# Simulacre Examen UF3 Seguretat

**1.-**Suposant que tens implementat en la teva xarxa local un atac de tipus "ARP-Spoofing" (o bé "DHCP-Starvation", també valdria) digues què faries per a què la màquina "víctima" deixés de tenir Internet.

## MITMPROXY

**2.-a)** Utilitza el paràmetre *-M* de *mitmdump* per aconseguir, després d'implementar l'atac "ARP-Spoofing", que quan una determinada màquina víctima atrapada per l'atac vulgui fer la recerca de la paraula "macarrons" en el buscador https://duckduckgo.com, de forma automàtica vagi a parar sempre a la web de Microsoft (https://www.microsoft.com)

> PISTA: Cal que defineixis una expressió regular compatible amb la URL "*duckduckgo.com/?q=macarrons&XXXX*" (on "XXXX" pot ser qualsevol cosa)

**b)** Utilitza el paràmetre *--map-local* de *mitmdump* per aconseguir, després d'implementar l'atac "ARP-Spoofing", que quan una determinada màquina víctima atrapada per l'atac vulgui visitar qualsevol pàgina web, totes les fotos de tipus JPG (o millor dit, tots els recursos la URL dels quals acabi en ".jpg") d'aquesta pàgina siguin substituïts per l'arxiu "/usr/share/pixmaps/faces/penguin.jpg" local (pertanyent al paquet "gnome-control-center").

**c)** Utilitza el paràmetre -B de *mitmdump* per aconseguir, després d'implementar l'atac "ARP-Spoofing", que quan una determinada màquina víctima atrapada per l'atac visiti qualsevol pàgina web, s'injecti el següent codi CSS al final de la secció *<head>...</head>* del codi font d'aquesta pàgina visitada: *<style> div {color:limegreen;}</style>*

> NOTA: El codi CSS injectat en aquest apartat és inofensiu, però n'hi ha d'altres que poden ser perillosos. Un llistat dels més famosos es troba aquí: https://github.com/jbtronics/CrookedStyleSheets

**d)** Implementa el següent script amb *mitmdump* i observa què passa quan intentes navegar per Internet des de la màquina víctima. Raona el per què. ¿Es podria haver obtingut la mateixa funcionalitat només utilitzant els paràmetres emprats als apartats anteriors?

```
from mitmproxy import http
foto=open("logo.png","rb").read()
def response(flux):
  if "image/" in flux.response.headers["content-type"]:
     flux.response.content=foto
  if flux.response.content.find(b"<a") != -1:
     flux.response.content=flux.response.content.replace(b"<a",b"<h3>HOLA</h3><a")
```

> NOTA: L'arxiu "logo.png" pot ser qualsevol imatge que estigui guardada a la mateixa carpeta que l'script

**e)** Implementa un script que bloqueji totes les respostes provinents d'Internet que tinguin un "Content-Type" amb el valor "text/css" ¿Què veus al navegador quan visites llavors qualsevol pàgina web?

**f)** Utilitza el programa interactiu Mitmproxy (i empra els filtres adients si cal) per localitzar la petició concreta que envia en nom d'usuari i la contrasenya quan s'omple el formulari d'inici de sessió en el lloc web https://www.twitch.tv

> NOTA: No cal que t'hagis de registrar per poder realitzar l'exercici, pots indicar un nom d'usuari i contrasenya inventats

**fII)** Fes el mateix que a l'apartat anterior però ara pel registre del lloc https://elpais.com

**METASPLOIT**

**3.-a)** Genera un "payload" Meterpreter basat en Python de tipus "TCP reverse" per tal de, un cop implantat d'alguna manera -això no importa- en la màquina víctima, puguis connectar-t'hi amb ell via la teva consola Metasploit

**b)** Utilitza Meterpreter per "robar" els fitxers on es troben emmagatzemades les credencials que el navegador Firefox de la màquina víctima té guardades dels llocs webs que s'han anat visitant.

**c)** Desxifra les credencials anteriors per obtenir les contrasenyes i els usuaris d'aquests llocs webs.

Una manera relativament senzilla d'implantar un payload Metasploit en una màquina víctima (i, en general, qualsevol tipus de malware) és mitjançant l´us de llàpisos USB que l'incorporen, els quals infecten el sistema on s'introdueixen. Una sofisticació d'aquest atac és fer servir els anomenats "BadUSBs", llàpisos USB que un cop intruduïts i reconeguts pel sistema repliquen de forma automàtica la pulsació de determinades tecles per automatitzar així accions al sistema víctima (aquestes tecles normalment obriran un terminal i hi escriuran algunes comandes, etc). Al següent exercici es mostra un exemple d'aquesta variant, implementada sobre una placa Arduino que emula el comportament d'un "BadUSB".

**4.-**Explica pas a pas què fa l'script Arduino descrit a https://www.vesiluoma.com/exploiting-with-badusb-meterpreter-digispark i quin paper hi juga la comanda *base64* en tot plegat

> NOTA: Altres exemples interessants similars, però tenint com a sistema víctima un Windows són https://jaymonsecurity.com/ataque-badusb-o-rubber-ducky/ o https://systemweakness.com/pwn3d-in-seconds-attack-of-the-rubber-duck-26f9ae1d08b0

**5.-a)** Digues quin mòdul Metasploit, d'entre els que apareixen llistats a https://www.offsec.com/metasploit-unleashed/scanner-http-auxiliary-modules, realitza una tasca equivalent a la de cadascun dels següents programes individuals:

* *Wuff* o *GoBuster*
* *Hydra* o *Medusa* o *Ncrack*
* *Nmap* (amb el paràmetre *-sV*)
* *WPScan*

**aII)** Prova cadascun dels mòduls que has trobat a l'apartat anterior contra la pàgina web del centre (https://elpuig.xeill.net) i observa quina informació obtens.

**b)** Pensa i digues com resoldries el problema de tenir la consola de Metasploit executant-se en un ordinador amb IP privada de cara a poder-se connectar a "payloads" implantats no en màquines de la mateixa LAN sinó localitzades a Internet (hi ha vàries solucions possibles).

**MALWARE**

**6.-**A partir d'un fitxer binari inicialment creat amb la comanda *sudo dd if=/dev/zero of=arxiu.exe bs=1 count=50* , i fent servir algun editor hexadecimal (per consola tenim per exemple *hexedit*, i per entorn gràfic tenim Ghex), edita el fitxer binari anterior per tal de què doni positiu si li apliquem un arxiu de regles ClamAV com el següent (tant se val si el positiu és donat per la primera o per la segona regla):

*PepiGomez1;Target:0;**0&1**;0,20:70657065::i;20,50:70657061::i*
*PepiGomez2;Target:0;((**0|1)&(0|1))&((2|3)&(2|3))**;70657065::i;70657061::i;6a4068;8d4db0*

**7.-a)** Descarrega el "rootkit" Diamorphine a la teva carpeta personal d'un sistema virtual qualsevol i instal·la'l. Tot seguit executa'l per obtenir un terminal com a "root".

**b)** Al mateix sistema de l'apartat anterior, fes un escaneig amb *clam**d**scan* (no *clamscan*!) de tot el contingut de la teva carpeta personal per comprovar si ClamAV detecta (o no) la presència del "rootkit" Diamorphine (no cal que estigui funcionant en el moment de l'escaneig).

**8.-** Instal·la el keylogger "logkeys" vist a classe amb el mapa de tecla espanyol i posa'l en marxa per tal de capturar en el seu fitxer "log" totes les tecles pulsades a partir de llavors al sistema.

**PENTESTING**

**9.-**Registra't a https://tryhackme.com gratuïtament. Un cop hagis iniciat sessió, vés a "Learn"->"Search" i busca una màquina que s'anomeni "Basic Pentesting". Segueix tot el procés per finalment obtenir la "flag" que apareix escrita dins del fitxer "pass.bak" ubicat a la carpeta personal de l'usuari "k++" (els "+" oculten el caràcter real).

**10.-**Vés a https://www.vulnhub.com i tria una màquina qualsevol de les allà disponibles que tingui algun "Walkthrough" fet (és a dir, que ofereixi públicament l'explicació de com algú ha aconseguit realitzar el repte proposat pas a pas...les màquines més noves encara no en tindran cap: busca de més antigues). Segueix el/s "walkthrough"/s que necessitis per aconseguir tu també el repte.

**11.-**Executa la següent comanda per tal de posar en marxa l'aplicació web vulnerable WebGoat...:

*podman run -it -p 127.0.0.1:8080:8080 -p 127.0.0.1:9090:9090 -e TZ=Europe/Madrid webgoat/webgoat*

...i seguidament obre un navegador i vés a http://localhost:8080/WebGoat Després de crear un usuari (amb el nom i contrasenya que vulguis), accedeix a dins del lloc web. Un cop allà, utilitza algun tipus de proxy web (com ara Mitmproxy, ZAProxy, Burp... o fins i tot les eines de desenvolupador de Firefox (F12)) per tal d'aconseguir les següents fites:

**a)** http://localhost:8080/WebGoat/start.mvc#lesson/HttpBasics.lesson/2 i
http://localhost:8080/WebGoat/start.mvc#lesson/HttpProxies.lesson/4

**b)** http://localhost:8080/WebGoat/start.mvc#lesson/SqlInjection.lesson/8 i
http://localhost:8080/WebGoat/start.mvc#lesson/SqlInjectionAdvanced.lesson/4

**c)** http://localhost:8080/WebGoat/start.mvc#lesson/BypassRestrictions.lesson/1

**d)** http://localhost:8080/WebGoat/start.mvc#lesson/HtmlTampering.lesson/1