωρεάν χωρίς εμπορικές εκδόσεις. Και δίνει την δυνατότητα ανάλυσης της κίνησης του δικτύου σε βάθος. Αναγνωρίζει σχεδόν όλα τα πρωτόκολλα που υπάρχουν και υπάρχει υποστήριξη αποκρυπτογράφησης για πολλά πρωτόκολλα, όπως τα IPsec, ISAKMP, Kerberos, SNMPv3, SSL / TLS, WEP και WPA / WPA2.

# Wireshark

| Port | Service name | Transport protocol |
|---|---|---|
| 20, 21 | File Transfer Protocol (FTP) | TCP |
| 22 | Secure Shell (SSH) | TCP and UDP |
| 23 | Telnet | TCP |
| 25 | Simple Mail Transfer Protocol (SMTP) | TCP |
| 50, 51 | IPSec | |
| 53 | Domain Name System (DNS) | TCP and UDP |
| 67, 68 | Dynamic Host Configuration Protocol (DHCP) | UDP |
| 69 | Trivial File Transfer Protocol (TFTP) | UDP |
| 80 | HyperText Transfer Protocol (HTTP) | TCP |
| 443 | HTTP with Secure Sockets Layer (SSL) | TCP and UDP |

# Wireshark

- Physical (e.g. utp stp cables ,fiber)
- Data Link (e.g. MAC, switches)
- Network (e.g. IP, routers)
- Transport (e.g. TCP, UDP, port numbers)
- Session (e.g. Syn/Ack)
- Presentation (e.g. encryption, ASCII, PNG, MIDI)
- Application (e.g. SNMP, HTTP, FTP)

# WireShark

## Χρωματικοί συνδυασμοί

Αρχικοί χρωματισμοί

- Γκρι - πακέτα TCP
- Μαύρο με κόκκινα γράμματα - Πακέτα TCP με σφάλματα
- Πράσινο - Πακέτα HTTP
- Ανοιχτό μπλε - Πακέτα UDP
- Ανοικτό μπλε - Πακέτα ARP
- Λεβάντα - Πακέτα ICMP
- Μαύρο με πράσινα γράμματα - Πακέτα ICMP με σφάλματα

Οι χρωματισμοί μπορούν να αλλάξουν μέσω *View -> Coloring Rules*

# WireShark

- Μία καταγραφή πακέτων περιέχει συνήθως πολλά πακέτα που δεν σχετίζονται με τον σκοπό ή και τον στόχο της ανάλυσης
- Για να αφαιρεθούν αυτά τα πακέτα από την οθόνη ή από το αρχείο που μπορούμε να δημιουργήσουμε … το Wireshark παρέχει τη δυνατότητα δημιουργίας φίλτρων.
- Τα φίλτρα υλοποιούνται για κάθε πακέτο μεμονωμένα.
- Υπάρχουν εκφράσεις Boolean που έχουν να κάνουν με τις ιδιότητες των πακέτων.
- Υποστηρίζει κανονικές εκφράσεις.
- Μπορούν είτε να κατασκευαστούν χειροκίνητα, είτε να συντεθούν μέσω του Expressions ή να συντίθεται με βάση κάποιες επιλεγμένες ιδιότητες του πακέτου.

# WireShark

**Σύνθετα φίλτρα**

- Τα φίλτρα μπορούν να αποτελούνται από πολλαπλές εκφράσεις που ενώνονται με boolean συνδέσμους.
    - && - λογική σύνδεση (π.χ. AND)
    - || - λογική διάζευξη (π.χ. OR)
    - ! - λογική άρνηση (π.χ. NOT)

- Υποστηρίζει τη σειρά των πράξεων.

**Κανονικές εκφράσεις**

- Τα πεδία μπορούν να αξιολογηθούν βάσει μιας κανονικής έκφρασης χρησιμοποιώντας την εντολή "matches".
- Χρησιμοποιεί τη σύνταξη regex της Perl.

# WireShark

**Πλαίσιο κειμένου φίλτρου**

- Πράσινο - έγκυρο φίλτρο
- Κόκκινο - άκυρο φίλτρο
- Κίτρινο - μπορεί να παράγει απροσδόκητα αποτελέσματα

**Φίλτρα βασισμένα σε πακέτα**

- Τα φίλτρα μπορούν να κατασκευαστούν με βάση μεμονωμένα πακέτα κάνοντας δεξί κλικ σε ένα πακέτο και επιλέγοντας το:
    - *Prepare as filter*- δημιουργεί ένα φίλτρο.
    - *Apply as filter*- δημιουργεί ένα φίλτρο και το εφαρμόζει στο ίχνος.
    - *Follow TCP Stream* - δημιουργεί ένα φίλτρο από το πακέτο TCP ροής και το εφαρμόζει στην ανίχνευση.

# WireShark

***not(tcp.port==80) and !(tcp.port==443)***

Εξαιρούμε τα πακέτα που αφορούν κίνηση στην πόρτα 80 (HTTP) και στην πόρτα 443 (HTTPS - HTTP πρωτόκολλο με TLS/SSL για κρυπτογραφημένη κίνηση)

~~***not(tcp.port==80) and !(tcp.port==443) and !dns and !arp***~~

Εξαιρούμε τα πακέτα που αφορούν κίνηση στην πόρτα 80 και στην πόρτα 443 και τα πακέτα που αφορούν το dns και το arp πρωτόκολλο.

# WireShark

***(not(tcp.port==80) and !(tcp.port==443) and !(udp.port==53) and !arp && !(ipv6.src == 2606:YYY0:20::YYYa:bf0) ) && !(ipv6.dst == 2606:47YYY00:::681a:bf0)***

Ένα φίλτρο που εκτός από πρωτόκολλα εξαιρεί και συγκεκριμένες IP
και στο επίπεδο του αποστολέα αλλά και του παραλήπτη

# NETCAT

Το Netcat είναι ένα εξαιρετικά ευέλικτο εργαλείο το οποίο έχει χαρακτηριστεί ως "ελβετικός σουγιάς". Το Netcat λοιπόν είναι ένα εργαλείο δικτύωσης υπολογιστών για την ανάγνωση και την εγγραφή συνδέσεων δικτύου χρησιμοποιώντας TCP ή UDP · αυτή η διπλή λειτουργία υποδηλώνει ότι το Netcat εκτελείται σε δύο λειτουργίες: "client" και "server". Επιπλέον έχει σχεδιαστεί για να είναι μια αξιόπιστη "back-end" συσκευή που μπορεί να χρησιμοποιηθεί από άλλα προγράμματα και σενάρια. Ταυτόχρονα, είναι ένα πλούσιο σε χαρακτηριστικά εργαλείο εντοπισμού σφαλμάτων και διερεύνησης δικτύου, καθώς μπορεί να παράγει σχεδόν οποιοδήποτε είδος συσχέτισης που θα χρειαστείτε και έχει πολλές ενσωματωμένες δυνατότητες.

Η λίστα των χαρακτηριστικών περιλαμβάνει τη σάρωση θύρας, τη μεταφορά αρχείων και την ακρόαση θύρας επίσης μπορεί να χρησιμοποιηθεί και ως backdoor.

# NETCAT

LAB 1 : Listening on a TCP/UDP port with Netcat

From kali1Machine we want to listen on a port (4444) and accept incoming connections, and after that there will be a connection to kali2Machine

Commands:

- nc -lvp 4444 : kali1 listen on port 4444 and accept incoming connections on this port
- nc -vv 10.10.136.85 4444 : connect to port 4444 from kali2 to kali1

# NETCAT

LAB 2 : Transferring files with Netcat

Netcat can also be used to transfer files from one computer to another. This applies to text and binary files

Commands:

- nc -lvp 4444 > dokimi1.txt :  listen to and accept the connection and to redirect any input into a file.type
- nc -vv 192.168.129.1 4444 < dokimi2.txt : we connect to  listening Netcat on kali1 (port 4444) and send the file

# NETCAT

LAB 3 : Remote Administration with Netcat

One of Netcat's features is command redirection. This means that Netcat can take an exe file and redirect the input, output and error messages to a TCP/UDP port, rather than to the default console.

Commands :

- nc -lvp 4444 -e /bin/bash : so that Anyone connecting to port 4444 on this machine will be presented with command prompt, with the permissions that nc was run with.(kali1)
- nc -v 10.10.36.144 4444 : connect to kali1 shell from kali2

# NMAP

Το Nmap ("Network Mapper") είναι ένα βοηθητικό open source πρόγραμμα για την ανίχνευση του δικτύου και τον έλεγχο ασφαλείας. Είναι χρήσιμο εργαλείο για εργασίες όπως απογραφή δικτύου, διαχείριση αναβαθμίσεων υπηρεσιών και παρακολούθηση χρόνου λειτουργίας του κεντρικού υπολογιστή ή της υπηρεσίας. Το Nmap μπορεί να βρει ποιοι κεντρικοί υπολογιστές είναι διαθέσιμοι στο δίκτυο, ποιες διαδικτυακές πόρτες, ποιες υπηρεσίες (όνομα και έκδοση εφαρμογών) προσφέρουν οι χρήστες, ποια λειτουργικά συστήματα (και εκδόσεις λειτουργικών συστημάτων) εκτελούν, ποιο είδος φίλτρων πακέτων / τείχη προστασίας χρησιμοποιούνται και δεκάδες άλλα χαρακτηριστικά. Σχεδιάστηκε για γρήγορη σάρωση μεγάλων δικτύων, αλλά λειτουργεί καλά ενάντια σε μεμονωμένους κεντρικούς υπολογιστές. Το Nmap λειτουργεί σε όλα τα μεγάλα λειτουργικά συστήματα όπως Linux, Windows και Mac OS X.

# NMAP

Scan IP address (Targets) :

| Command | Description |
|---|---|
| nmap 10.0.0.1 | Scan a single host IP |
| nmap 192.168.10.0/24 | Scan a Class C subnet |
| nmap 10.1.1.5-100 | Scan the range of IPs between 10.1.1.5 up to 10.1.1.100 |
| nmap -iL hosts.txt | Scan the IP addresses listed in text file “hosts.txt” |
| nmap 10.1.1.3 10.1.1.6 10.1.1.8 | Scan the 3 specified IPs only |
| nmap www.istoselida.com | First resolve the IP of the domain and then scan its IP address |

# NMAP

Port Related Commands :

| Command | Description |
|---|---|
| nmap -p80 10.1.1.1 | Scan only port 80 for specified host |
| nmap -p20-23 10.1.1.1 | Scan ports 20 up to 23 for specified host |
| nmap -p80,88,8000 10.1.1.1 | Scan ports 80,88,8000 only |
| nmap -p- 10.1.1.1 | Scan ALL ports for specified host |
| nmap -sS -sU -p U:53,T:22 10.1.1.1 | Scan ports UDP 53 and TCP 22 |
| nmap -p http,ssh 10.1.1.1 | Scan http and ssh ports for specified host |

# NMAP

Different Scan Types, Versions of Services and Operating Systems :

| Command | Description |
|---|---|
| nmap -sS 10.1.1.1 | TCP SYN Scan (best option) |
| nmap -sT 10.1.1.1 | Full TCP connect scan |
| nmap -sU 10.1.1.1 | Scan UDP ports |
| nmap -sP 10.1.1.0/24 | Do a Ping scan only |
| nmap -Pn 10.1.1.1 | Don't ping the hosts, assume they are up |
| nmap -sV 10.1.1.1 | Version detection scan of open ports (services) |
| nmap -O 10.1.1.1 | Identify Operating System version |
| nmap -A 10.1.1.1 | This combines OS detection, service version detection, script scanning and traceroute. |

# NMAP

| Command | Description |
|---|---|
| nmap -T0 10.1.1.1 | Slowest scan (to avoid IDS) |
| nmap -T1 10.1.1.1 | Sneaky (to avoid IDS) |
| nmap –T2 10.1.1.1 | Polite (10 times slower than T3) |
| nmap –T3 10.1.1.1 | Default scan timer (normal) |
| nmap –T4 10.1.1.1 | Aggressive (fast and fairly accurate) |
| nmap –T5 10.1.1.1 | Very Aggressive (might miss open ports) |
| nmap -oN [filename] [IP hosts] | Normal text format |
| nmap -oG [filename] [IP hosts] | Grepable file (useful to search inside file) |
| nmap -oX [filename] [IP hosts] | XML file |
| nmap -oA [filename] [IP hosts] | Output in all 3 formats supported |

# NMAP

| Command | Description |
| --- | --- |
| nmap -PS22-25,80 10.1.1.0/24 | Discover hosts by TCP SYN packets to specified ports (in our example here the ports are 22 to 25 and 80) |
| nmap -Pn 10.1.1.0/24 | Disable port discovery. Treat all hosts as online. |
| nmap -PE 10.1.1.0/24 | Send ICMP Echo packets to discover hosts. |
| nmap -sn 10.1.1.0/24 | Ping scan. |
| nmap --script=”name of script” 10.1.1.0/24 | Run the specified script towards the targets |
| nmap --script=”name of script” --scriptargs=”argument=arg” 10.1.1.0/24 | Run the script with specified arguments |
| nmap --script-updatedb | Update script database |

# NMAP

Perform Full Port Scan using the Live Hosts List :

nmap -p- -Pn -sS -A -T4 -iL livehosts.txt -oA MyFile

- -p- : This scans all ports
- -Pn : Do not perform host discovery again
- -sS : Perform TCP SYN scan
- -A : This combines OS detection, service version detection, script scanning and traceroute
- -T4 : Pretty fast and accurate scanning
- -iL livehosts.txt : Scan the IPs contained in file “livehosts.txt”
- -oA : Export the results in file “MyFile”

# NMAP

Find well known vulnerabilities related to an open port :

STEP 1: download the “nmap-vulners” script from Git and place it under the script directory of nmap:

```
# cd /path/paths/scripts
# git clone https://github.com/vulnersCom/nmap-vulners.git
```

STEP 2 :

```
# nmap -Pn -sV -p80 --script=vulners scanme.nmap.org
```

There you will find CVE codes of the vulnerabilities

# NMAP

Find the list of IP addresses that are currently in the network!

```
nmap -sP 192.168.233.139/24 | awk '/is up/ {print up}; {gsub (/\(|\)/,""); up = $NF}'
```

# Bettercap

# Bettercap

$apt install bettercap

$bettercap

>>net.probe on >>net.show

>>set arp.spoof.targets 192.168.233.139

>>arp.spoof on

>>net.sniff on >>net.sniff off

Open the 192.168.233.139 browser >>clear