POLITECNICO DI TORINO

Fundamentals of Information Systems Security

Student:

Gianmarco Michelini

Academic Year 2024/2025

All notes are derived from oral presentations and written papers.

Contents

List of Figures

List of Tables

Chapter 1

LABoratories

1.1 First LAB

[7]

1.1.1 Software Tools

Tool	Description
nmap	It is designed to perform quick scanning of large networks.
Ettercap	Allows performing man-in-the-middle (MITM) attacks and sniffing attacks in a Local Area Network (LAN).
Wireshark	Allows to capture network traffic.
GVM	Performs vulnerability scanning.

Table 1.1: Main Software Tools

Command	Description
Apache2	Web server.
VSFTP	FTP server.
SSH2	SSH server
Exim	Mail server

Table 1.2: Additional Software Tools

Commands for Softwares

Command	Description	Options
sudo systemctl [options] apache2	Configuration of server ports are at: /etc/apache2/ports.confs	start; stop; restart
systemctl [options] vsftpd	Configuration of server ports are at: /etc/vsftpd.conf	start; stop; restart
sudo systemctl [options] ssh **Before starting the server the first time you must generate the keys of the host with the command:** `ssh-keygen -A`	Configuration of server ports are at: /etc/ssh/sshd_config	start; stop; restart
sudo systemctl [options] exim4	Configuration of server ports are at: /etc/default/exim4.	start; stop; restart
sudo wireshark	Network analyzer	

Table 1.3: Softwares Commands

1.1.2 Commands for Networking

Command	Description	Options
arp	Manipulets the system ARP cache.	-d(elete) <ipAddr>; Show: -e (fixed); -a
ip	Analyse and manipulate the routing of IP packets	neigh flush all (delete); -s -s neigh flush all (all-verbose)
netstat	Displays detailed information about a network.	-l(istening) -t(cp); -u(dp)
nmap <ipAddrVictim>	Obtain information of a victim in the network	-O(S); -sT(CP); -Pn (ping) -p <port>; -v(erbose); -sV (service/Version) -T<num> (timer 0-6) -A(ggressive)
ettercap configuration files: `man etter.conf` `/etc/ettercap/etter.conf`.	Executes man-in-the-middle attacks in a LAN.	-T(ext UI); -q(uiet) -M(ITM) [arp;icmp;dhcp;port] -e "<regExpr>" -L <logFile>; -P(lugin)
s-nail	Mail Client. Send and receive Internet mail	-s(ubject); -S(et var)

Table 1.4: Network Commands

Insights

- Network Fingerprinting allows obtaining information on the remote host's operating system. Possible using the tool nmap. This technique is based on the fact that different types of operating systems implement differently the TCP/IP stack →nmap.
- The technique known as Port Scanning is used to obtain information about which ports of a particular host are open →nmap.
- A special expression can be used with the port parameter as `-p <initial>-<ending>`
- Identification of services →nmap or a vulnerabilities scanner like GVM (Uses an in-depth scanning).
- MITM attacks (like ARP poisoning) →ettercap.

Commands from the Text

```
#FINGERPRINTING
#Bob(attacker) tries to establish a TCP connection (-sT) on the port 80 (-p 80) of the target host Alice, in order to obtain information about the operating system (-O) running on the victim's machine
nmap -sT -p 80 -O -v <ipAddrVictim>

#PORT SCANNING
#the attacker wants to scan ports of the victim that use tcp connections. Scan the first 1024 ports
nmap -sT -p 1-1024 -v <ipAddrVictim>
#version with ping interaction
nmap -Pn -p 1-1024 -v <ipAddrVictim>

#IDENTIFICATION of SERVICES
#the attacker wants to identify the (application) services running on the open ports on victims's machine
nmap -sV -Pn -p 1-1024 -v <ipAddrVictim>
#or a more aggressive version:
nmap -sV -A -Pn -p 1-1024 -v <ipAddrVictim>
#-A: Enable OS detection, version detection, script scanning, and traceroute
#more information are provided on the target machine!!

#ARP poisoning
ettercap -Tq -M arp /<ipAddrVictim1>// /<ipAddrVictim2>//
#in addition with regular expression
ettercap -Tq -M arp /<ipAddrVictim1>// /<ipAddrVictim2>// -e "<regExpr>"
```

1.1.2.1 How to use Mail Server - exim

```
#mail server configuration
dpkg-reconfigure exim4-config

#start the mail server
sudo systemctl start exim4

#another user sends an e-mail to the mail server
    #follow the configuration details reported below
#all on the same line
s-nail -S mta=smtp://10.0.24 -S 'from=<userMittent> \\ @kali' -
   s "<subjectText>" <userReceiver>@kali
#press enter when finished and then ctrl-D
```

1. Alice configures the `exim` mail server with the command:

   ```
   dpkg-reconfigure exim4-config
   ```

   and by selecting the parameters in the following manner:

   (a) General type of mail configuration: Internet site; mail is sent and received directly using SMTP.
   (b) System mail name: kali
   (c) IP-addresses to listen on for incoming SMTP connections: *// leave blank (delete data if present)*
   (d) Other destinations for which mail is accepted: kali
   (e) Domains to relay mail for: *// leave blank (delete data if present)*
   (f) Machines to relay mail for: *// leave blank (delete data if present)*
   (g) Keep number of DNS-queries minimal (Dial-on-Demand) ?: No
   (h) Delivery method for local mail: mbox format in /var/mail
   (i) Split configuration into small files ? : No
   (j) Root and postmaster mail recipient: *// leave blank (delete data if present)*

Figure 1.1: Configuration parameters for exim mail server in the LAB

1.2 Second LAB

[8]

1.2.1 OpenSSL Commands

Insights

- 1 Byte = 2 HEX characters.
- In order to decrypt a file you need to know: iv, K and cipher algorithm.
- In practice, if you have an N-Bytes RSA key, you can perform successfully encryption/decryption operations with OpenSSL only if the (plaintext) data is at most N-11 bytes long.
- RSA-encrypt →public key.
- RSA-decrypt →private key.
- RSA-sign →private key.
- RSA-verify →public key.
- The `pubin` parameter is used to specify that the input key it has to be a public key.

1.2.2 Utility Commands

Command	Description	Options
`systemctl [options] ssh`	Must be enabled on the Receiver **Remember to stop it at the end**.	start ; stop ; restart enable ; status
`scp <user>@<ipReceiver>:<dirFullName>`	Transfers a file to the specified user's directory	-r(ecursive)
`openssl rand -out <outputFile> <numBytes>`	Creates a file numBytes long.	
`time <openssl_command>`	Measures the elapsed time of a command.	
`expr <arg1> <basicOperation> <arg2>`	Performs basic operations. Such as: * / + -	
`wget <URL>`	For non-interactive download of files from the Web.	
`atril <fileName> &`	A simple multi-page document viewer.	
`sha1sum`	Easy computation of the hash of one or more files.	
`hashdeep <file/dirName>`	Easy computation of the hash of one or more files. Processes recursively the files contained in a directory with a chosen algorithm.	-r; -c <dgstAlgorithm> -m (match) -x (negative match) -k <fileName> (for m or x)

Table 1.6: Utility Commands

Command	Description	Options
`man openssl <command>`		
`openssl enc`	Allows the encryption and decryption of data with several symmetric cipher routines.	-help, -ciphers, -p(rint); -<algorithm>; -nopad; -K <hexKey>; -iv <hexVector> -in <inputFile>; -out <outputFile> -iter <n>; -pbkdf2; -nosalt -e (default); -d;
`openssl rand <numBytes>`	Generates nBytes pseudo-random data.	-hex -out <outputFile>
`openssl genrsa <numBits>`	Performs simple asymmetric (key pair) operations with the RSA algorithm.	-out <outputFile>
`openssl rsa`	To manage and use the RSA keys in cryptographic operations.	-in <inputFile>; -out <outputFile> -text; -noout -pubin; -pubout
`openssl ecparam`	To manage and manipulate the EC algorithm parameters.	-list_curves -name <curveName>; -genkey -out <outputFile>
`openssl ec`	To manage and manipulate the EC algorithm keys.	-in <inputFile>; -out <outputFile> -pubin; -pubout -text
`openssl pkeyutl` **Supported algorithms:** RSA, DSA, Diffie-Hellmann and Elliptic Curve. **The order in which the parameters are passed is important.**	Performs asymmetric encryption/decryption, signature/verification, and key exchange, by using various asymmetric algorithms.	-encrypt; -decrypt; -sign; -verify; -verifyrecover -in <inputFile>; -out <outputFile> -pubin; -inkey <keyFile> -sigfile <signatureFile> (verify) -derive (shared secret); ↷ -peerkey <key_file>
`openssl dgst <inputFile>`	Allows to calculate the digest of data using different algorithms.	-list; -<algorithm>; -out <outputFile> -verify <pub_key> -signature <sig_file>
`openssl speed`	Measures the performance of the various algorithms implemented by OpenSSL	-evp (ctr)

Table 1.5: `openssl` commands

Insights

- File-transfer protocol: enable ssh server on the receiver (remember to stop it at the end), send the file from the mittent with scp tool.
- Command: `scp <fileName> <user>@<ipReciever>:/home/<user>/Desktop`
- `scp`: in my case user=Alice or Bob, with their ip provided from `ifconfig`, password=0000

Symmetric Encryption

```
#encrypt ptext using aes 128bit-key with cbc mode
openssl enc -in ptext -e -out ctext.aes128 -aes-128-cbc -nosalt
  #the symmetric is derived from a password, with no password:
openssl enc -in ptext -out ctext.aes128 -aes-128-cbc -nosalt -K
   00112233445566778899aabbccddeeff -iv 00112233445566778899
   aabbccddeeff -p
```

Operations with Digests

```
#generate hashes for the files within the "tree" directory and save
   them to hash_list
hashdeep -c sha256 -r tree > hash_list
#check for differences on the same files
hashdeep -c sha256 -r -x -k hash_list tree
```

Operations on Key Pair

```
#create a key pair and save them to a file
openssl genrsa -out rsa.key.Alice 2048
#read the key file
openssl rsa -in rsa.key.Alice -text
#extract only the public key and save it to a file
openssl rsa -in rsa.key.Alice -out rsa.pubkey.Alice -pubout

#encrypt a plain text with a public key
openssl pkeyutl -encrypt -in plain -out encRSA -pubin -inkey rsa.key.
   Alice
#decrypt a cipher text encrypted with RSA, knowing the private key
penssl pkeyutl -decrypt -in plain.enc.RSA.for.Alice -inkey rsa.key.
   Alice

#sign a file ("plain") using the private key of Alice
openssl pkeyutl -sign -in plain -inkey rsa.key.Alice -out sig.Alice
#verify the signature (on the file "plain") using the public key of
   Alice
openssl pkeyutl -verify -in plain -pubin -inkey rsa.key.Alice -
   sigfile sig.Alice

#generate a SECG curve over a 192 bit prime field and save it to a
   file
openssl ecparam -name secp192k1 -genkey -out ec.key.Alice
#extract the ec public key from a file and save to another file
openssl ec -in ec.key.Alice -pubout -out ec.pubkey.Alice

#sign a file ("plain") with  ECDSA and save the signature to a file
openssl pkeyutl -sign -in plain -inkey ec.key.Alice -out ecsig
#verify the signature of the file signed with ECDSA
openssl pkeyutl -verify -in plain -pubin -inkey ec.pubkey.Alice -
   sigfile ecsig
```

Symmetric Algorithms Performances

Be aware: The real time reported by the time command in the table 1.7 refers to the elapsed wall clock time — the total time from when the command starts executing to when it finishes.
Creating files: `openssl rand -out <outputFile> <numBytes>`.
Measuring elapsed time: `time <opensslEncryptionCommand>`.

	100 B	10 kB	1 MB	100 MB
des-ede3	0.01 s	0.01 s	0.11 s	9.91 s
aes-128-cbc	0.01 s	0.01 s	0.11 s	10.21 s
aes-192-cbc	0.01 s	0.01 s	0.11 s	10.48 s
aes-256-cbc	0.01 s	0.01 s	0.11 s	10.37 s
aes-128-ctr	0.01 s	0.01 s	0.14 s	10.39 s
chacha20	0.01 s	0.01 s	0.12 s	9.18 s

Table 1.7: Performance of some symmetric encryption algorithms.

Digest Algorithms Performances

	100 B	10 kB	1 MB	100 MB
sha256	0.01 s	0.01 s	0.01 s	0.12 s
sha512	0.01 s	0.01 s	0.02 s	0.15 s

Table 1.8: Costs associated with some digest algorithms

Bibliography

[1] Antonio Lioy. Introduction to cybersecurity, 2024.

[2] Antonio Lioy. Cryptographic techniques for cybersecurity, 2024.

[3] Giuseppe Tipaldo. Cybersecurity and society, 2024.

[4] Antonio Lioy. Security of ip networks, 2024.

[5] Antonio Lioy. Firewall and ids/ips, 2024.

[6] Antonio Lioy. Security of network applications, 2024.

[7] Antonio Lioy. Lab, network security, basic attacks, 2024.

[8] Antonio Lioy. Lab, cryptography with openssl, basic operations, 2024.

[9] Antonio Lioy. Lab, cryptography with openssl - applications, 2024.