TALLER 4
EXPLOITS DE SISTEMA

Curs: PQTM 19. Tècnic/a en Ciberseguretat

Ivan Ricart Borges iricartb@uoc.edu

ÍNDEX

1. Riscos que implica un atac de desbordament de buffer.....3
2. Riscos que implica un atac d'injecció SQL.....4
3. Possibles atacs web sobre el procés d'autentificació.....6
4. Descripció de l'eina Nmap.....8
5. Vulnerabilitats CVE.....11
6. Descripció de l'eina Nessus.....13
6. Fonts d'informació.....14

1. Quins són els riscos que implica un "overflow", segons allò que heu vist a les transparències del webinar?

Els llenguatges de programació proporcionen mecanismes per gestionar la memòria, principalment mitjançant la definició de variables, aquestes poden ser definides per reservar una certa quantitat de memòria i per tant emmagatzemar una certa quantitat d'informació. Un *overflow* es produeix quan la informació a emmagatzemar supera els límits de la reserva i per tant es produeix una pèrdua d'informació que pot afectar al comportament de l'aplicació.

Un *buffer overflow* es produeix quan l'aplicació sobreescriu les zones de memòria no assignades i adjacents al buffer, superant la seva limitació degut a la falta de mecanismes de control i que per tant, podrà provocar un comportament de l'aplicació anòmal e inesperat. Els desbordaments de buffer sovint es desencadenen de forma intencionada, amb la entrada d'informació mal formada, on la quantitat d'informació a emmagatzemar és superior a la mida de la reserva del buffer, provocant així el seu desbordament. Els llenguatges de programació C i C++ estan associats habitualment a desbordaments de memòria, degut a que no proporcionen cap tipus de protecció integrada contra l'accés o la sobreescriptura de dades en cap part de la memòria, la comprovació de límits pot evitar desbordaments de buffer, però requereix de coneixement, codi addicional i temps de processament. A continuació podem veure un exemple d'atac de desbordament de buffer sobre la funció sprintf.

sprintf(char *buffer, const char *format, argv)

Imaginem que l'aplicació reserva una quantitat de memòria de 100 bytes, que fa ús de la funció sprintf per copiar la informació d'entrada per paràmetre de l'aplicació cap a aquesta zona i que l'usuari introdueix un valor d'entrada de 128 bytes, ex. 'A'x128.

BUFFER						MEMÒRIA ADJACENT			
0x00ACFA00	0x00ACFA01	0x00ACFA02	..	0x00ACFA62	0x00ACFA63	0x00ACFA64	0x00ACFA65	..	0x00ACFA7F
0	1	2	..	98	99	100	101	..	127
A	A	A	..	A	A	A	A	..	A

L'aplicació sobreescriurà 28 bytes de memòria adjacent al buffer, la qual cosa podrà provocar un comportament de l'aplicació anòmal, segons com, l'aplicació deixarà de respondre degut a que haurà sobreescrit informació necessària per retornar al punt anterior a la crida de la funció i per tant l'aplicació executarà la instrucció JMP 0x41414141, provocant una violació de segment.

Un atac de desbordament de buffer podrà provocar principalment els següents riscos:

RISC	DESCRIPCIÓ
Denegació de servei [D] Disponibilitat	L'aplicació podrà sofrir una denegació de servei, en el cas de que es sobreescriguin zones de memòria necessàries per establir un JMP, provocant així una violació de segment, per tant, podem dir que la disponibilitat del servei o aplicació es veurà clarament afectada.
Execució de codi remot [D] Disponibilitat [I] Integritat [C] Confidencialitat [A] Atomicitat	L'aplicació podrà executar codi especificat per l'atacant (payload), aquests tipus d'atacs són molt complexes, és necessita fer un estudi minuciós del comportament de l'aplicació mitjançant eines de depuració. L'atacant podria crear una shell remota fent ús de sockets amb els permisos de l'aplicació, comprometent així tota la seguretat del terminal. Una tècnica utilitzada per aconseguir realitzar aquests tipus d'atacs en Windows, és fer un estudi de les DLL que carrega l'aplicació, trobar la direcció de memòria on es podria trobar la instrucció JMP ESP i sobreescriure el buffer de tal manera que realitzi el salt apuntant cap a aquesta última, posteriorment l'aplicació executarà les instruccions apuntades pel registre ESP, que en aquest cas hauria d'haver el començament del payload del atacant.

Una possible solució que pot adoptar el usuari per mitigar aquests tipus d'atacs és fer ús de funcions segures que tenen en compte els límits dels buffers o bé implementar una funció genèrica pròpia amb mesures de control i utilitzar-la en tots els casos on es desitgi copiar informació d'una zona de memòria cap a un altra.

2. Quin són els riscos que implica un atac d'injecció SQL, segons allò que heu vist a les transparències del webinar?

La injecció SQL és una tècnica d'injecció de codi, utilitzada per atacar aplicacions basades en dades, en les quals s'insereixen instruccions SQL malintencionades en un camp d'entrada per explotar la existència d'una vulnerabilitat relacionada amb la mancança de control de filtres de caràcters d'escapament. Els atacs d'injecció de SQL permeten als atacants fer malbé la identitat, manipular les dades existents, provocar problemes de repudi com ara cancel·lar transaccions o canviar saldos, permetre la divulgació completa de totes les dades del sistema, destruir les dades o fer que no estiguin disponibles i convertir-se en administradors de la servidor de bases de dades. A continuació podem veure un exemple d'atac d'injecció SQL sobre una aplicació que no controla els caràcters d'escapament.

Aplicació vulnerable a injecció SQL
Imaginem que l'aplicació proporciona a l'usuari un formulari de validació mitjançant una serie de camps d'usuari i contrasenya, que la informació enviada acaba executant una sentencia SQL en el servidor i que l'usuari introdueix en el camp d'usuari la sentencia iricartb i de contrasenya password123!.
“SELECT * FROM users WHERE username='” + sUsername + “' AND password='” + sPassword + “';” SELECT * FROM users WHERE username='{username}' AND password='{password}'; SELECT * FROM users WHERE username='iricartb' AND password='password123!';
Un correcte funcionament de l'aplicació executaria la sentencia anterior per comprovar les credencials del usuari que s'intenta identificar, aquest serà identificat correctament en el cas de que la sentencia SQL retorni la fila afectada. A continuació l'usuari introdueix en el camp de contrasenya la sentencia ' OR username='administrator'; --.
SELECT * FROM users WHERE username='iricartb' AND password='' OR username='administrator'; --
La sentencia OR amb la informació proporcionada provoca que la sentencia SQL sigui correcta i que retorni la fila relacionada amb l'usuari administratiu, d'aquesta manera el atacant podrà identificar-se com administrador de la plataforma sense conèixer la seva contrasenya.

Un atac d'injecció SQL podrà provocar principalment els següents riscos:

RISC	DESCRIPCIÓ
Execució de codi remot [D] Disponibilitat [I] Integritat [C] Confidencialitat [A] Atomicitat	L'aplicació podrà executar codi SQL especificat per l'atacant, per tant, fer malbé la identitat, manipular les dades existents, provocar problemes de repudi, permetre la divulgació completa de totes les dades del sistema, destruir les dades o fer que no estiguin disponibles i convertir-se en administradors de la servidor de bases de dades. Resumint l'atacant podria fer-se amb el control del servei de base de dades i ens segons quins casos, una mica més complexes, combinant aquest amb un altre tipus de vulnerabilitat, fer-se amb el control del terminal, com per exemple executar comandes remotes mitjançant sentencies SQL amb xp_cmdshell.

Una possible solució que pot adoptar el usuari per mitigar aquests tipus d'atacs és fer ús de frameworks, llibreries o funcions que tenen en compte aquests tipus d'atacs o bé implementar una funció genèrica pròpia amb mesures de control de caràcters d'escapament i utilitzar-la en tots els casos on es permet a l'usuari introduir informació.

3. Suposeu que esteu dissenyant una pàgina web per validar un usuari amb el seu password. Expliqueu quins atacs es podrien dur a terme contra aquesta pàgina, quins riscos hi ha, i què faríeu per mitigar-los.

Hem de diferenciar els atacs que es poden dur a terme degut a la infraestructura adoptada, com per exemple la existència de vulnerabilitats del programari relacionat amb el servei web o bé els protocols utilitzats per intercanviar informació amb aquells atacs relacionats amb les vulnerabilitats provocades durant la fase del desenvolupament lògic de l'aplicació.

Com atacs que es poden dur a terme degut a la infraestructura podem citar els següents:

ATACS / RISCOS	DESCRIPCIÓ	POSSIBLE SOLUCIÓ
Explotació d'una vulnerabilitat relacionada amb la versió del programari del servei web	Els serveis web són programaris complexes on es gestionen les comunicacions i les peticions dels usuaris per mostrar la informació sol·licitada. Aquests com qualsevol altre tipus de programari poden contindre errors de codi i en alguns casos crear la existència de vulnerabilitats. Per curiositat el nom del servidor web Apache fa referència a “patched system”, degut a la gran quantitat de vulnerabilitats que s'han solucionat i que han transformat aquest servei com un dels més segurs en l'actualitat.	Per mitigar o reduir el risc d'explotació d'aquestes vulnerabilitats, es necessari consultar i actualitzar les versions del programari, tal i com ho especifiqui el fabricant.
Intercepció d'informació no xifrada mitjançant atacs MITM	El protocol de comunicació http (hyper text transfer protocol) va ser dissenyat per intercanviar informació sense cap tipus de mecanisme de xifratge, degut això, la informació viatja en forma plana i si algú la pogués interceptar la podria interpretar, vulnerant així la seva confidencialitat.	Configurar el servidor web per permetre la comunicació xifrada mitjançant el protocol https.

Com atacs que es poden dur a terme degut a la presencia de vulnerabilitats durant la fase de desenvolupament lògic de l'aplicació podem citar els següents:

ATACS / RISCOS	DESCRIPCIÓ	POSSIBLE SOLUCIÓ
Injecció SQL en els camps dels formularis web	Els camps dels formularis d'una pàgina web són propensos a sofrir atacs d'injecció SQL, això és així degut a que normalment la informació enviada interactua amb una base de dades i per la falta de conscienciació dels desenvolupadors sobre la gravetat o el riscos que és poden produir per no controlar la informació enviada. La efectivitat d'aquests depenen principalment del disseny de l'aplicació i en cas de efectuar-se poden provocar problemes de seguretat importants, fins i tot, perdre el control del sistema afectat.	Fer ús de frameworks, llibreries o funcions que tenen en compte aquests tipus d'atacs o bé implementar una funció genèrica pròpia amb mesures de control de caràcters d'escapament i utilitzar-la en tots els casos on es permet a l'usuari introduir informació.
Cross-Site Scripting en els missatges de resposta d'usuari o contrasenya incorrecta	Normalment quan s'introdueix incorrectament el usuari o la contrasenya en el formulari de validació d'una pàgina web, aquest ens ho notifica mitjançant un missatge d'error, depenen del disseny d'aquest missatge i si l'aplicació renderitza el valor passat en els camps del formulari, l'aplicació podria sofrir atacs XSS. Cal dir a més a més, que la gravetat de la vulnerabilitat dependrà en gran mesura de si l'aplicació fa ús de paràmetres GET per renderitzar aquesta informació.	Utilitzar mecanismes de control de caràcters d'escapament i evitar el ús de paràmetres GET. Aquest últim no garanteix la inexistència de la vulnerabilitat però si que minimitza el impacte, de tal manera que l'atacant no podrà enviar un simple enllaç a la víctima per a que executi el codi JavaScript.
Redireccions fraudulentes sobre els mecanismes d'autentificació OAuth	A mesura que avança la tecnologia també avancen els mètodes d'autentificació, el protocol OAuth permet comunicar-se de forma segura amb altres plataformes per realitzar el procès d'autentificació. El problema existeix quan hi han mancances d'implementació d'aquest, per exemple, no controlar correctament el valor del paràmetre redirect_uri, d'aquesta manera un atacant podria enviar el enllaç a la víctima per a que realitzes el procés d'autentificació i que posteriorment l'aplicació el redirigís cap a una web fraudulenta controlada per l'atacant.	Controlar correctament el valor del paràmetre redirect_uri, sense permetre la introducció de dominis del estil: google.com.attacker.com com podem observar anteriorment, aquest domini podria estar comprat per l'atacant ja que google.com és un subdomini del domini attacker.com.

4. Instal·leu-vos nmap a la vostra màquina host. A la web nmap.org hi ha instal·ladors per als sistemes operatius Windows, Linux i Mac. Cerca informació sobre quines exploracions bàsiques pots fer amb nmap i fes-ne un resum. A continuació, executa una d'aquestes exploracions a la xarxa d'àrea local on estiguis connectat (se suposa que la de casa, mira de no fer-ho a la feina ;-) i una altra exploració contra la màquina virtual Ubuntu (o similar) que tinguis creada. Explica una mica els resultats obtinguts.

Nmap és un eina *open source* que permet realitzar exploracions de xarxa i auditories de seguretat, es va dissenyar per analitzar grans xarxes però també funciona molt bé contra equips individuals. *Nmap* s'utilitza per determinar quins equips es troben disponibles en una xarxa, quins serveis (noms i versions de les aplicacions) ofereixen, quins sistemes operatius executen, quins tipus de filtratge de paquets o tallafocs s'utilitzen, etc. Generalment s'utilitza en auditories de seguretat, però molts administradors de xarxes i sistemes el troben útil per realitzar tasques habituals, com per exemple, el inventariat de la xarxa, la planificació d'actualització i la monitorització del temps en que els equips o serveis es mantenen actius.

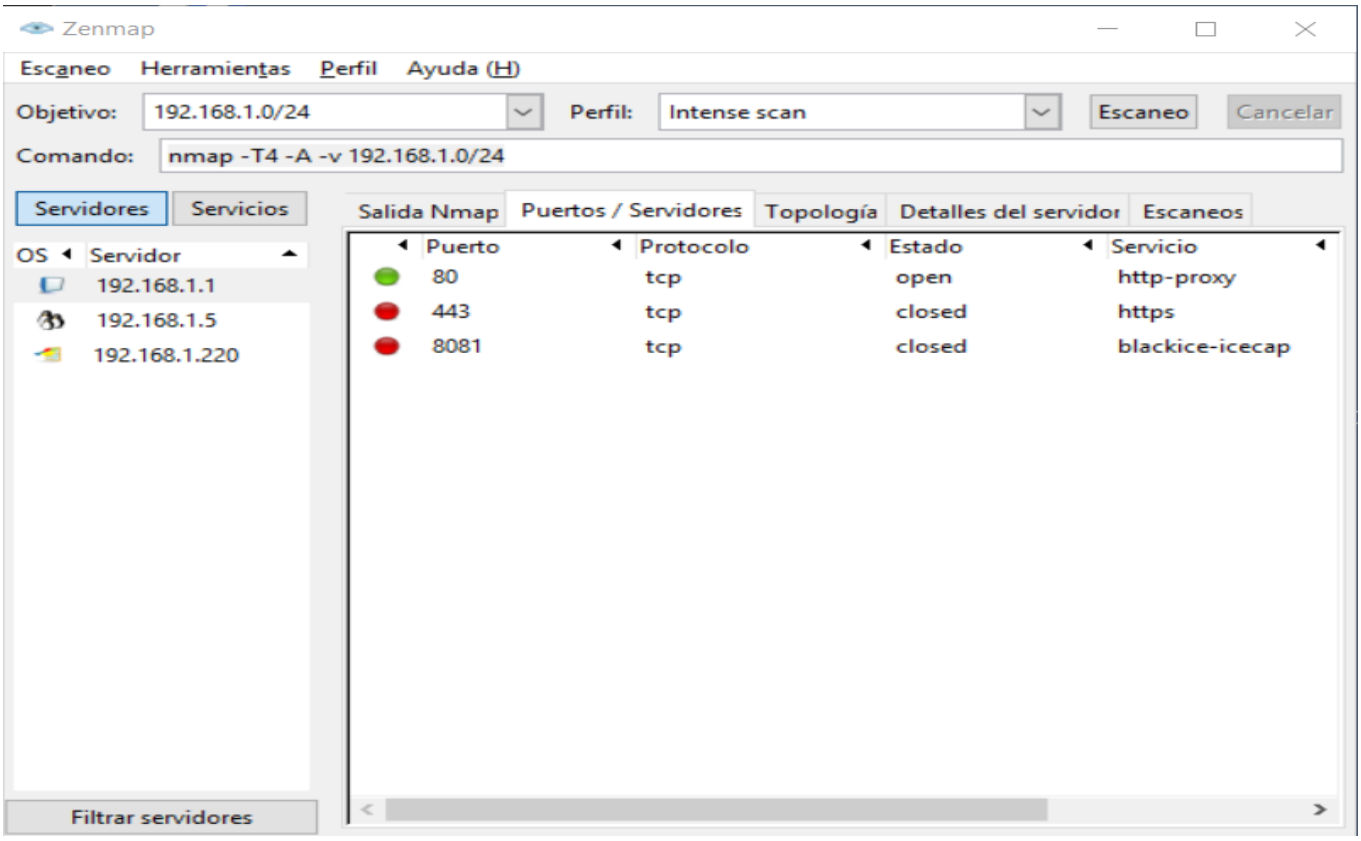
A continuació citem algunes de les exploracions que es poden realitzar amb l'eina *nmap*.

EXPLORACIÓ	COMANDA	DESCRIPCIÓ
Intensiva	nmap -T4 -A -v	Aquest és un dels anàlisi típics, s'utilitza el paràmetre -T4 per accelerar el procès, la opció -A per intentar detectar el sistema operatiu i les versions, i la opció -v (verbose) per mostrar informació més detallada.
Intensiva + UDP	nmap -sS -sU -T4 -A -v	S'utilitza el paràmetre -sS per fer un sondeig TCP SYN, aquesta és la opció per defecte i el més popular, es capaç de sondejar milers de ports per segon, és poc molest perquè no s'arriben a completar les connexions TCP, la opció -sU permet realitzar un sondeig UDP i es totalment compatible amb l'anterior, les altres opcions ja han sigut comentades anteriorment.
Intensiva + ports	nmap -p 1-65535 -T4 -A -v	S'utilitza el paràmetre -p per especificar els ports a escanejar, en aquest cas tots els 65535 ports, les altres opcions ja han sigut comentades anteriorment.

Intensiva + no ping	nmap -T4 -A -v -Pn	S'utilitza la opció -Pn per analitzar totes les màquines especificades, independentment de que si la maquina respon o no a les peticions ICMP, les altres opcions ja han sigut comentades anteriorment.
Ràpida	nmap -T4 -F	S'utilitza el paràmetre -F per indicar que només es desitja sondejar el ports llistats en el fitxer nmap-services, les altres opcions ja han sigut comentades anteriorment.
Ràpida ++	nmap -sV -T4 -O -F --version-light	S'utilitza el paràmetre -sV per indicar la detecció de versions, la opció -O per detectar el sistema operatiu , la opció –version-light permet que la detecció de versions encara sigui més ràpida, les altres opcions ja han sigut comentades anteriorment.

Posteriorment he realitzat la exploració intensiva usant l'eina *nmap* dintre de la xarxa local i he obtingut els resultats següents:

nmap -T4 -A -v 192.168.1.0/24



Com podem observar el escaneig ha trobat tres màquines dintre de la xarxa local, el router amb direcció IPv4 192.168.1.1, la màquina virtual LUbuntu amb direcció IPv4 192.168.1.5 i finalment la màquina local amb sistema operatiu Windows amb direcció IPv4 192.168.1.220. El escaneig intensiu utilitzant el paràmetre -A ha fet que l'eina *nmap* detectés els sistemes operatius de cada una de les màquines, a més a més dels serveis proporcionats. El router i la màquina LUbuntu presenten un servei web usant el protocol http en el port 80, en canvi, la màquina local Windows, que fins i tot ha detectat la seva versió, es a dir Microsoft Windows 10 1703, presenta els serveis msrpc en el port 135 i netbios-ssn en el port 139.

Posteriorment he realitzat la exploració intensiva usant l'eina *nmap* dirigida cap a la màquina virtual LUbuntu.

```
nmap -T4 -A -v 192.168.1.5
```

De forma anàloga, l'eina *nmap*, ha detectat els serveis proporcionats i el sistema operatiu de la màquina virtual, ha detectat que presenta públicament un servei web en el port 80 mitjançant un servidor Apache amb versió 2.4.29 i el sistema operatiu LUbuntu 19.04.

5. Connecteu-vos a https://cve.mitre.org i descriviu de què es tracta. Trieu alguna de les vulnerabilitats i feu-ne un petit resum.

El sistema C*ommon Vulnerabilities and Exposures (CVE)* proporciona un mètode de referència per les públicament conegudes *information-security vulnerabilities and exposures*. La *National Cybersecurity FFRDC*, dirigida per *Mitre Corporation*, manté el sistema.

La documentació de la *Mitre Corporation* defineix els identificadors CVE com a únics, identificadors comuns per les públicament conegudes *information-security vulnerabilities* dels paquets de software, aquests són assignats per la *CVE Numbering Authority (CNA)*.

La pàgina web https://cve.mitre.org mostra informació de seguretat oficial de les vulnerabilitats conegudes sobre els paquets de software e identificades amb el identificador CVE, també ens permet cercar la informació mitjançant tokens o paraules claus, cal dir que el sistema emmagatzema una gran quantitat d'informació i que per tant fins i tot, podem trobar identificades les vulnerabilitats més antigues dels paquets de software.

A continuació, degut a la seva gravetat, he cercat i seleccionat una vulnerabilitat del any 2001, concretament la vulnerabilitat amb número de identificació CVE-2001-0333.

CVE-ID

CVE-2001-0333 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Directory traversal vulnerability in IIS 5.0 and earlier allows remote attackers to execute arbitrary commands by encoding .. (dot dot) and "\" characters twice.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- BID:2708
- URL:http://www.securityfocus.com/bid/2708
- BUGTRAQ:20010515 NSFOCUS SA2001-02 : Microsoft IIS CGI Filename Decode Error Vulnerability
- URL:http://marc.info/?l=bugtraq&m=98992056521300&w=2
- CERT:CA-2001-12
- URL:http://www.cert.org/advisories/CA-2001-12.html
- MS:MS01-026
- URL:https://docs.microsoft.com/en-us/security-updates/securitybulletins/2001/ms01-026
- OVAL:oval:org.mitre.oval:def:1018
- URL:https://oval.cisecurity.org/repository/search/definition/oval%3Aorg.mitre.oval%3Adef%3A1018
- OVAL:oval:org.mitre.oval:def:1051
- URL:https://oval.cisecurity.org/repository/search/definition/oval%3Aorg.mitre.oval%3Adef%3A1051
- OVAL:oval:org.mitre.oval:def:37
- URL:https://oval.cisecurity.org/repository/search/definition/oval%3Aorg.mitre.oval%3Adef%3A37
- OVAL:oval:org.mitre.oval:def:78
- URL:https://oval.cisecurity.org/repository/search/definition/oval%3Aorg.mitre.oval%3Adef%3A78
- XF:iis-url-decoding(6534)
- URL:https://exchange.xforce.ibmcloud.com/vulnerabilities/6534

Com podem observar en la imatge anterior, la informació s'estructura de tal manera, on primerament s'identifica la vulnerabilitat amb el número CVE corresponent, posteriorment es realitza una breu descripció de la vulnerabilitat i finalment es citen referencies d'interès per poder profunditzar en el tema.

En aquest cas he seleccionat la vulnerabilitat amb número de identificació CVE-2001-0333, la qual comprometia totalment la seguretat dels servidors que utilitzaven el servei web IIS de Microsoft per allotjar pàgines web. La vulnerabilitat consistia principalment en realitzar peticions doblement codificades d'algun dels caràcters de la sentencia "../", d'aquesta manera es saltaven els filtres de seguretat del servei web IIS de tal manera que no se'n adonava de que el usuari intentava accedir a carpetes del nivell superior del projecte web i posteriorment aquest descodificava la sentencia assumint que la petició era segura. Tenint aquesta informació present i que tots els servidors tenien instal·lat el sistema operatiu Windows NT o 2000, els atacants no van tardar massa en realitzar peticions més complexes per intentar que el servei web IIS executes l'aplicació cmd.exe mitjançant comandes i que retornés la informació de resposta mitjançant el protocol http, de forma resumida podríem dir, que els atacants tenien una shell remota sense gaire esforç de tots els servidors web amb sistema operatiu Windows.

A continuació es mostra algunes de les possibles peticions que podien fer els atacants per explotar aquesta vulnerabilitat.

```
http://[servidor]/msdac/..%255c../winnt/system32/cmd.exe?c+dir+c:\
http://[servidor]/scripts/..%25%35%63..%25%35%63../winnt/system32/cmd.exe?c+dir+c:\
http://[servidor]/_vti_bin/..%%35c..%%35c..%%35c..%%35c../winnt/system32/cmd.exe?c+dir+c:\
```

L'atacant podia utilitzar aquesta codificació tantes vegades com volgués dintre de la mateixa petició URL, com més cops la utilitzes i per tant, més llarga fos la sentencia, més probabilitats tenia d'arribar a l'arrel on hi havia la carpeta d'instal·lació del sistema operatiu Windows. Si descodifiquem dues vegades les sentencies hexadecimals anteriors veurem la típica sentencia **../**. Finalment podem observar com executava el fitxer cmd.exe amb l'ús del símbol **?** per passar el paràmetre **c** i així poder executar la comanda. El símbol **?** s'utilitza per passar paràmetres mitjançant el protocol http i el paràmetre **c** de la comanda cmd s'utilitza per enviar comandes.

Explotant la vulnerabilitat utilitzant alguna de les sentencies anteriors, l'atacant podia veure la resposta de la comanda en el navegador web, en aquest cas, el llistat d'arxius de la unitat c.

6. Busqueu informació sobre Nessus, què és i per a què serveix. Feu-ne també un petit resum.

Nessus és un escàner remot de vulnerabilitats que permet identificar tots els serveis proporcionats per una màquina remota i determinar si aquests estan protegits sobre tots els exploits coneguts. Segons la pàgina oficial de Nessus, es a dir, http://www.nessus.org, aquest és l'escàner de vulnerabilitats més popular del món que és utilitzat en més de 75.000 organitzacions d'arreu del món.

El projecte *"Nessus"* va ser començat per *Renaud Deraison* l'any 1988, posteriorment, concretament l'any 2002, *Renaud* funda juntament amb *Ron Gula* (creador de *Dragon Intrusion Detection System*) i *Jack Huffard* la organització *Tenable Network Security.*

El sistema d'escaneig de vulnerabilitats de *Nessus* està compost d'un servidor i d'un client, aquests dos poden residir en màquines separades. El programa servidor s'anomena *nessusd* (dimoni amb la funcionalitat "d'atacar" les altres màquines de la xarxa) i la seva instal·lació normalment es localitza en el path /opt/nessus/sbin/nessusd, per una altra banda, el client s'anomena *nessus* i la seva instal·lació normalment es localitza en el path /opt/nessus/bin/nessus, aquest últim, defineix el comportament del servidor, de tal manera que li indica com ha de realitzar els atacs i a quina ubicació ha d'emmagatzemar la informació de seguretat obtinguda. El client pot emmagatzemar diferents tipus d'escenaris d'atac usant diferents nomenclatures i utilitzar-los per comprovar el nivell de seguretat que s'estableix en les diverses màquines. Els tests de seguretat del sistema *Nessus* estan escrits en un llenguatge especial anomenat *Network Attack Scripting Language (NASL),* cada un d'ells consisteix d'un *plugin* extern i actualment n'existeixen uns 70.000, la actualització es pot dur a terme automàticament, mitjançant la comanda nessus-update-plugin. També cal dir, que *Nessus* és capaç de detectar els serveis, encara que aquests estiguin executant-se en ports diferents segons l'estàndard i aplicar així, les proves corresponents.

6. Fonts d'informació.

Recursos

Explotació de vulnerabilitats.

PID_00217345.pdf

Buffer overflow

Explicació sobre en que consisteix la tècnica de desbordament de buffers.

https://en.wikipedia.org/wiki/Buffer_overflow

Web hacking 101 - Vulnerabilitats web

Explicació i exemples d'atacs BoF, SQLi, XSS, CSRF, IDOR, RCE, etc.

http://index-of.es/Miscellanous/LIVRES/web-hacking-101.pdf

Nmap

Pàgina oficial - Informació genèrica de l'eina.

http://www.nmap.org

Manual d'utilització de l'eina Nmap.

https://we.riseup.net/assets/77169/Manual-de-uso-de-Nmap.pdf

CVE

Pàgina oficial - Informació i llistat de vulnerabilitat identificades.

https://cve.mitre.org

Definició de Common Vulnerabilities and Exposures (CVS).

https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures

Informació sobre la vulnerabilitat CVS-2001-0333.

https://cve.mitre.org

http://www.securityfocus.com/bid/2708

https://packetstormsecurity.com/files/cve/CVE-2001-0333

Nessus

Pàgina oficial - Informació genèrica de l'eina.

http://www.nessus.org

Informació detallada de l'eina Nessus.

https://es.scribd.com/document/386071484/Nessus-pdf