**INTRO**

site : https://keller.lc/ece/

1h30 final exam

QCM and open questions

Different audit types

A security audit gives an independent and impartial opinion about the security of the analysed perimeter at a precise time.

Goal is to check the policy and procedures to keep the SI :

- in compliance with the threat model of the auditee

- in compliance with the actual standards

- in accordance with the technical measures in place

- efficiency applied

Documents :

- ISSP

- Access control policy

- Risk analysis methdology

- Other docs for the SI security and access control

Interviews to make :

- Chief information security officer

- Chief information officer

- Operator of the security solutions and services

- Physical security officer

ISO 27001 and 27002 are security norms

Architecture audit :

check if there is a VPN, firewall, antivirus, everything is update etc

Documents :

- Technical architecture file

- Network equipement configuration

- OSI level 2 and 3 diagrams

- Interconnection with another network

- Risks analysis

- Procedures of the analysed perimeter

- Previous architecture questions

Interviews :

- System and network infrastrucutre manager

- System and network administrators

---------------------------------------------------------------------------------------------------------------------------

**CRYPTO**

Feistel cypher :

Zk = Rk XOR K

Rk+1 = Lk XOR Zk

Lk+1 = Rk

Then Rk becomes Lk

And Lk becomes Rk

Modern cryptography

Symmetric-key cryptography

- CBC = concept of a plain text won't be ciphered the same way twice

Public-key cryptography

- Receiver send an open lock to Sender without the key

- Sender send the message in the lock

- Receiver open the lock

RSA : factoring large integers

- 2 primes numbers (big ones in real life) (2048 bits recommended) : p and q

- n = p\*q

- Phi = (p-1)(q-1)

- We search for e and d in order that : ed = 1 (mod Phi) to construct public and private keys.

- Public Key = RSA(n,e) ; Private Key = RSA(n,d)

- Someone send me : message\*\*e%n

- I receive the cypher and I do 46\*\*d%n to retreive the plain text

We also can assure the identity of the sender

- The sender sends : message\*\*d%n

- The receiver retreives by doing : message\*\*e%n

ord('letter') gives the ASCII code of this letter

chr(ASCII) gives back the letter

BOB sends :

- send his message : message\*\*(Alice\_e)%(Alice\_n)

- send his signature : cypher\*\*(Bob\_d)%(Bob\_n)

ALICE

- get the message with : message\*\*(Alice\_d)%(Alice\_n)

- get the signature with : message\*\*(Bob\_e)%(Bob\_n)

To ensure that there is no man in the middle :

We have to ensure the public key is really BOB's public key. You can meet BOB and ask him.

But you can also trust the certificate authority that garanty you that the public key is the BOB's public key. They use challenges and checks to ensure it.

We can find certificates on your browser.

---------------------------------------------------------------------------------------------------------------------------

**LINUX**

whoami : affiche le nom d'utilisateur courant

id : renvoie l'utilisateur ID uid (identifiant unique)

renvoie group id gid

renvoie les groupes dont je fais partie

ls : montre les fichiers dans le dossier actuel

ls -a : renvoie également les fichiers cachés

ls -l : liste le contenu avec plus de détails (droits, groupes, taille, date modification)

ls -h : human readable (le taille sont donnée avec Kilo, Mega etc)

tree : permet de voir l'arborescence des fichiers à partir du dossier actuel

tree -L 2 : permet de limiter au 2ème niveau de l'arborescence

tree -d : permet d'afficher que les dossiers

cd : change directory

pwd : montre le chemin dans lequel on est

grep : permet de chercher des chaîne de caractères dans les fichiers : grep 'password' ./\* : cherche le mot password dans tous les fichiers du dossier actuel

grep -i : case sensitive

grep -R : cherche dans tous les sous dossiers de manière récursives

grep -l : affiche les fichiers dans lesquels le mot est présent

Directories

/bin : essential user command binaries

/etc : configuration files

/sbin : essential system command binaries

/usr : read-only user application support and binary

/var : variable data files

/dev : device files

/home : user home directories

/lib : librairies and kernel modules

/mnt : mount files and temporary files

/opt : optional software application

/proc : process kernel and information files

/root : home for the root user

/home/user/ dossier intéressants pour l'élévation de privilèges

.bashrc : configuration du terminal

.bash\_history : historique des commandes tapées

.ssh/ : dossier de configuration pour SSH. On peut trouver les clefs SSH de l'utilisateur

other configuration files : on peut accéder au cache des applications

$PATH

is an environment variable (on doit mettre $ pour l'afficher)

donne des chemins d'accès à des répertoires (séparés par :)

quand on appel un éxécutable, on parcours le PATH jusqu'à temps de trouver l'éxécutable dans les chemins

which id donne le chemin d'un executable

ajouter le local folder au PATH : PATH = .:$PATH

Files permissions

d | RW- | RW- | R-X : donne des infos sur le type du file (d=directory) | user | group | others

R : read

W : write

W : execute

R W X = 4 2 1

Les permissions peuvent se résumer avec des nombres

RW- | RW- | R-X : 6 | 6 | 5 : chmod 665

droit d'execution dans un dossier permet de le traverser et de l'ouvrir

droit d'écriture

find / -perm -o=r (ou -444) -type f -print : on cherche les fichiers dont les permissions sont au moins un read dans tous les fichiers

SUID SGID

set user identifier

set group identifier

dans les privilèges on peut retrouver la lettre s qui indique que c'est SUID ou GUID.

s indique que lors de l'éxécution, les droits du propriétaires du fichier sont utilisés

dans la notation avec les numéros pour les permissions, il peut y avoir un 4ème chiffre devant les chiffres de permissions habituels. Ce 4ème chiffre (en 1ère place) indique un SUID, GUID ou sticky:

2 : GUID

4 : SUID

1 : sticky bit

sudo

permet d'éxécuter des programmes avec d'autres privilèges que les notres

sudo -l donne les droits que l'utilisateur a en utilisant la commande sudo

Capabilities

permet de faire des droits plus fins sur les programmes

CAP\_DAC\_OVERRIDE : override R/W/X permission checks

CAP\_KILL : bypass permission checks for sending kill signals

CAP\_SETFCAP : can set capabilities everywhere

Pour lister les capabilities : getcap -r /

cap\_net\_rap+ep : e dit que c'est effective

p dit que c'est permitted

/etc/passwd

root:x:0:0:root:/root:/bin/hash : user : password (x indique qu'il est socké ailleurs) : UID : GUID : comment : user's home : user's default shell

Tout le monde a les droits de lecture ici

/etc/shadow

user:hash du mot de passe:last password change since January 1 1970: minimal password age : maximum password age : nb of days to warn before password's expiration : nb of days whithin the password can be used after expiration : date d'expiration password

le groupe shadow peut lire ce fichier et accéder aux hashs

hashs de etc/shadow

les champs sont séparés par $

$algorith used $salt (valeur qui est ajouté et qui permet d'avoir des hashs différents, tous les utilisateurs ont un salt différent) $ le hash

---------------------------------------------------------------------------------------------------------------------------

**HASHES**

Hash function

One way cypher function

Take an input of any length

Output is at a fixed length

Avoid collisions ( 2 inputs have the same output )

Stirct avalanche criterium ( changing 1 bit in the input have to change >50% of the output )

Hashes exemple

SHA1

SHA2

MD5

LMH

BCrypt

Argon2

Hash usage for password

User enters a password

Directly hashed

Compared with hashes stocked in the database

for file integrity

Download a file

A SHA256 sum is given on the website

Once downloaded, we compute the SHA256 sum

We compare the hashes to ensure the integrity

for signature

to ensure the public key is the good one

How to crack hashes

Bruteforce

Dictionnary : file containing possible passwords. We hashes those most possible passwords and we compare to the database

Combining dictionnary attacks and rules attacks : take a dictionnary and generate password by adding characters to the dictionnary password (rules)

Mask attacks (hashcat) : Bruteforce equivalent but we specify a specific type order to bruteforce ( {A}{A}{A}{A}{N}{S} ; A = letters, N = numbers, S = special charatcers

Rainbow tables : on cherche le hash qu'on a dans cet immense fichier avec des correspondance ( Hash -> clearpassword)

John the ripper

In the hash file, type : user:hash for him to tell wich password we cracked

john hash.txt --format=Raw-MD5 --wordlist=dictionnary.txt --rules=AppendNumbers\_and\_Specials\_Simple

john hash.txt --format=Raw-MD5 --show permet d'afficher les mots de passe qu'on a déjà cracké avec ce format

Site web : korelogic.com permet d'accéder à toutes les rules et leurs effets

---------------------------------------------------------------------------------------------------------------------------

**WIRESHARK**

Statistic -> Protocol Hierarchy

Permet d'accéder à tous les protocoles dans les packets réseau capturés

Filter

irc

ip-addr=192.168.2.0

ip-dest=192.168.2.0

File -> export object

HTTP because the resquest we wanted are transmitted by HTTP

Download the archive file

Follow -> TCP stream

to access the discussion between the client and the server

cewl to build dictionnary from website

cewl http:// -d 0 -w dico.txt

/usr/share/john

some tools to extract hash

perl 7z2john.pl archive.7z > archive\_hash.txt

john archive\_hash.txt -w=dico.txt

----- def.pcap indication : host ssc.teaser.insomnihack.ch : 52.214.142.175 -----

SMTP : mail intéressant avec un nom de domaine

FTP : connexion avec usr, pwd et envoie d'une ssc.key

FTP data : ssc.key qui est une clef RSA

ip.addr ==52.214.142.175 && tls filter to find the right conversation

edit -> preferences -> TLS -> add RSA key. On peut maintenant décrypter la connexion chiffrée (TLS)

We know can access the TLS stream on the right server (given) and access the flag

---------------------------------------------------------------------------------------------------------------------------

**NETWORK**

nmap

scan network

hosts available

services available

nmap -A -T4 scanme.nmap.org -Pn

there are 2¹⁶ ports

Parameters

-sT : TCP connect

-sS : TCP SYN (SYN = message d'acquisition lors d'une connection TCP : --SYN--> / <--ACK-- ) running this command requires being root

-sU : UDP Scan

-sV : Probe open ports to determine service

- O : Enable OS detection

-A = -sV + -O + -sC

-p22,80,443 : scan specific ports

--top-ports N : scan les N ports les plus courants. 10 000 est souvent bien

-p- : scan all ports

Known ports (in /etc/services)

21/tcp : FTP

22 : SSH

23 : telnet

25 : smtp

80 : HTTP

443 : HTTPS

8080 : HTTP-ALT (souvent admin)

194 : irc

53/udp : DNS

88 : kerberos

123 : ntp

161 : snmp

SNMP (simple network management protocol)

communication protocol allowing admins réseau de gérer les équipements du réseau

3 versions :

- SNMPv1 : no real security, community string (identify you to make you access only certain data. But it can be bruteforce

- SNMPv2 : chiffrement, authentification, community string

- SNMPv3 : stronger security mechanism

SNMP exploit

snmpbulkwalk -c public -v2c 10.4.77.18

snmp-check 10.4.77.18

Bruteforce a SNMP community string

msfconsole --> we get in metasploit

search snmp --> list tools about SNMP

user auxiliary/scanner/snmp/snmp\_login --> change directory

show options --> give the different options and their basic value etc

set rhosts 10.4.77.18

run --> gives the different community string found

snmpbulkwalk -c pr1v4t3 -v2c 10.4.77.18 --> new flag

ARP spoofing

bettercap

net.probe on

net.show --> we get the vitcim's IP

bettercap -caplet web-override.cap -eval "set arp.spoof.targets 192.168.1.64

---------------------------------------------------------------------------------------------------------------------------

**WINDOWS**

Windows

cd

dir

dir -force (marche pas en france)

net user : list the users

net user /domain : try to connect to the domain

net user $username : get information on a particular user

type : équivalent de cat

Boot

Session initialization -> Winlogon Initialization

Before the login, all the process are run as root. We can try to modify

Hashes

LM Hash : chaîne de 14 caractères en 2 chaînes de 7 caractères

NT Hash (NTLM Hash) : MD4(UTF-16-LE(password))

NTLMv1 Hash : utilisé pour de l'authentification réseau

NTLMv2 Hash : même que v1 en plus long

--> récupérer les formats john sur les slides

Hashes storage

Local : bases de registres sam et system

Pour extraire les hashes, on utilise mimikatz : lsadump::sam /system:C:\system /sam:C:\sam

Then use john the ripper to crack the hashes : --format=nt

Sylabus