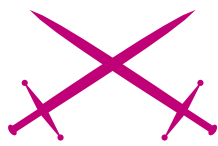


Analisi remota di un host



Attacco

Pierpaolo Palazzoli, Matteo Valenza, Luca Leone, Nicola Mondinelli, Michele Meneghelo

Grado di difficoltà



L'analisi remota permette di conoscere, via rete, porte e servizi aperti di un host. Utilizzata insieme a tecniche di analisi delle vulnerabilità sui servizi aperti ci permette di individuare i potenziali punti di accesso ad un pc/server su internet o su una lan.

Per effettuare una buona analisi, tuttavia, è bene non sottovalutare una serie di fattori.

Innanzitutto è opportuno conoscere gli eventuali meccanismi di protezione, hardware e software, introdotti dagli amministratori di rete tra il nostro host e il target dell'analisi.

E' opportuno raccogliere il maggior numero di informazioni possibili sull'host da analizzare, in modo da poter scegliere gli strumenti più adatti a testarne le peculiarità.

Un'altro utile accorgimento prevede la possibilità di utilizzare più prospettive, sfruttando, quindi, più connessioni da diversi provider.

In questo senso è altrettanto importante utilizzare diverse tipologie di piattaforme dalle quali poter eseguire gli strumenti di analisi. In caso si abbia a disposizione una sola macchina, si potrà ricorrere ad un sistema di virtualizzazione.

Una volta pronti gli strumenti, si può procedere all'analisi vera e propria, che può essere schematizzata in tre fasi:

- scansione delle vulnerabilità,
- assessment della dicurezza della rete,
- attacco simulato.

Sottolineiamo un ultimo aspetto, spesso trascurato dalle trattazioni che si focalizzano prevalentemente sugli strumenti dimenticando il fattore che caratterizza maggiormente l'analisi: le decisioni dell'operatore.

E' bene non dimenticare che l'analisi non è fatta dagli strumenti, ma dalle decisioni e dalle scelte dell'operatore che li utilizza.

Tool e strumenti

Gli strumenti che abbiamo oggi a disposizione per effettuare analisi remota sono numerosi, classificabili in diverse tipologie. In questo

Dall'articolo imparerai...

- L'utilizzo di strumenti per l'analisi delle vulnerabilità,
- La lettura dei report di strumenti di assessment,
- Le tecniche di riconoscimento dei servizi.

Cosa dovresti sapere...

- Basi di Neworking,
- Basi della struttura del TCP/IP,
- Basi di Network auditing.

articolo cercheremo di individuare quelli più adatti alle nostre applicazioni.

Spesso questi strumenti non sono legati ad uno specifico sistema operativo.

Una prima classificazione potrebbe essere la seguente:

- **Strumenti di base:** utilizzati per una diagnostica di base, spesso sono comandi standard del sistema operativo (ping, whois, host, dig, traceroute, telnet ...),
- **Strumenti di port scanning e network mapping:** sono gli strumenti che ci permettono di individuare servizi TCP/UDP in ascolto su un host remoto (nmap, amap, hping2, xprobe2, Angry IP scanner),
- **Strumenti di analisi di vulnerabilità:** sono gli strumenti più evoluti, in grado di riconoscere eventuali vulnerabilità sul servizio esaminato (Nessus, SARA, Foundstone SuperScan, ISS scanner),
- **Strumenti di analisi del servizio:** sono strumenti più specifici, mirati ad un preciso servizio, possono essere utilizzati per un'analisi più approfondita (Nikto, N-stealth, gobblessh, epdump, rpcscan, WenInspect).

In determinate situazioni risulta importante non solo scegliere il tool più idoneo, ma anche *camuffare* i tentativi di *accesso/analisi* effettuati.

Strumenti che possiamo utilizzare a questo scopo sono: proxy anonymizer, fragrouter, fragtest, fragroute, Ping tunnel.

I tool menzionati sono solo una piccola parte di un mondo in continuo fermento: tenersi aggiornati, anche con ricerche al momento stesso dell'analisi aiuta ad utilizzare gli strumenti giusti al momento giusto.

Approccio e Metodologia

Un approccio metodologico all'analisi è fondamentale: è necessario pianificare e avere chiaro i punti di partenza e di arrivo della propria attività.

La nostra finalità è quella di reperire tutte le potenziali vulnerabilità di un host. Avendo ben chiaro questo obiettivo è necessario seguire una metodologia che ci permetta di individuare in modo completo tutte le informazioni relative all'host in questione.

Se l'obiettivo è un server con nome è possibile controllarne la proprietà tramite il comando *whois*. Nota la proprietà del server si ha a disposizione un contatto mail amministrativo e un contatto mail tecnico.

Con una risoluzione del dominio in tutte le sue tipologie (A, MX, NS..) possiamo conoscere tutti gli host in relazione a questo dominio.

Ottenuti gli indirizzi IP potremo iniziare ad indagare su:

- gestore del DNS autoritativo,
- gestore del server WEB,
- il gestore del server di posta.

Tutte queste informazioni sono ottenibili con il comando *whois*:

```
whois hakin9.org
whois 62.111.243.84
```

Dalle informazioni ottenute dal *whois* dell'indirizzo IP si può risalire con certezza al provider gestore della linea o addirittura del server condiviso o dedicato dove è configurato il servizio web.

Tutte queste informazioni sono fondamentali se desideriamo intraprendere delle azioni di social engineering verso il provider ospitante.

I sistemi di sicurezza di rete organizzano i livelli di accesso a seconda delle reti di provenienza, se riuscissimo ad ottenere una connessione RTG/ISDN dal provider gestore del server potremmo sottostare a livelli di controllo di accesso meno restrittivi.

Ottenute le informazioni riguardanti l'indirizzo IP si può iniziare ad usare tools per reperire tutte le informazioni possibili sull'host remoto.

Le prime informazioni da acquisire sono quelle relative alla topologia

Listato 1: Host recovery - nmap -sP 192.168.3.* oppure nmap -sP 192.168.3.0/24

```
Starting Nmap 4.01 ( http://www.insecure.org/nmap/ )
Host 192.168.3.221 appears to be up.
Host 192.168.3.254 appears to be up.
MAC Address: 00:11:95:XX:XX:XX (D-Link)
Nmap finished: 256 IP addresses (2 hosts up) scanned in 6.173 seconds
```

Listato 2: Scan attivo delle porte TCP - nmap -sS 172.21.3.1

```
Starting Nmap 4.01 ( http://www.insecure.org/nmap/ )
Interesting ports on serverweb(172.21.3.1):
(The 1667 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
3306/tcp   open  mysql
MAC Address: 00:14:5E:XX:XX:XX (IBM)
Nmap finished: 1 IP address (1 host up) scanned in 0.643 seconds
```

Listato 3: Scan attivo delle porte UDP - nmap -sU servervoip -p 4569,5060

```
Starting Nmap 4.01 ( http://www.insecure.org/nmap/ )
Interesting ports on servervoip:
PORT      STATE SERVICE
4569/udp  open|filtered unknown
5060/udp  open|filtered sip
Nmap finished: 1 IP address (1 host up) scanned in 4.128 seconds
```



della rete e l'enumerazione dei servizi dell'host remoto.

Per poter *disegnare* una mappa topologica possiamo utilizzare un programma in grado di elaborare le informazioni di ritorno di un traceroute. Un ottimo programma per questo scopo è cheops-ng. Fig 1

Cheops funziona in base ad un'architettura client server: dopo aver installato il programma è necessario lanciare prima l'agent (cheops-agent) e successivamente il client (cheops-ng). Questo tipo di architet-

tura permette di usufruire di più punti di vista di uno stesso host o rete.

L'analisi effettuata da cheops mette in correlazione i dati provenienti dai passaggi di traceroute con quelli ottenuti dall'enumerazione fornita da nmap.

L'utilizzo base di cheops-ng è abbastanza intuitivo: una volta mandati in esecuzione sia l'agent che il client verrà richiesto l'IP dell'agent (127.0.0.1, se eseguiti sul medesimo host). Selezionando dal menu View-space, Add Network si potrà aggiun-

gere il range di IP che desideriamo tracciare.

Il programma tenterà di disegnare la rete nella sua struttura, calcolando i passaggi e tentando di riconoscere il sistema operativo dei singoli nodi (dev'essere specificato nelle opzioni). Per migliorare la risposta di cheops potremmo utilizzare un traceroute con opzione -I per l'incapsulamento nel protocollo icmp (per evitare qualche blocco messo su *router/firewall* nella rete remota).

Questa prima parte di analisi, chiamata anche ricognizione, è finalizzata esclusivamente alla collezione di informazioni non strettamente tecniche, allo scopo di documentare il contesto e le proprietà dell'host remoto.

Listato 4: Controllo del Sistema operativo e dei servizi in ascolto (fingerprint attivo) - nmap -O -sV -T4 serverweb oppure nmap -A -T4 serverweb

```
Starting Nmap 4.01 ( http://www.insecure.org/nmap/ )
Interesting ports on serverweb:
(The 1667 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.3p2 (protocol 2.0)
25/tcp    open  smtp     Postfix smtpd
80/tcp    open  http     Apache httpd
443/tcp   open  ssl/http Apache httpd
Device type: general purpose
Running: Linux 2.4.X|2.6.X
OS details: Linux 2.4.6 - 2.4.26 or 2.6.9, Linux 2.6.5 - 2.6.11
Service Info: Host: severweb.domanin.com
```

Listato 5: Scansione di RPCscan - Source Destination Summary

```
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 SYN SEQ=2397105131 LEN=0
WIN=5840
[192.168.0.7] [192.168.0.8] TCP: D=51008 S=2049 SYN ACK=2397105132
SEQ=2432014017 LEN=0 WIN=65535
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014018 WIN<<2=5840
[192.168.0.8] [192.168.0.7] RPC: C XID=59188766 PROG=Port mapper
VERS=434311 PROC=0 (Do nothing)
[192.168.0.7] [192.168.0.8] RPC: R XID=59188766 - Program unavailable
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014046 WIN<<2=5840
[192.168.0.8] [192.168.0.7] RPC: C XID=59188767 PROG=Remote Statistics
VERS=434311 PROC=0 (?)
[192.168.0.7] [192.168.0.8] RPC: R XID=59188767 - Program unavailable
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014074 WIN<<2=5840
[192.168.0.8] [192.168.0.7] RPC: C XID=59188768 PROG=Remote Users
VERS=434311 PROC=0 (?)
[192.168.0.7] [192.168.0.8] RPC: R XID=59188768 - Program unavailable
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014102 WIN<<2=5840
[192.168.0.8] [192.168.0.7] RPC: C XID=59188769 PROG=NFS VERS=434311
PROC=0 (Do nothing)
[192.168.0.7] [192.168.0.8] RPC: R XID=59188769 - Program version mismatch
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014138 WIN<<2=5840
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 FIN ACK=2432014138
SEQ=2397105308 LEN=0 WIN<<2=5840
[192.168.0.7] [192.168.0.8] TCP: D=51008 S=2049 ACK=2397105309 WIN<<1=65358
[192.168.0.7] [192.168.0.8] TCP: D=51008 S=2049 FIN ACK=2397105309
SEQ=2432014138 LEN=0 WIN<<1=65358
[192.168.0.8] [192.168.0.7] TCP: D=2049 S=51008 ACK=2432014139 WIN<<2=5840
```

Utilizzo degli strumenti

Lo strumento principe per port scanning, fingerprint ed enumeration è NMAP. Nmap è un tool per l'analisi di rete nato per effettuare port-scanning ovvero individuare le porte sulle quali vi è un servizio di rete attivo.

Nelle sue ultime versioni questo strumento si è evoluto con funzionalità estremamente avanzate come il Fingerprint attivo che permette di individuare il tipo di sistema operativo e la versione dei servizi in ascolto, inviando in rete dei pacchetti sonda e comparando le risposte ottenute con un database interno.

Nel Listato 1 sP ping scan è utilizzato per l' host discovery. Nmap manderà un ICMP echo request e un pacchetto TCP SYN alla porta 80 . In questo modo è possibile ottenere informazioni sugli host interni ad una rete LAN

Altre opzioni interessanti per questo tipo di scan sono:

- PS22,23,25,80 pacchetto SYN verso le porte 22 23 25 80,
- PA22,23,25,80 pacchetto ACK al posto del SYN,
- PU udp ping,
- P0 no ping icmp echo request.

sS *Syn Scan* - partirà un pacchetto TCP con flag SYN attivo, se la porta da controllare è aperta risponderà

con un pacchetto TCP con i flag SYN|ACK attivi.

sF Fyn scan - verrà inviato un pacchetto TCP con flag FIN attivo, scan difficile da rilevare da un dispositivo IDS.

sA Ack scan - inviando pacchetti ACK si potrebbe stabilire se il firewall

interposto è di tipo stateful o è un semplice filtro di pacchetti.

sU UDP scans - vengono inviati header UDP vuoti (senza dati) alle porte di destinazione, questo scan risulta molto lento perché è basato su risposte icmp port unreachable limitate per evitare DoS.

p porte - indica le porte da controllare, nell'esempio si cerca di individuare un eventuale servizio voip (IAX o SIP) in ascolto sul server.

O OS detection - attiva la rilevazione del sistema operativo in ascolto, per poter funzionare questo tipo di scan devono essere rilevate almeno una porta aperta ed una chiusa ma non filtrata.

sV Version detection - attiva la rilevazione del tipo di servizio e della sua versione.

T< paranoid (0), sneaky (1), polite (2), normal (3), aggressive (4), insane (5)> - configura il Timing relativo al tempo dello scan ed all'impegno di rete.

Per quanto riguarda l'enumerazione dei servizi rpc è molto comodo l'utilizzo combinato di RPCscan con nmap.

RPCscan è uno strumento fondamentale per la rilevazione delle vulnerabilità, è uno scanner RPC (*Remote Program Call*) che consente di localizzare e identificare applicazioni RPC. Una volta scoperte le porte aperte di un host tramite un altro scanner di vulnerabilità, RPCscan invia ad ogni porta aperta un RPC nullo, provocando una risposta da tutte le applicazioni che utilizzano RPC attive in quel momento.

RPCscan viene lanciato automaticamente inserendo *-sV* in una normale scansione di nmap.

Vediamo un esempio di scansione di RPCscan (Listato 5).

Poi lanciamo nmap sull'host che abbiamo rilevato essere RPC-affected (vedi il Listato 6).

Nelle opzioni di Nessus è possibile decidere di utilizzare nmap come scanner. Nessus consente l'analisi delle vulnerabilità attraverso una serie di plugin che implementano test attivi o passivi sulle porte aperte.

Esistono due differenti versioni di Nessus, di cui una open source (GPL 2.2.).

Per la configurazione e la gestione è possibile utilizzare un comodo frontend in php chiamato inprotect. Grazie a inprotect e all'architettura

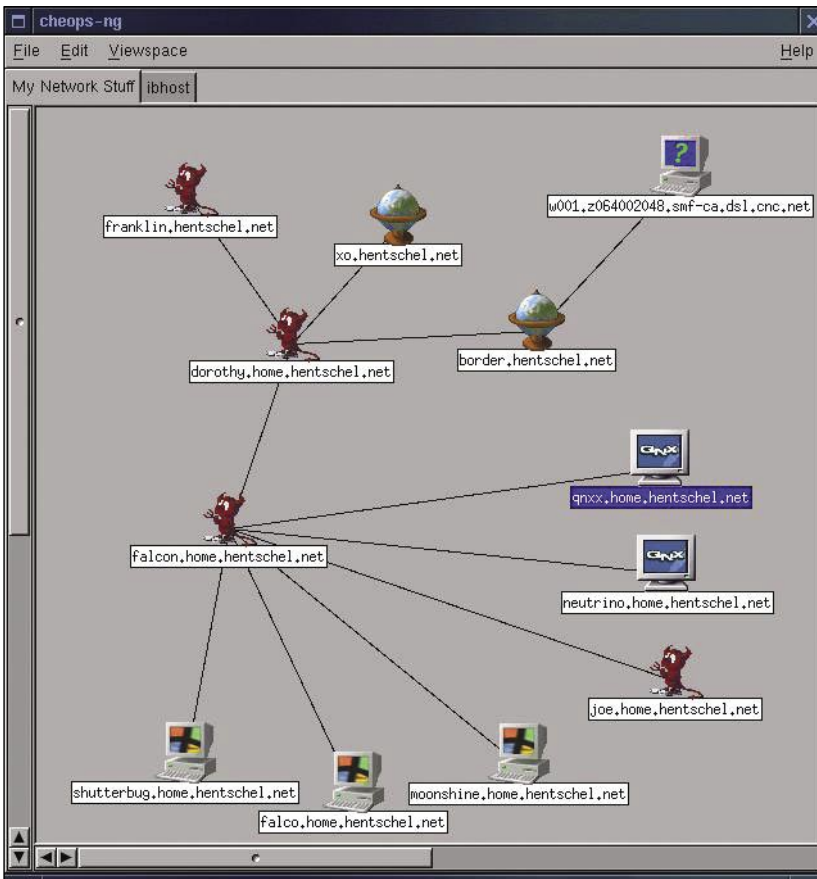


Fig.1: Cheops-ng

```

vollbracht@ds9.landeco.rwth-aachen.de:~/vollbracht
[root@ds9 vollbracht]# nmap -sS -n -O sisko
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.100.12):
(The 1532 ports scanned but not shown below are in state: closed)
Port      State  Service
7/tcp     open   echo
9/tcp     open   discard
13/tcp    open   daytime
17/tcp    open   qotd
19/tcp    open   chargen
135/tcp   open   loc-srv
139/tcp   open   netbios-ssn
5800/tcp  open   vnc-http
5900/tcp  open   vnc
Remote operating system guess: Windows NT4 or 95/98/98SE

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
[root@ds9 vollbracht]#
    
```

Fig.2: Nmap



client-server di Nessus è possibile gestire più server di analisi contemporaneamente, avendo, quindi, a disposizione diverse prospettive.

Giunto alla versione GPL 2.2.9 questo software open source vanta una grande comunità di persone che contribuiscono attivamente al progetto. In Nessus possiamo trovare migliaia di plugin (NASL) creati appositamente per rilevare ognuno una specifica vulnerabilità.

Per poter scaricare i plugin è necessario registrare un indirizzo email su nessus.org e attivare il server nessus con la chiave fornita all'indirizzo specificato. La dimensione complessiva dei plugin disponibili è attualmente di circa 5Mb.

L'installazione in nove semplici passi del server nessus e del client inprotect:

Primo è scaricare i sorgenti da nessus.org: `nessus-plugins-2.2.9.tar.gz`, `libnasl-2.2.9.tar.gz`, `nessus-core-2.2.9.tar.gz`, `nessus-libraries-2.2.9.tar.gz`. Poi dobbiamo estrarre i sorgenti di nessus:

```
tar -zxvf *.tar.gz.
```

In seguito installiamo da gestore pacchetti `apt` o `yum`: `flex`, `bison`, `shareutils`, `gcc`, `gtk+-devel` (solo per client nessus – facoltativo).

Quarto passo è compilare i sorgenti di nessus (esattamente nell'ordine indicato):

```
cd nessus-libraries
./configure && make && make install
cd ../libnasl
./configure && make && make install
cd ../nessus-core
./configure -disable-gtk && make &&
make install (per versione senza gtk)
./configure && make && make install
(per versione con gtk)
```

Successivamente dobbiamo includere nella path delle librerie in `/etc/ld.so.conf`, `/usr/local/lib` e lanciare `ldconfig` due volte:

- `vi /etc/ld.so.conf`,
- `ldconfig`.

Listato 6: # nmap -v -sR 192.168.0.7

```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-04-22 23:12 EDT
Initiating SYN Stealth Scan against 192.168.0.7 [1663 ports] at 23:12
Discovered open port 22/tcp on 192.168.0.7
Increasing send delay for 192.168.0.7 from 0 to 5 due to max_successful_tryno
increase to 4
Discovered open port 2049/tcp on 192.168.0.7
Discovered open port 111/tcp on 192.168.0.7
Discovered open port 886/tcp on 192.168.0.7
The SYN Stealth Scan took 10.26s to scan 1663 total ports.
Initiating RPCGrind Scan against 192.168.0.7 at 23:12
The RPCGrind Scan took 1.11s to scan 4 ports on 192.168.0.7.
Host 192.168.0.7 appears to be up ... good.
Interesting ports on 192.168.0.7:
(The 1659 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh
111/tcp   open  rpcbind (rpcbind V2-4) 2-4 (rpc #100000)
886/tcp   open  unknown
2049/tcp  open  nfs (nfs V2-3) 2-3 (rpc #100003)
MAC Address: 00:03:47:6D:28:D7 (Intel)
Nmap finished: 1 IP address (1 host up) scanned in 12.228 seconds
Raw packets sent: 1900 (76KB) | Rcvd: 1664 (76.5KB)
```

Poi si deve effettuare la registrazione su www.nessus.org per poter ricevere i plugin:

- codice di esempio :

```
9C74-4B2E-D7EE-7FBD-XXXX,
```

- `nessus-fetch -register`:

```
9C74-4B2E-D7EE-7FBD-XXXX.
```

Passo successivo è eseguire l'aggiornamento dei plugin: `nessus-update-plugins`

Poi inseriamo l'aggiornamento automatico dei plugin tramite una stringa nel crontab:

```
00 2 * * * root /usr/local/sbin/nessus
-update-plugins
```

L'ultimo passo è creare un utente nessus: `nessus-adduser`.

Al termine di questi passaggi Nessus è pronto per essere utilizzato.

Lanciano il comando `nessusd` viene eseguito il demone server nessus, dopodichè ci si può connettere eseguendo il client nessus da riga di comando.

Consigliamo tuttavia l'utilizzo del client web Inprotect. Vediamo brevemente come installarlo.

Prima dobbiamo scaricare ed estrarre inprotect da <http://inprotect.sourceforge.net/> - `tar -zxvf Inprotect-0.22.05.tar.gz` (si fa riferimento alla versione corrente).

Il pacchetto richiede l'installazione delle seguenti dipendenze: `apache`, `mysql`, `php`, `php-gd`, `perl`.

Installiamo le librerie del perl:

```
perl -MCPAN -e shell
cpan> install DBI
cpan> install MIME::Lite
cpan> install Parallel::ForkManager
cpan> install Date::Calc
```

Poi facciamo il setup di mysql (solo se non avete già modificato la password di root):

```
/etc/init.d/mysqld restart
mysql -u root
SET PASSWORD FOR root@localhost=
PASSWORD(<root_pwd>);
SET PASSWORD FOR root@
<TUO_HOSTNAME>=PASSWORD(<root_pwd>);
FLUSH PRIVILEGES;
```

Lanciamo l'installazione di Inprotect e selezioniamo l'opzione 2:

```
cd inprotect-0.22.05
sh install.sh
```

Creiamo account mysql per inprotect (presentato nel Listato 7).

Poi facciamo l'installazione del client web, lanciamo l'installazione e selezioniamo l'opzione 1: `sh install.sh`, selezionata l'opzione 1 inserite la path di destinazione, esempio :

```
/var/www/html/inprotect
```

Modifichiamo i file di configurazione di inprotect: cambiare in `/usr/local/etc/inprotect.cfg`:

```
DATABASEHOST=<hostname of web console>
DATABASEUSER=<inprotect_username>
DATABASEPASSWORD=<inprotect_password>
```

Poi cambiare in `/var/www/html/inprotect/config.php`:

```
$dbhost="<hostname of web console>";
$dbuname="<inprotect_username>";
$dbpass="<inprotect_password>";
```

Apriamo con un browser `http://localhost/inprotect` ed accediamo con Admin e password (case sensitive).

Nelle opzioni modifichiamo la password dell'utente Admin.

In Settings aggiungiamo un nuovo server nessus inserendo i dati richiesti (l'utente da utilizzare sarà quello impostato per nessus).

In console lanciamo il comando `updateplugins.pl` per popolare il database di Inprotect con i plugin di Nessus.

Andiamo in Settings, aggiungiamo un nuovo profilo ed associamogli un utente.

L'uso di *inprotect* come client di nessus permette di accodare più scansioni contemporaneamente. L'uso di un *db* è molto utile per l'archiviazione in modo da visualizzare i trend di miglioramento o peggioramento delle vulnerabilità in base all'aggiornamento delle rules o al tuning delle configurazioni.

La scelta di abilitare e di disabilitare i safe check permette l'utilizzo da parte di nessus, di tutti i plugin che potrebbero provocare un'interruzione del servizio in testing.

Come in tutti i sistemi di scansione potrebbero presentarsi casi di falsi positivi o falsi negativi. I report devono essere successivamente ricontrollati. Grazie alle funzionalità del frontend è facile segnalare i falsi positivi rilevati.

Per ridurre al minimo questi inconvenienti è consigliabile l'attivazione dei plugin in intelligent mode, in questo modo nessus non spreca risorse eseguendo plugin per check di servizi che non hanno una porta TCP/UDP aperta.

Inprotect risulta, inoltre, comodo se si desidera centralizzare su una console tutti i risultati.

La configurazione di nessus diventa molto intuitiva attraverso l'uso di *inprotect*; nella sezione delle opzioni di scansione è anche possibile selezionare tra le diverse opzioni di nmap (già citate in precedenza).

SARA è un sistema di analisi meno dettagliato rispetto a nessus, tuttavia permette un'analisi veloce delle porte aperte e delle vulnerabilità più rilevanti di modo da ottenere un report correlabile con nmap e nessus.

Dai risultati ottenuti e incrociati è possibile con buona approssimazione conoscere il servizio più vulnerabile e agire di conseguenza. Nell'eventualità in cui fosse, per esempio, il servizio di pubblicazione web, potremo utilizzare un tool specializzato per l'analisi dei servizi web: *Nikto*.

Nikto è un tool che non si deve trascurare se si desidera effettuare una buona analisi di rete. E' stato progettato per esaminare Server Web e cercare al loro interno oggetti che appartengono alle seguenti categorie: configurazioni errate o aperte ad attacchi, file e script con configurazioni di default, file e script notoriamente insicuri, software non aggiornato e quindi con falle di sicurezza.

Alla fine dello scan nikto sarà in grado di elencare le debolezze riscontrate ed anche dei consigli su cosa modificare per porvi rimedio. Può essere utilizzato rapidamente da linea di comando:

```
$ nikto -h host
```

Tuttavia, prima di eseguire il comando sopra descritto è utile fare delle precisazioni. Nikto è un tool scritto in Perl per l'analisi remota che esegue un numero estremamente alto di richieste al server indicato, e in alcuni casi ciò potrebbe portare al blocco del server. E' inoltre illegale usare Nikto (e strumenti analoghi) su server senza un'esplicita autorizzazione di chi ha commissionato il test.

Come utilizzare a fondo tutte le potenzialità di Nikto?

Il primo passo è conoscere su quali porte del server bersaglio sono in ascolto servizi che implementano il protocollo HTTP. Ovviamente nikto include al suo interno un port scanner, ma, dato che è scritto in Perl

Listato 7: Account mysql per inprotect

```
mysql -uroot -p<root_pwd>
USE mysql;
INSERT INTO user (host, user, password, select_priv, insert_priv,
  update_priv, delete_priv, create_priv, drop_priv) VALUES
  ('localhost', '<inprotect_username>', PASSWORD
  ('<inprotect_password>'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
INSERT INTO user (host, user, password, select_priv, insert_priv,
  update_priv, delete_priv, create_priv, drop_priv) VALUES
  ('127.0.0.1', '<inprotect_username>', PASSWORD
  ('<inprotect_password>'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
INSERT INTO user (host, user, password, select_priv, insert_priv,
  update_priv, delete_priv, create_priv, drop_priv) VALUES
  ('<hostname of web console>', '<inprotect_username>',
  PASSWORD('<inprotect_password>'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
FLUSH PRIVILEGES;
exit
```



risulta nettamente meno efficiente di Nmap, per questo gli sviluppatori hanno predisposto, nel file di configurazione, la possibilità di appoggiarsi a Nmap per il port scanning. E' bene assicurarsi, quindi, di avere nel file di configurazione (*nikto.conf* | *config.txt*) la seguente riga:

```
NMAP=/usr/bin/nmap # path  
to your nmap binary
```

Se non si desidera eseguire un portscan, o perchè si dispone già delle informazioni necessarie, oppure perchè si desidera testare solo determinate porte, è possibile utilizzare il parametro *-p* nei seguenti modi: *\$ nikto -h target -p 80 # standard*, *\$ nikto -h target -p 80,443 # controlla anche la porta https, nikto parla ssl!*, *\$ nikto -h target -p 80-90 # controlla dalla 80 alla 90*, *\$ nikto -h target -p 80-90,443 # combinazione delle precedenti*.

Nikto può eseguire scan anche all'interno di socket con lo stack ssl, se per varie ragioni nikto incotrassero problemi a rilevare se una porta implementa http o https è possibile forzare il protocollo https con il parametro *-s*, e tutte le porte saranno interrogate tramite richieste https.

E' utile anche rendere le richieste quanto più possibile varie o anonime: Nikto di default si annuncia ai server web con il suo nome; spetta all'operatore decidere se modificare il campo `USER_AGENT` con un valore differente, per farlo è necessario editare la variabile `$NIKTO(useragent)` direttamente in *nikto.pl*.

Si tenga presente che alcuni grandi siti web rispondono in modalità differente in base al browser da cui ricevono la richiesta. A questo proposito è utile sperimentare diverse alternative per il campo

```
USER_AGENT.
```

In ogni caso questo tipo di accorgimenti sono del tutto inutili per mascherare gli attacchi, dato che i log del server si riempiranno di centinaia (se non migliaia) di righe che riveleranno l'esecuzione dello scan. Nikto non è nato per essere silenzioso!

In secondo luogo tra voi e il server bersaglio potrebbero essere

presenti dei sistemi di tipo IDS o addirittura IPS che potrebbero vanificare i vostri tentativi, per questo all'interno di Nikto sono stati inclusi i meccanismi di evasione IDS implementati nella libreria libWhisker.

Per attivarli basterà utilizzare il parametro *-e* seguito da uno o più numeri identificativi delle tecniche di evasione da adottare. Per un loro elenco è sufficiente leggere l'output di `$ nikto -h`

Ovviamente non sono da considerare infallibili.

Un'altra tecnica utile consiste nel forzare lo scan generico ignorando gli header di risposta dei server web: il parametro *-g* permette di utilizzare tutti i test conosciuti e non basarsi sulle informazioni del server.

In caso di intoppi come, per esempio, pagine protette da http-authentication il parametro *-id* (seguito da `userid:password`) risulta molto comodo.

E' possibile anche combinare i vari attacchi tra di loro tramite l'opzione *-m*, anche se sconsigliabile dato che il numero di richieste generate e il conseguente utilizzo di banda crescerebbe in modo vertiginoso.

Altre opzioni utili sono:

Update che permette a nikto di cercare in internet aggiornamenti del db di attacchi. *F* permette di specificare il formato dell'output, normalmente è usato il formato di testo semplice TXT, è possibile utilizzare HTM per l'output in html utile per essere visualizzato su una pagina web, oppure il CSV (valori separati da virgola) utile per avere dei dati pronti per essere inseriti in un database o per definire il file dove scrivere l'output del programma, di default viene utilizzato lo standard *output*.

V che non sta per verbose, ma per virtual-host. E' utile in caso si voglia specificare a mano il virtual host da analizzare, rispetto a quello rilevato dallo scan iniziale.

H è il parametro che definisce il bersaglio. E' possibile anche passare il nome di un file al posto dell'indirizzo del bersaglio, ciò permette di definire bersagli multipli, la sintassi di ogni riga del file è molto semplice: `nomehost:porta,porta`.

Se insieme a questa opzione si utilizza anche il parametro *-p* le porte aggiuntive verranno analizzate per ogni server presente nel file.

Nel file di configurazione è possibile configurare un proxy da sfruttare per tutte le richieste che nikto effettuerà, oltre a poter definire altri parametri come nomi di utenti comuni da usare, directory standard da analizzare...

Analisi dei risultati

I risultati ottenuti dagli strumenti utilizzati per analizzare servizi e vulnerabilità permetteranno, nel caso si voglia procedere con un penetration testing, ad identificare gli attacchi corretti per ogni situazione.

Quando si ottengono informazioni di vulnerabilità è necessario interpretarle di modo da capire come sfruttare opportunamente le falle nel servizio. Nel caso in cui siate gli amministratori stessi del sistema che viene analizzato, i report prodotti risultano fondamentali per individuare i pacchetti da aggiornare o da configurare in modo corretto.

Un'ultima raccomandazione: l'uso di questi strumenti non va fatto in modo indiscriminato, poiché, senza le corrette autorizzazioni diventa illegale anche un semplice portscanning. ●

Cenni sull'autore

Snortattack.org, Portale orientato alla sicurezza, è il risultato della fusione di conoscenze e collaborazione del team. Compare in internet circa 10 mesi fa ma nasce nella mente dell'uno dei fondatori circa 2 anni orsono.

Grande punto di forza sono guide e script, per la non semplice installazione di Snort in Italiano e Inglese e non solo. Un forum e mailinglist che occorrono a tenere aggiornati gli utenti per nuove problematiche. Con Snortattack.org, il team, intende creare uno Snort User Group finalizzato alla collaborazione per l'Italia e tutto il resto del Mondo.