

UNIVERSITÀ DEGLI STUDI DI CAMERINO
SCUOLA DI ATENEO DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica Classe L-31



**METASPLOIT FRAMEWORK:
UN FRAMEWORK PER
IL PENETRATION TESTING**

RELATORE
Prof. Fausto Marcantoni

TESI DI LAUREA DI
Fabrizio Ippoliti

Anno Accademico 2009/2010

Dedicato al mio futuro nipotino/a; fra qualche mese porterà
ancora più gioia nella nostra famiglia.

Indice

1	Introduzione alla sicurezza informatica	1
1.1	In cosa consiste	1
1.2	Come ottenerla	2
1.3	Penetration Test	3
1.3.1	Perché effettuare un penetration test?	3
1.3.2	Cosa può essere testato?	4
1.3.3	Cosa dovrebbe essere testato?	5
2	Vulnerabilità e exploit	6
2.1	Introduzione	6
2.2	Vulnerability assessment	7
2.2.1	Nmap	8
2.3	Vulnerability scanner	13
2.3.1	Nessus	14
2.3.2	OpenVAS	19
2.3.3	NeXpose	23
2.4	Web vulnerability scanner	27
2.4.1	Nikto	28
2.4.2	HP WebInspect	29
2.5	Vulnerability Exploitation	30
2.5.1	Metasploit Project	30
2.5.2	Core Impact	31
2.5.3	Canvas	34
2.5.4	W3af	37

3	Metasploit Project	39
3.1	Panoramica	39
3.2	Metasploit Pro	43
3.3	Metasploit Express	43
3.4	Metasploit Framework	44
3.4.1	La struttura	45
3.4.2	Le interfacce	49
3.4.3	Armitage	54
3.4.4	Installazione	56
3.4.5	Aggiornamento	58
4	Conclusioni	60
A	Demo 1	62
B	Demo 2	69
	Ringraziamenti	75

Elenco delle figure

2.1	Parte dell'help di Nmap.	9
2.2	Esempio del risultato di una scansione eseguita con Nmap. . .	11
2.3	Zenmap: l'interfaccia di Nmap.	13
2.4	Gestione degli utenti in Nessus.	14
2.5	Visualizzazione di un report relativo ad una scansione.	17
2.6	Creazione di una nuova policy.	19
2.7	La sincronizzazione dei NVTs tramite feed free o commerciali.	21
2.8	I vari protocolli usati internamente da OpenVAS.	21
2.9	La schermata di accesso del client OpenVAS.	22
2.10	La schermata principale di OpenVAS.	23
2.11	Risultati di una scansione effettuata con Nexpose.	24
2.12	NeXpose e Metasploit sfruttano i dati forniti dalla comunità di Metasploit per identificare le vulnerabilità critiche di un ambiente.	25
2.13	Valutazione dei rischi e assegnazione priorità.	26
2.14	Esempio di valutazione di sicurezza tra diversi strati.	32
2.15	Schermata per il Network RTP.	34
2.16	La schermata di default dell'interfaccia grafica di Canvas. . . .	35
2.17	Avvio di exploit lato client tramite il server HTTP compreso. .	36
2.18	La gui di w3af.	38
3.1	Architettura del Metasploit Framework.	45
3.2	L'interfaccia a console di MSF: msfconsole.	50
3.3	Una interfaccia di MSF: msfcli.	52
3.4	La Gui in java del Metasploit Framework.	53

3.5	La schermata principale di Armitage, con alcuni host già inseriti.	55
3.6	La finestra di connessione di Armitage.	56
3.7	Esecuzione di msfupdate su una macchina con BackTrack 4. . .	59
A.1	Lancio di Armitage.	63
A.2	Selezione dell'exploit.	64
A.3	Un utente clicca su di un link contraffatto.	65
A.4	Esecuzione con successo dell'exploit.	66
A.5	Sono possibili vari modi per interagire con la sessione di Meterpreter.	67
A.6	Apertura di una sessione VNC tramite Meterpreter.	68
B.1	Esecuzione di una scansione tramite Nmap, da Armitage. . . .	70
B.2	Inserimento del range di indirizzi IP da scansionare.	71
B.3	Ricerca degli attacchi disponibili.	72
B.4	Visualizzazione degli attacchi relativi.	73
B.5	La funzione "check exploits...".	74

Elenco delle tabelle

2.1	Confronto abbonamenti Nessus: HomeFeed vs ProfessionalFeed.	16
2.2	Comparativa tra le varie versioni di NeXpose.	27
3.2	Comparativa tra le tre versioni di Metasploit.	42

Introduzione

L'epoca nella quale viviamo è più che mai caratterizzata dalla continua evoluzione in campo scientifico. L'informatica non è da meno: quotidianamente e in ogni parte del mondo possiamo ricevere un susseguirsi di aggiornamenti e novità tecnologiche, sia in campo software sia in quello dei componenti hardware.

In particolare, il crescente sviluppo delle tecnologie relative a internet e il costante processo di informatizzazione hanno avuto come diretta conseguenza l'inserimento in rete di un considerevole numero di sistemi informatici, in tutte le parti del globo. Essendo divenuta, con il passare degli anni, una costante di tutte le azioni che caratterizzano la nostra realtà. La quasi totalità dei gesti che compiamo, sono regolati da un qualche dispositivo informatico.

Sebbene spesso non ne siamo consapevoli, siamo giunti oramai a un livello di dipendenza da tali apparecchiature molto alto; così elevato da risultare preoccupante per certi versi. Basti considerare che, oltre ai servizi di e-commerce dediti alla vendita di beni e servizi, sono sempre maggiori i sistemi bancari, finanziari, assicurativi e di pubblica amministrazione che propongono, per mezzo delle potenzialità offerte da internet, svariati servizi agli utenti.

Tutta questa evoluzione non poteva far altro che catalizzare un interesse a livello mondiale, senza paragoni. Ed è proprio in un contesto del genere che si è iniziato a discutere con insistenza sulla sicurezza informatica. Fin da subito ci si è resi conto dell'importanza della questione che ben presto è divenuta un vero e proprio ramo dell'informatica.

Questo documento vuole trattare uno dei lati più interessanti dell'intero campo della sicurezza informatica: il penetration testing. In dettaglio, sarà analizzato approfonditamente il Metasploit Framework; un insieme di

strumenti open source in grado di fornire una valida alternativa ad altri strumenti a pagamento, e che può contare su una vastissima comunità che sviluppa quotidianamente nuovi exploit.

Struttura della tesi

Il primo capitolo cerca di introdurre il lettore nella giusta ottica, quella della sicurezza informatica. Per fare questo, è necessario familiarizzare con termini specifici, propri del dominio nel quale ci si sta addentrando.

Il secondo capitolo porterà nel vivo della discussione. Dopo aver spiegato il significato di vulnerabilità ed exploit, si analizzerà come poter effettuare un vulnerability assesment e con quali strumenti si dovrà lavorare. In generale è possibile individuare tre famiglie di strumenti: vulnerability scanner, web vulnerability scanner e strumenti per il vulnerability exploitation.

Il terzo capitolo rappresenta la parte principale dell'intero documento: tratta infatti del Metasploit Project e delle sue diverse versioni per eseguire penetration testing. L'attenzione maggiore è riservata al Metasploit Framework, il cui studio è il motivo dell'intero documento.

Sono altresì presenti due appendici che mostrano, anche grazie a molte immagini, come utilizzare il Metasploit Framework e Armitage insieme.

Abstract

The age in which we live is characterized by ever evolving science. Computer science is no exception: every day and in every part of the world, we can receive a series of upgrades and new technology, both in software and in hardware.

In particular, the increasing development of Internet-related technologies and the ongoing process of computerization have been as a direct result of the networking of a large number of computer systems, in all parts of the globe. Becoming, over the years, a constant of all the actions related to our reality. Almost all of the gestures we make are governed by computing device.

Although they often are not aware, we have arrived now at a level of dependence on these devices very high, so high as to be worrisome in some ways. Just consider that, in addition to e-commerce dedicated to the sale of goods and services, there are growing banking, financial, insurance and public service, who propose using the potential offered by the Internet, a variety of services to users.

All these developments could not help but to catalyze a worldwide interest, with no precedent. It is in such a context that has been in discussion with emphasis on computer security. From the beginning we have realized the importance of the issue that soon became a true branch of computer science.

This document will discuss one of the most interesting sides of the entire field of computer security: the penetration testing. In detail, we analyze in detail the Metasploit Framework, a set of open source tools can provide a viable alternative to other payment instruments, and can count on a huge community that develops every day new exploits.

Structure of the thesis

The first chapter seeks to introduce the reader in the right perspective, that of security. To do this, you need to become familiar with specific terms of their domain, in which one is entering.

The second chapter will lead to the heart of the debate. After explaining the meaning of vulnerability and exploit, we will analyze how to perform a vulnerability assesment and by what means you will be working. In general it is possible to identify three families of instruments: vulnerability scanner, web vulnerability scanner and tools for vulnerability exploitation.

The third chapter is the main part of the document as is in fact the Metasploit Project and its various versions to perform penetration testing. The main focus is reserved for the Metasploit Framework, the study of which is why the entire document.

There are also two appendices that show, even with many images, how to use the Metasploit Framework and Armitage together.

Capitolo 1

Introduzione alla sicurezza informatica

1.1 In cosa consiste

Con il termine sicurezza informatica si intende quel processo di prevenzione e individuazione dell'uso non autorizzato di un sistema informatico.

Nella parola prevenzione è racchiuso tutto quell'insieme di misure atte alla protezione di informazioni dall'accesso, dalla modifica o dal furto da parte di "attività" non previste. Tali attività indesiderate, possono essere eseguite da soggetti che abbiano un qualche interesse nei confronti di dati contenuti in un sistema. Tuttavia, devono essere comunque tenuti in considerazione anche eventi accidentali, quali il comportamento difforme dal consigliato di qualche utente interno oppure ai vari tipi di guasti che possono coinvolgere il sistema in analisi.

L'eventuale rilevamento consente di determinare se qualche soggetto abbia tentato di entrare in un determinato sistema, se abbia avuto successo e, in questo caso, le conseguenze di tale intrusione. La sua successiva individuazione potrebbe portare ripercussioni anche dal punto di vista legale: questo a seconda della gravità del gesto compiuto e della legislazione del paese dove è stata effettuata tale azione.

Nel concetto di sicurezza, inoltre, non va considerata solo la protezione dei

dati, intesi nel senso stretto del termine. È necessario concepire una visione più ampia dell'argomento; il concetto va esteso anche per quanto riguarda le persone, le comunicazioni e la protezione fisica dei computer.

1.2 Come ottenerla

È necessario innanzitutto chiarire che la sicurezza informatica non può essere intesa come un valore booleano, dove ad esempio 0 indica la sua assenza, mentre 1 il suo raggiungimento. Assume più concretezza impostare il discorso, e avere una visione d'insieme, orientato a identificare un *livello* di sicurezza. Questo permetterà di poter esprimere una valutazione sulla sicurezza di un sistema informatico, descrivendo quale livello di sicurezza si siano impegnati a raggiungere gli amministratori. Per quanto riguarda eventi di natura dolosa, si parlerà quindi, ad esempio, di un buon livello di sicurezza quando lo sfruttamento delle più note falle e vulnerabilità di sistema viene negato. Mentre, per quel che concerne le circostanze accidentali, si dovranno impostare politiche di backup e disaster recovery. Con il termine backup si intende la creazione di una copia di dati contenuti in un sistema informatico, conservandoli in dispositivi di archiviazione al fine di prevenire la perdita totale, o parziale, di dati informatici. L'attività di backup è un aspetto fondamentale della gestione di un computer: in caso di guasti, manomissioni, furti, ecc., ci si assicura l'esistenza di una copia dei dati. Invece, con disaster recovery, processo facente parte del più ampio campo della business continuity, si intende l'insieme di misure atte a ripristinare sistemi, dati e infrastrutture in seguito ad un disastro di qualsiasi genere (sia umano che naturale). Ogni organizzazione dovrebbe avere un piano di disaster recovery per tutelarsi contro situazioni di grave emergenza che possono causare interruzioni più o meno lunghe nel funzionamento di sistemi. Solo in questo modo, nel caso si verificasse un evento dannoso si potrebbe garantire la business continuity, proseguendo con le attività fino al ritorno alla normalità. Diversamente, l'organizzazione potrebbe subire danni economici incalcolabili: alcuni immediati, per stop forzato dei servizi, altri più evidenti per la perdita di dati.

Una volta assodato questo, è chiaro maturare l'idea che non esistano dei passi prefissati da seguire, per riuscire ad ottenere una soluzione assoluta adatta a risolvere il problema della sicurezza informatica, visto che il problema assume innumerevoli forme, e in altrettanto numerosi contesti.

1.3 Penetration Test

Nel seguito del documento verrà trattato soltanto un particolare aspetto, quello del *penetration test*. Termine spesso circondato da un po' di confusione, derivante dal fatto che si tratta di un settore relativamente recente e, com'era facile aspettarsi, in rapida evoluzione. Nella sua forma più semplice indica il testing di un dispositivo informatico, di una rete o di un'applicazione web per trovare vulnerabilità che un utente malintenzionato potrebbe sfruttare. Tale processo di valutazione viene definito attivo in quanto i sistemi informatici sono testati concretamente per individuare eventuali problemi di sicurezza, al contrario di un controllo esclusivamente teorico o cartaceo.

Con l'obiettivo principale di determinare le debolezze di sicurezza, il penetration testing è anche utilizzato per verificare la capacità di un'organizzazione nell'identificare e rispondere ad eventuali minacce. Il test, infine, può verificare la conformità di una politica di sicurezza adottata dall'ente e riuscire, in maniera pratica, a sensibilizzare i suoi dipendenti sull'importanza nell'adottare certe tecniche di protezione.

Parte del processo del penetration testing, in particolare il vulnerability assesment, può essere automatizzato attraverso l'utilizzo di applicazioni software che, in un arco di tempo relativamente breve, riescono a testare con una manciata di click migliaia di vulnerabilità note. Parte del prossimo capitolo sarà dedicato alla descrizione di questi software.

1.3.1 Perché effettuare un penetration test?

Da una prospettiva di business, il penetration testing aiuta a evitare delicate problematiche che potrebbero sorgere all'interno di un'azienda. Vediamo come scongiurarle:

- prevenire perdite finanziarie mediante frodi o attraverso le entrate perse a causa di sistemi business e processi inaffidabili;
- dimostrare una dovuta diligenza e il rispetto nei confronti dei clienti e degli azionisti. Il mancato rispetto, infatti, può comportare una diminuzione del business dell'organizzazione anche dovuto, in ultima analisi, alla raccolta di cattiva pubblicità;
- proteggere il brand, evitando la perdita di fiducia dei consumatori o degli azionisti e la reputazione aziendale. Una diminuzione del business può essere dovuta, in ultima istanza, anche per mezzo di pubblicità negativa.

1.3.2 Cosa può essere testato?

Possono essere testate tutte le componenti, umane o fisiche, con cui un'organizzazione acquisisce, memorizza e elabora informazioni: i sistemi con cui le informazioni vengono memorizzate, i canali di trasmissione che le trasportano, i processi e il personale che le gestiscono. Esempi di aree che vengono comunemente testate:

- prodotti acquistati al dettaglio: sistemi operativi, applicazioni, database, apparati di rete ecc;
- applicazioni di sviluppo personalizzato, su misura: siti web dinamici, applicazioni interne ecc;
- telefonia: accesso remoto, wardialing¹ ecc;
- wireless: WiFi, Bluetooth, IR, GSM, RFID ecc;
- risorse umane: processo di screening, social engineering ecc;

¹tecnica, che ai giorni d'oggi trova tuttavia una scarsa applicazione, la quale consiste nell'uso di un modem per scansionare automaticamente una lista di numeri telefonici, alla ricerca di un segnale di portante del modem.

- risorse fisiche: controllo degli accessi, dumpster diving, ovvero l'attività di rovistare nella spazzatura a caccia di documenti contenenti informazioni riservate ecc.

1.3.3 Cosa dovrebbe essere testato?

Idealmente, ogni organizzazione dovrebbe aver già effettuato una valutazione dei rischi. In questo modo, sarà già a conoscenza delle principali minacce (come, ad esempio, errori di comunicazione, falle nel sistema di e-commerce, perdita di informazioni confidenziali ecc), ma avrà a questo punto la possibilità di utilizzare il penetration test per identificare eventuali vulnerabilità che fanno riferimento a queste minacce. Nel caso in cui non si sia condotta una valutazione dei rischi, è normale iniziare analizzando le aree di maggiore esposizione, come siti web, portali di posta elettronica e piattaforme di accesso remoto. Il tutto, ovviamente, sarà da decidere in relazione alle effettive esigenze di cui l'organizzazione necessita svolgendo il proprio lavoro.

Capitolo 2

Vulnerabilità e exploit

2.1 Introduzione

Dopo aver visto brevemente il contesto nel quale ci stiamo inserendo, passeremo ora ad aspetti più tecnici della materia.

Possiamo, quindi, dire che le vulnerabilità di un sistema consistono nelle sue, varie ed eventuali, falle di sicurezza. che il sistema presenta, sotto diverse forme. Il termine *exploit*, invece, è usato per identificare un software, o una particolare tecnica, che permette di sfruttare bug o vulnerabilità, al fine di provocare comportamenti indesiderati o non previsti nei confronti di un sistema software o hardware. Evidente è la relazione che lega i due termini: ad una vulnerabilità è possibile associare un exploit; ma anche una soluzione. Questo documento non si occuperà di soluzioni, bensì di come scoprire e sfruttare vulnerabilità, attraverso il test di exploit.

Con il termine *0day exploit*, inoltre, viene definito quel particolare tipo di exploit che, associato ad una vulnerabilità resa nota al pubblico mondiale solamente da poche ore, colpisce la quasi totalità dei sistemi coinvolti. Questo avviene in quanto non c'è stato tempo necessario e sufficiente affinché un aggiornamento, o una patch, siano stati rilasciati riuscendo così a correggere o risolvere la vulnerabilità.

2.2 Vulnerability assessment

Volendo valutare il livello di sicurezza di una certa configurazione di un sistema, il primo passo da compiere è sicuramente quello di eseguire un *vulnerability assessment*. Tale valutazione delle vulnerabilità consiste di un processo volto a valutare l'efficacia dei meccanismi di sicurezza e quindi alla individuazione, alla quantificazione e all'assegnazione di priorità riguardo le eventuali vulnerabilità di un sistema. Le operazioni appena descritte hanno lo scopo, una volta trovate le falle di sicurezza, di migliorare il sistema e prevenire eventuali attacchi basati sui dati riscontrati.

Questa ricerca può essere effettuata da vari strumenti, alcuni free altri a pagamento. Uno scanner di vulnerabilità può eseguire prove "invasive" o non. Un test invasivo tenta di sfruttare la vulnerabilità a proprio piacimento, rischiando di mandare in crash, o comunque di alterare, la destinazione remota. Un test non invasivo cerca invece di non arrecare alcun danno al bersaglio. La prova consiste di solito nel controllo della versione del servizio remoto, o nel verificare se alcune opzioni vulnerabili sono abilitate. La prima tipologia di test è in genere quella più accurata, ma ovviamente non può essere eseguita in un ambiente di produzione. Al contrario, un test non invasivo non può determinare per certo se un servizio installato è effettivamente vulnerabile, bensì solo se può essere vulnerabile.

Uno *scanner di vulnerabilità* si differenzia da uno strumento usato invece per il penetration testing, dal modo in cui sfrutta le vulnerabilità. Vale a dire che uno scanner assicura che la vulnerabilità esiste, ma non tenta di compromettere il software vulnerabile. Un eventuale crash o negazione del servizio è solo un effetto collaterale di un test invasivo, non certo un obiettivo.

In ogni caso, il primo passo da eseguire, genericamente, è quello di cercare di ottenere più informazioni possibili sulla macchina target. Un tentativo possibile è quello di lanciare un *port scanning*: ovvero una scansione di un indirizzo IP o di una subnet, per ricavare informazioni sui servizi attivi di una macchina.

2.2.1 Nmap

Nmap, letteralmente Network Mapper, è uno strumento open source per la network exploration e l'auditing. È stato progettato per scansionare rapidamente reti di grandi dimensioni, ma è indicato anche per l'utilizzo verso singoli host. Nmap usa pacchetti IP cosiddetti "raw" (grezzi, non formattati) in varie modalità per determinare quali host sono disponibili su una rete, che servizi (nome dell'applicazione e versione) vengono offerti da questi host, che sistema operativo, con relativa versione, è in esecuzione, che tipo di firewall e packet filters sono usati, e molte altre caratteristiche. Proprio quella di determinare da remoto quale sistema operativo è installato su una macchina, è una delle più famose caratteristiche di NMap. Questo avviene tramite il *fingerprint* dello stack TCP/IP. Nmap invia una serie di pacchetti TCP ed UDP all'host remoto ed esamina ogni bit ricevuto in risposta. Ogni fingerprint comprende una descrizione del sistema operativo ed una classificazione indicante: il vendor (per esempio Sun), il sistema operativo (per esempio Solaris), la versione (per esempio 10) ed il tipo di device ("general purpose", router, switch, "game console", ecc.).

Nmap è multiplatforma; è infatti disponibile per la maggior parte dei sistemi operativi disponibili.

Nonostante Nmap sia comunemente usato per audits di sicurezza, molti sistemisti e amministratori di rete lo trovano utile per tutte le attività quotidiane come ad esempio l'inventario delle macchine presenti in rete, per gestire gli aggiornamenti programmati dei servizi, e per monitorare gli host o il loro uptime.

```
root@bt:~# nmap -h
Nmap 5.35DC1 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
EXAMPLES:
  nmap -v -A scanme.nmap.org
  nmap -v -sn 192.168.0.0/16 10.0.0.0/8
  nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (http://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
```

Figura 2.1: Parte dell'help di Nmap.

L'output di Nmap è una scansione di un elenco di obiettivi, con informazioni supplementari per ciascuno a seconda delle opzioni usate. Tra queste informazioni è fondamentale la "tabella delle porte interessanti". Questa tabella elenca il numero della porta e il protocollo, il nome del servizio e lo stato attuale. Lo stato può essere:

- open: un'applicazione accetta attivamente su questa porta connessioni TCP o UDP. La ricerca di questo tipo di porte è spesso l'obiettivo primario del port scanning. Chi si dedica alla sicurezza sa che ogni porta aperta è una possibile strada verso un attacco. Gli attaccanti e i penetration tester hanno come obiettivo quello di trovare e trarre vantaggio dalle porte aperte, mentre d'altro canto gli amministratori di rete e i sistemisti provano a chiuderle o a proteggerle con firewall senza limitare gli utenti autorizzati al loro uso. Le porte aperte sono anche interessanti per le tutta una serie di scansioni non indirizzate unicamente alla sicurezza, perché mostrano che servizi sono disponibili in una rete.

- **filtered**: Nmap non può determinare con esattezza se la porta sia aperta o meno, perché un filtro di pacchetti impedisce ai probe di raggiungere la porta. Questo filtro può esser dovuto a un firewall dedicato, alle regole di un router o a un firewall software installato sulla macchina stessa. Queste porte forniscono poche informazioni e rendono frustrante il lavoro del penetration tester. A volte esse rispondono con un messaggio ICMP del tipo 3, codice 13 ("destination unreachable: communication administratively prohibited"), ma in genere sono molto più comuni i filtri di pacchetti che semplicemente ignorano i tentativi di connessione senza rispondere. Questo obbliga Nmap a riprovare diverse volte, semplicemente per essere sicuri che il pacchetto non sia stato perduto a causa di una congestione di rete o di problemi simili piuttosto che dal firewall o dal filtro stesso. Questo riduce drammaticamente la velocità della scansione.
- **closed**: una porta chiusa è accessibile (riceve e risponde ai pacchetti di probe di Nmap) ma non vi è alcuna applicazione in ascolto su di essa. Esse possono rendersi utili nel mostrare che un host è attivo su un indirizzo IP (durante l'host discovery o il ping scanning) o in quanto parte integrante dell'Operating System discovery. Poiché una porta chiusa è raggiungibile, può essere interessante effettuare una scansione più tardi nel caso alcune vengano aperte. Chi amministra una macchina o una rete può voler bloccare tali porte con un firewall; in questo caso esse apparirebbero come filtrate, come mostrato nel punto precedente.
- **unfiltered**: indica che una porta è accessibile, ma che Nmap non è in grado di determinare se sia aperta o chiusa. Solo la scansione di tipo ACK, usata per trovare e classificare le regole di un firewall, posiziona una porta in questo stato. Una scansione di porte in questo stato mediante altri tipi di scansione come il Window scan (scan per finestre di connessione), il SYN scan o il FIN scan aiuta a determinare se la porta sia aperta o chiusa.
- **open|filtered**: Nmap posiziona le porte in questo stato quando non è in

grado di determinare se una porta sia aperta o filtrata. Questo accade in quelle scansioni per le quali una porta aperta non risponde in alcun modo. La mancanza di informazioni può significare inoltre che un filtro di pacchetti ha lasciato cadere ("drop") il probe o qualsiasi risposta sia stata generata in seguito a questo. Scansioni che classificano porte in questo stato sono le scansioni IP, UDP, FIN, Null, e Xmas.

- closed|filtered: stato usato quando Nmap non è in grado di determinare se una porta sia chiusa o filtrata. Esso viene usato solo per l'IPID Idle scan.

```

root@bt:~# nmap -A 192.168.1.5 playground

Starting Nmap 5.35DC1 ( http://nmap.org ) at 2011-03-29 17:04 CEST
Nmap scan report for 192.168.1.5
Host is up (0.00067s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE          VERSION
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn     netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:D2:D3:0A (VMware)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=5.35DC1%0=3/29%0T=135%CT=1%CU=40269%PV=Y%DS=1%DC=D%G=Y%M=000C29%T
OS:M=4D91F511%P=1686-pc-linux-gnu)SEQ(SP=FD%GCD=1%ISR=10F%TI=I%CI=I%II=I%SS
OS:=S%TS=U)SEQ(SP=FD%GCD=2%ISR=110%TI=I%CI=I%II=I%SS=S%TS=U)SEQ(SP=FE%GCD=1
OS:=ISR=110%TI=I%CI=I%II=I%SS=S%TS=U)OPS(O1=M5B4NW2NNS%02=M5B4NW2NNS%03=M5B
OS:4NW2%04=M5B4NW2NNS%05=M5B4NW2NNS%06=M5B4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%
OS:W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%DF=Y%T=80%W=FFFF%0=M5B4NW2NNS%CC=N%0=)T1
OS:(R=Y%DF=Y%T=80%S=0%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=N%T=80%W=0%S=Z%A=S+F=AR%0
OS:=%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=FFFF%S=0%A=S+%F=AS%0=M5B4NW2NNS%RD=0%Q=)T4(
OS:R=Y%DF=N%T=80%W=0%S=A%A=0%F=R%0=%RD=0%Q=)T5(R=Y%DF=N%T=80%W=0%S=Z%A=S+%F
OS:=AR%0=%RD=0%Q=)T6(R=Y%DF=N%T=80%W=0%S=A%A=0%F=R%0=%RD=0%Q=)T7(R=Y%DF=N%T
OS:=80%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=B0%UN=0%RIPL=G%RID
OS:=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=80%CD=Z)

Network Distance: 1 hop
Service Info: OS: Windows

Host script results:
|_nbstat: NetBIOS name: VIRTUALUSER, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:d2:d3:0a (VMware)
|_smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   Name: WORKGROUP\VIRTUALUSER
|_ System time: 2011-03-29 17:04:49 UTC+2
|_smbv2-enabled: Server doesn't support SMBv2 protocol

```

Figura 2.2: Esempio del risultato di una scansione eseguita con Nmap.

La tabella delle porte può anche includere dettagli quali le versioni dei software disponibili, se è stata usata l'opzione appropriata. Quando viene richiesto una scansione IP ("-sO"), Nmap fornisce informazioni sui protocolli IP supportati anziché sulle porte in ascolto. In aggiunta alla tabella delle

porte notevoli, Nmap può fornire ulteriori informazioni sugli obiettivi come ad esempio i nomi DNS risolti (reverse DNS names), il tipo di device e l'indirizzo fisico (MAC address).

Zenmap è la Gui ufficiale di Nmap; esso è disponibile per Linux, Windows, Mac OS X e BSD. L'applicazione è gratis ed open source; mira a rendere Nmap più facile da usare per i principianti ma, al tempo stesso, fornendo comunque funzionalità avanzate per gli utenti più esperti. Scansioni eseguite frequentemente possono essere salvate come profili per aumentare la praticità d'uso.

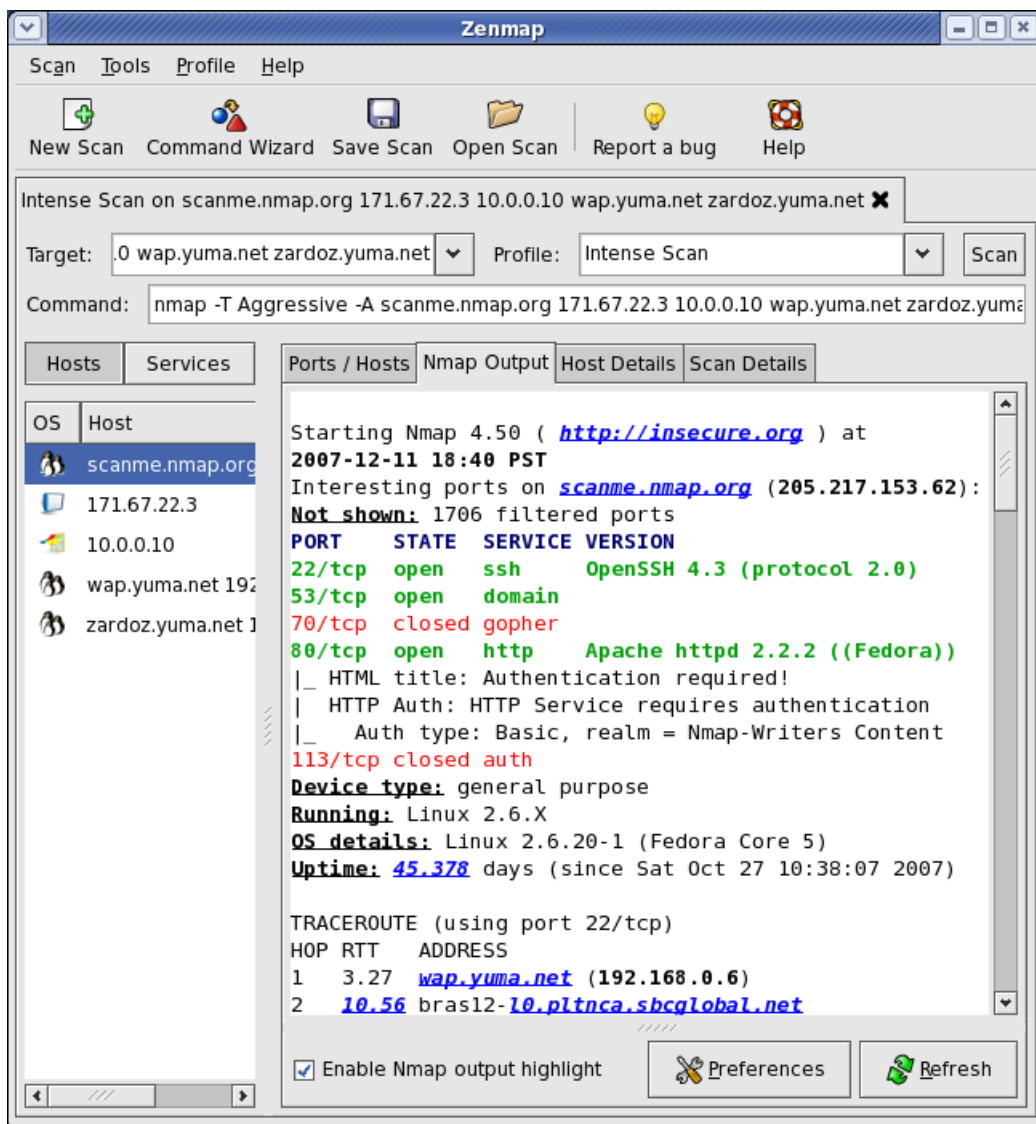


Figura 2.3: Zenmap: l'interfaccia di Nmap.

2.3 Vulnerability scanner

Nel seguito riporteremo quelli che vengono ritenuti tra i migliori software capaci di effettuare un vulnerability assessment.

indicante le varie licenze di abbonamento, annuali, e per ognuna i differenti servizi offerti.

- Registered Scanner: ogni installazione di Nessus richiede la registrazione presso il sito ufficiale. Per eseguire l'aggiornamento dei plugin sarà infatti necessario procedere con l'attivazione. Verrà così inviato il codice della licenza per posta elettronica. Da notare che tale license key è valida solo per una singola installazione di Nessus. Questo rappresenta una condizione "sine qua non" affinché lo scanner sia utilizzabile, a prescindere da quale abbonamento si voglia poi sottoscrivere.
- HomeFeed Subscription: abbonamento disponibile solo per un uso personale, in ambiente "domestico": trattandosi infatti di una sottoscrizione non commerciale. La limitazione è chiara: l'utilizzo permesso da questo abbonamento dovrà essere esclusivamente per rilevare vulnerabilità solo sul proprio sistema personale (o per la propria rete personale) che si utilizza per fini non commerciali. L'utilizzo dei plugin non è quindi concesso per attività all'interno di nessuna azienda, ente o organizzazione.
- ProfessionalFeed Subscription: sottoscrizione richiesta per qualsiasi utilizzo di Nessus fuori dalle mura domestiche. Essendo un abbonamento commerciale, consente ovviamente l'uso dello scanner per rilevare vulnerabilità presenti sul proprio sistema e sulla propria rete, ma anche il lancio di test nei confronti di sistemi e reti di terzi per i quali si effettuano servizi di scansione, di valutazione delle vulnerabilità o comunque di consulenza per la sicurezza. Nel costo della sottoscrizione è compreso anche un supporto tramite e-mail.

Nel particolare, i benefici di un abbonamento ProfessionalFeed rispetto a uno HomeFeed sono rappresentati nella seguente tabella:

Caratteristica	HomeFeed	ProfessionalFeed
Utilizzo	non commerciale	commerciale
Aggiornamento delle vulnerabilità in tempo reale	si	si
Possibilità di download di una VMware Virtual Appliance	no	si
Verifica di policy che fanno riferimento a standard quali PCI, FDCC, CIS ecc	no	si
Rilevamento e auditing di dispositivi SCADA per quanto riguarda la sicurezza e la configurazione	no	si
Supporto commerciale	no	si
Prezzo	\$0	\$1.200 annuali

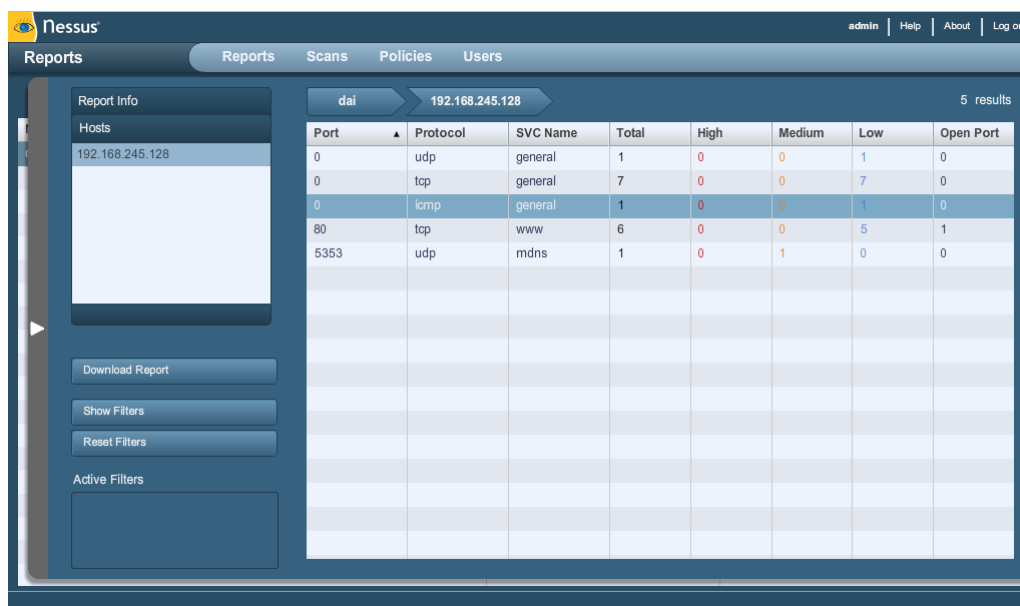
Tabella 2.1: Confronto abbonamenti Nessus: HomeFeed vs ProfessionalFeed.

Utilizzo

È possibile eseguire una scansione dall'interno della rete per ottenere quante più informazioni è possibile su potenziali vulnerabilità o debolezze del sistema target. In alternativa, è possibile lanciare la scansione dell'esterno della rete, per comprendere meglio ciò che effettivamente un attaccante è in grado di vedere. Solitamente si vorrà fare un'approfondita analisi di tutti i server a livello di interfaccia tra la rete locale e internet, ovvero quell'insieme di server che viene normalmente definito come DMZ (zona demilitarizzata). Quest'ultimo termine specifica una sottorete inserita come "zona neutrale" tra la rete privata di un'organizzazione e internet, che permette connessioni esclusivamente verso l'esterno: server di posta elettronica, server HTTP con applicazioni web o server VPN.

Dapprima, Nessus lancia una scansione delle porte per identificare i servizi in esecuzione e il sistema operativo della macchina o della sottorete target. La fase di scansione delle porte è fondamentale: viene utilizzata da Nessus per sapere quali plugin sono pertinenti (ad esempio Apache o ISS plugin per un server web o vulnerabilità del sistema operativo) e quale servizio è attivo su tale porta. Nessus è anche in grado di rilevare servizi su porte non

standard. Se la scansione lanciata coinvolge una rete di grande dimensioni, è più efficiente posizionare un server Nessus per ogni segmento di rete. Se si conoscono dettagli della macchina sulla quale si esegue la scansione, quali sistema operativo o servizi in esecuzione sull'host, è possibile impostarli al momento della configurazione della policy.



The screenshot shows the Nessus Reports interface. The main content area displays a table of scan results for host 192.168.245.128. The table has the following columns: Port, Protocol, SVC Name, Total, High, Medium, Low, and Open Port. The data rows are as follows:

Port	Protocol	SVC Name	Total	High	Medium	Low	Open Port
0	udp	general	1	0	0	1	0
0	tcp	general	7	0	0	7	0
0	icmp	general	1	0	0	1	0
80	tcp	www	6	0	0	5	1
5353	udp	mdns	1	0	1	0	0

Figura 2.5: Visualizzazione di un report relativo ad una scansione.

Al termine di una scansione, Nessus genera un report indicante una lista di tutte le porte aperte e i potenziali rischi associati. Sarà poi possibile esportare tali report in vari formati, come ad esempio HTML o PDF. Tuttavia, è necessario prendere con dovuta cautela questi risultati: spesso potrebbero esserci dei falsi positivi nelle vulnerabilità riscontrate da Nessus. Vediamo brevemente quali circostanze possano portare ad esiti non corretti:

- un firewall o un altro dispositivo di sicurezza possono aver rilevato la scansione in corso. Ad esempio se il nostro firewall è in grado di rilevare la scansione dopo pochi secondi, blocca tutto il traffico generato da Nessus. Il rapporto mostra un sacco di porte aperte che non esistono sul target perché ha travisato la perdita di pacchetti.

- alcuni controlli di vulnerabilità sono troppo superficiali. A volte, un plugin cerca la versione solamente nel banner. Questo potrebbe non essere sufficiente per sapere se il servizio è a tutti gli effetti vulnerabile.
- è possibile riscontrare un falso rilevamento se la destinazione è dietro un Port Address Translator¹, in quanto ogni porta può corrispondere a un diverso sistema operativo.

In generale possiamo dire che i risultati della scansione evidenziano potenziali problemi che dovrebbero poi essere controllati uno ad uno.

Policy e plugin

Invece di eseguire una scansione completa, è possibile personalizzare le aree che dovrebbero essere controllate. Riducendo il numero dei controlli che vengono fatti e configurando le impostazioni, normalmente di default, è possibile sia ridurre la durata della scansione sia migliorare la sua accuratezza. Le impostazioni sono associate a una policy. Questo significa che ogni target che richiede particolari impostazioni (password diverse per esempio), impone la propria policy. Una volta installato, Nessus non ha nessuna policy: prima di lanciare una scansione è necessario quindi averne creata una.

Una volta dato un nome alla policy, è possibile settare tutte le credenziali della macchina target conosciute. I plugin disponibili sono attualmente più di 40.000; per una migliore organizzazione sono raggruppati in poco più di 40 famiglie di plugin (ad esempio CISCO, FTP, Web Servers ecc). Per ogni plugin è disponibile una breve delucidazione, contenete questi campi: Synopsis, Description, Solution e See Also.

¹La Port Address Translation è una particolare tecnica di NAT che consente di far corrispondere più indirizzi IP di una rete privata a un singolo indirizzo IP di una rete pubblica.

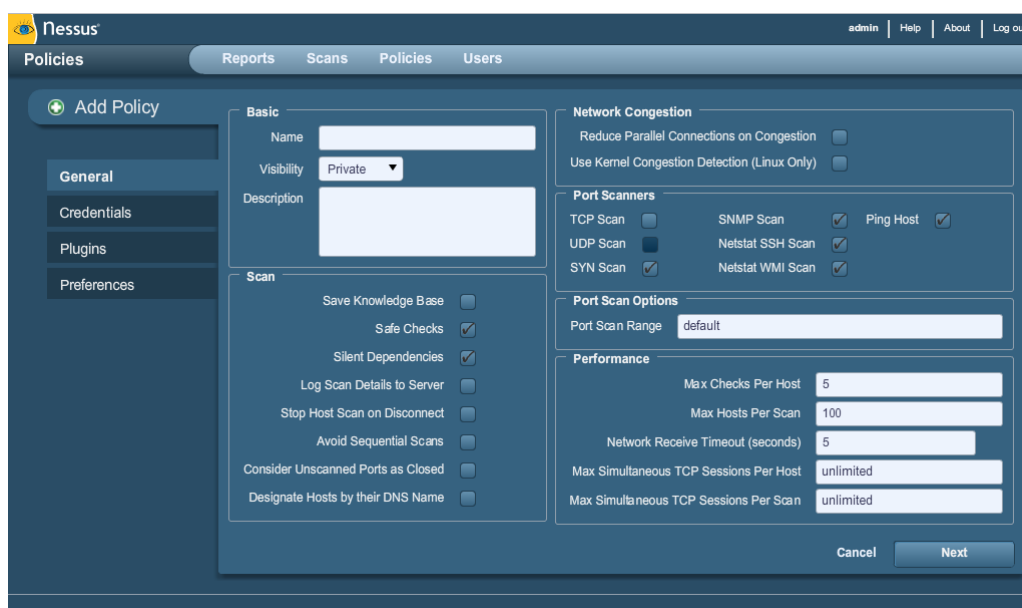


Figura 2.6: Creazione di una nuova policy.

2.3.2 OpenVAS

Open Vulnerability Assessment System è un framework di vari servizi e strumenti che offrono un completo e potente scanner di vulnerabilità e una soluzione per la loro gestione. Il nome originario dell'applicazione era "GNessus", nata come fork di Nessus. La nascita di questo fork, attualmente multiplatforma, fu la naturale risposta da parte della comunità open source, al cambio di licenza con cui i dirigenti della Tenable Security decisero di rilasciare Nessus a partire dalla versione 2.5. A partire da tale versione infatti, quello che era considerato il miglior security scanner open source divenne software proprietario. OpenVAS, al contrario, è rilasciato sotto licenza GPL. Nel seguito, vedremo spesso riferimenti a Nessus, in quanto dal punto di vista implementativo molti aspetti sono strettamente correlati.

OpenVAS è basato su un'architettura server/client che comprende i seguenti componenti:

- OpenVAS-Server: è il nucleo di OpenVAS. Contiene le funzionalità usate per la scansione di un vasto numero di macchine target, con una

velocità elevata. Le scansioni sono sempre originate dall'host nel quale OpenVAS-Server è in esecuzione; pertanto, questa macchina deve essere in grado di raggiungere gli obiettivi previsti. Il server richiede 3 altri moduli:

- OpenVAS-Libraries: modulo contenente le funzionalità usate dal server OpenVAS.
 - OpenVAS-LibNASL: i *Network Vulnerability Tests*, NVTs, sono scritti nel "Nessus Attack Scripting Language", NASL. Questo modulo contiene, invece, le funzionalità di cui il server OpenVAS necessita per interfacciarsi con NASL.
 - OpenVAS-Plugins: modulo contenente un set base di NVTs, che potrà poi essere aggiornato.
- OpenVAS-Client: controlla il server OpenVAS: elabora i risultati della scansione e li visualizza all'utente. Il client può essere eseguito in qualsiasi macchina in grado di connettersi all'OpenVAS-Server e può controllare più server.

OpenVAS è multiplatforma: i più diffusi sistemi operativi sono attualmente supportati.

Lo scanner esegue gli attuali NVTs, che sono forniti tramite aggiornamenti attraverso l'"OpenVAS NVT Feed" oppure da un servizio di feed commerciale. Il feed pubblico contiene più di 20.000 NVTs; cifra che, di solito, sale quotidianamente. Per la sincronizzazione on line, è possibile usare un apposito comando oppure, in alternativa, è anche scaricabile un singolo archivio contenente tutti i NVTs. I file del OpenVAS NVT Feed sono firmati dal certificato denominato: "OpenVAS: Transfer Integrity". Sebbene la presenza di tale firma non sia per garantire un controllo o un certo giudizio di qualità sullo script stesso, essa è destinata esclusivamente per aiutare l'utente a verificare la corretta integrità dei NVTs dopo una sincronizzazione. Oltre l'OpenVAS NVT Feed, è disponibile anche un feed commerciale, offerto dalla "Greenbone Networks" e si chiama Greenbone Security Feed.

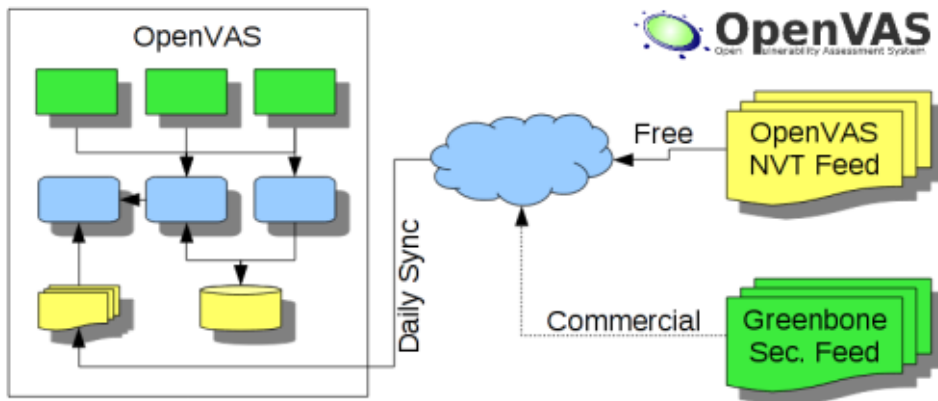


Figura 2.7: La sincronizzazione dei NVTs tramite feed free o commerciali.

L'*OpenVAS Manager* è il servizio principale che consolida la semplice scansione delle vulnerabilità, in una soluzione completa di gestione delle vulnerabilità. Dato che tutte le informazioni sono gestite dal Manager, è possibile diversi client leggeri che si comporteranno in modo coerente. L'OpenVAS Manager controlla anche un database SQL (basato su sqlite), dove tutta la configurazione e i risultati della scansione sono memorizzati centralmente.

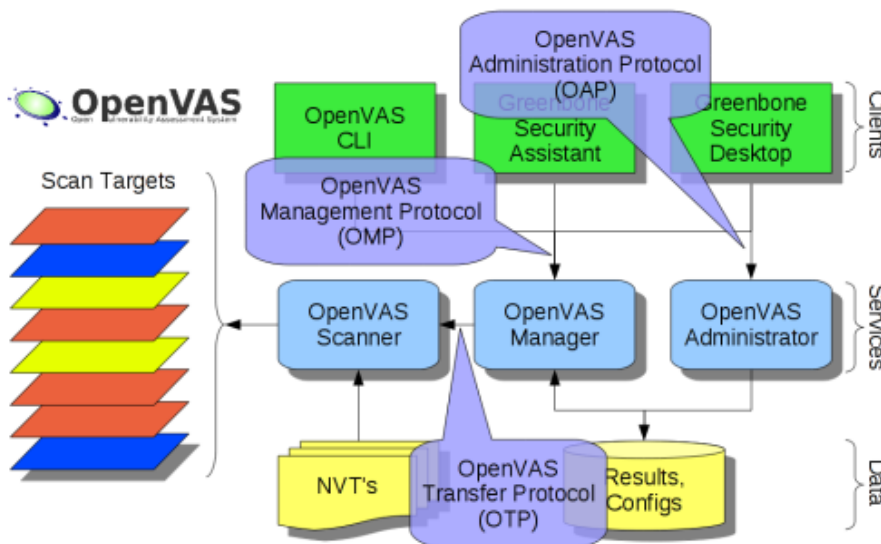


Figura 2.8: I vari protocolli usati internamente da OpenVAS.

Sono disponibili due client per l'OMP: OpenVAS Management Protocol.

Il Greenbone Security Assistant (GSA) è un servizio web che offre una interfaccia utente snella per i browser web. GSA utilizza fogli di stile XSL che convertono le risposte OMP in HTML. Il Greenbone Security Desktop (GSD) è un client desktop basato su Qt per OMP. Funziona sulle varie versioni di Linux, di Windows e altri sistemi operativi.

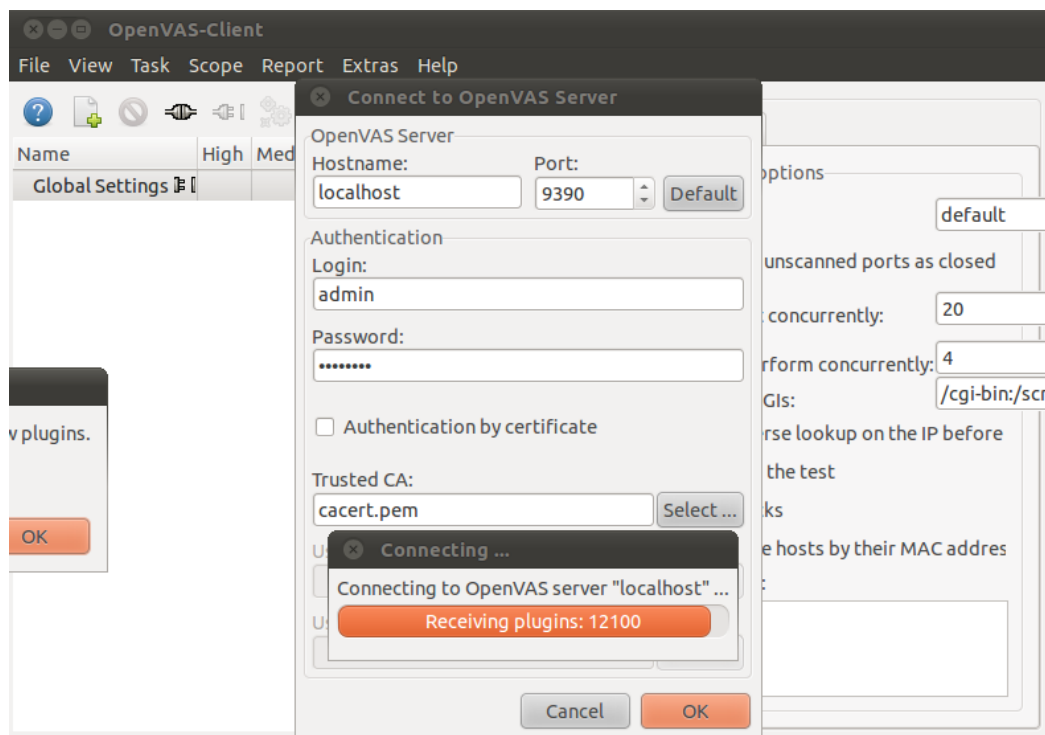


Figura 2.9: La schermata di accesso del client OpenVAS.

L'*OpenVAS Administrator* agisce come uno strumento a riga di comando o come un demone di servizio completo, offrendo l'*OpenVAS Administration Protocol* (OAP). Le attività più importanti svolte sono la gestione degli utenti e la gestione dei feed. GSA supporta OAP e gli utenti di tipo Admin possono accedere alle funzionalità proprie di OAP.

La maggior parte degli strumenti elencati precedentemente, condividono le funzionalità che sono aggregate nelle *OpenVAS Libraries*.

L'*OpenVAS Scanner* offre il protocollo di comunicazione OTP (*OpenVAS Transfer Protocol*), il quale consente di controllare l'esecuzione della scansione.

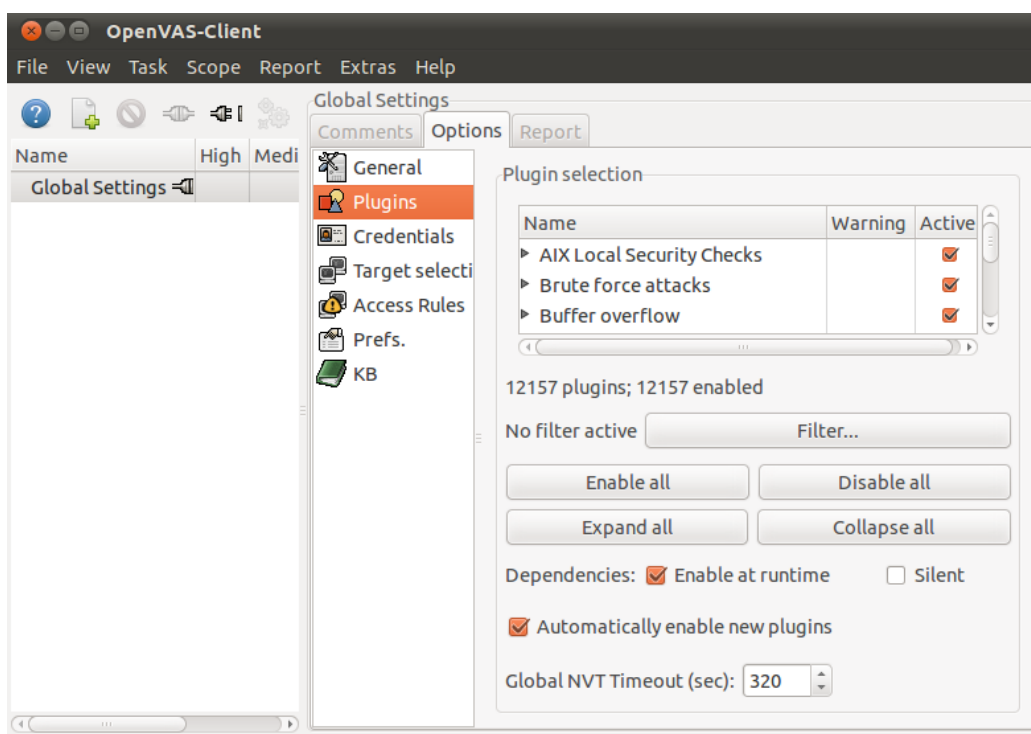


Figura 2.10: La schermata principale di OpenVAS.

2.3.3 NeXpose

NeXpose è un software per la valutazione delle vulnerabilità dei software, sviluppato da Rapid7. Nelle sue scansioni, analizza accuratamente le applicazioni Web, database, reti, sistemi operativi e altri software per individuare le minacce, valutarne il rischio e mettere a punto un piano di riparazione per ridurre rapidamente tali rischi. Esso permette a esperti di sicurezza di realizzare un processo proattivo di gestione delle vulnerabilità, che elimina le falle di sicurezza all'interno di una infrastruttura di rete, prima che la rete stessa venga penetrata e informazioni riservate siano compromesse.

The screenshot displays the Nexpose Community edition interface. At the top, there is a navigation menu with options: Home, Assets, Reports, Vulnerabilities, and Administration. The user is logged in as 'dookie'. The main content area shows the following sections:

- Device Properties:**
 - Addresses: 192.168.1.161
 - Hardware Address: C6:CE:4E:D9:C9:6E
 - Aliases: XEN-XP-SP2-BARE
 - Operating System: Microsoft Windows XP
 - CPE: cpe:/o:microsoft:windows_xp
 - Site: hotzone
- Vulnerability Listing:**

Vulnerability	Severity	Instances
Microsoft Server Service / CanonicalizePathName() Remote Code Execution Vulnerability	Critical	1
MS09-001: Vulnerabilities in SMB Could Allow Remote Code Execution	Critical	2
MS06-035: Vulnerability in Server Service Could Allow Remote Code Execution (917159)	Critical	1
Default or Guessable SNMP community names: private	Severe	1
Default or Guessable SNMP community names: public	Severe	1
CIFS NULL Session Permitted	Moderate	1
ICMP timestamp response	Moderate	1
- Policy Listing:** There are no policies to display.
- Installed Software Listing:** There is no software to display.
- Service Listing:** A table with columns: Service Name, Product, Port, Proto, Vulnerabilities, Users, Groups.

Figura 2.11: Risultati di una scansione effettuata con Nexpose.

Uno dei punti di forza, ricavabili dall'utilizzo di NeXpose, è la forte integrazione con i prodotti Metasploit, appunto sviluppati sempre dal team Rapid7. A questo, verrà dedicato approfonditamente tutto il prossimo capitolo.

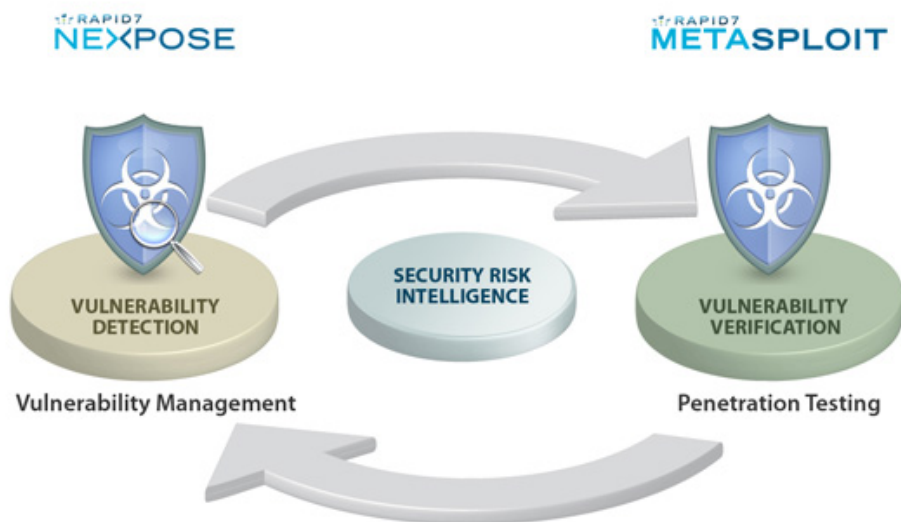


Figura 2.12: NeXpose e Metasploit sfruttano i dati forniti dalla comunità di Metasploit per identificare le vulnerabilità critiche di un ambiente.

Tale integrazione permette, infatti, di identificare le vulnerabilità su reti, sistemi operativi, database, applicazioni web e una vasta gamma di piattaforme di sistema attraverso un sistema integrato di ricerca. In questo modo, NeXpose riesce ad assegnare delle priorità alle vulnerabilità riscontrate, usando come criterio la probabilità che l'exploit venga sfruttato. I report, tenendo in considerazione queste valutazioni, cercano di ridurre l'esposizione delle proprie falle di sicurezza, indicando quale interventi quali segmenti dell'infrastruttura necessitano interventi prima di altri. Tutto al fine di ridurre i costi operativi, focalizzando dapprima l'attenzione degli amministratori sulle minacce ritenute più importanti.

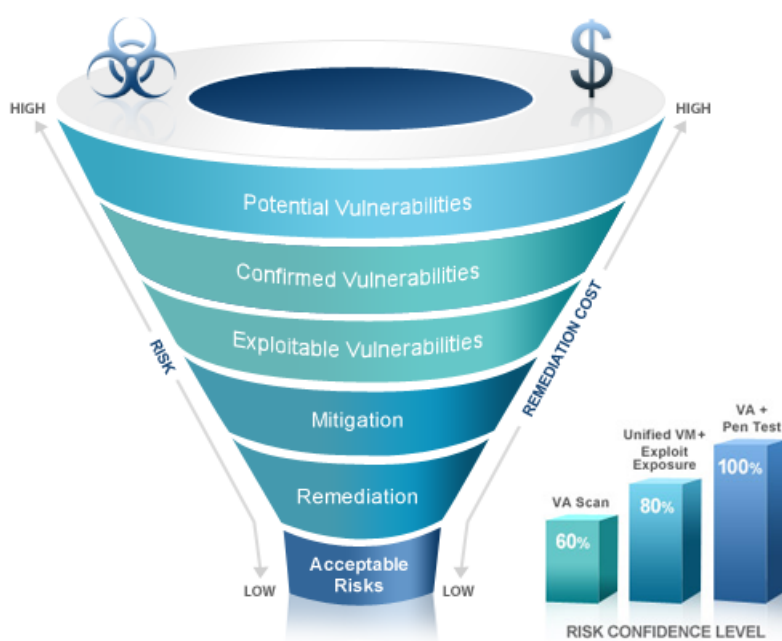


Figura 2.13: Valutazione dei rischi e assegnazione priorità.

Sono disponibili varie versioni di NeXpose, per cercare di soddisfare i vari bisogni della clientela. Si elencheranno brevemente le varie versioni attualmente disponibili, per poi lasciare spazio a una tabella comparativa che evidenzierà le differenze in termini di prezzo, caratteristiche e potenzialità.

- NeXpose *Community*: l'unica versione gratis, anche per uso commerciale.
- NeXpose *Express*: progettata per le piccole medie imprese, con un personale dedito all'IT limitato. Abbina un prezzo contenuto a dei requisiti hardware minimi che sono facilmente soddisfacibili.
- NeXpose *Consultant*: versione rivolta, per lo più, a consulenti indipendenti. Essa offre caratteristiche e benefici simili alla versione maggiore ma, allo stesso tempo, razionalizza la modalità di utilizzo.
- NeXpose *Enterprise*: essendo il prodotto con maggiore potenzialità, fornisce funzionalità progettate appositamente per essere sfruttate in ambienti IT molto complessi.

Risulta evidente che le prime due versioni sono rivolte ad appassionati o responsabili di sicurezza, in ogni caso in contesti aziendali medio piccoli. Mentre le altre due, sono progettate per le aziende di grandi dimensioni.

Con la seguente tabella, si cercherà di mostrare le caratteristiche fondamentali che differenziano le quattro versioni di NeXpose.

	NeXpose Community	NeXpose Express	NeXpose Consultant	NeXpose Enterprise
prezzo	gratis	3.000\$/ anno/utente	possibilità preventivo	possibilità preventivo
n. massimo di ip	32	256	illimitato	illimitato
n. massimo di utenti	1	1	un laptop	illimitato
massima ram supportata	4GB	8GB	illimitata	illimitata
assistenza clienti on-line	community	inclusa	inclusa	inclusa
assistenza clienti telefonica	non disponibile	opzionale	inclusa	inclusa
vulnerabilità di rete	si	si	si	si
vulnerabilità s.o.	si	si	si	si
vulnerabilità applicazioni	si	si	si	si
applicazioni web	no	opzionale	si	si
vulnerabilità database	si	si	si	si

Tabella 2.2: Comparativa tra le varie versioni di NeXpose.

Tutte le varie versioni prevedono, comunque, l'aggiornamento delle vulnerabilità e la perfetta integrazione con Metasploit.

2.4 Web vulnerability scanner

La stragrande maggioranza dei software esposti a internet è composta da applicazioni web. Al di là delle grandi organizzazioni, sulla cui sicurezza potrebbe essere possibile stare più o meno sereni, vi sono migliaia e migliaia di applicazioni web che spesso non raggiungono neanche un livello minimo di

sicurezza. Vediamo brevemente nel seguito, degli scanner riguardanti proprio le applicazioni web.

Le più comuni vulnerabilità nelle applicazioni web sono:

- **SQL injection:** se l'applicazione non filtra correttamente i moduli di input da parte gli utenti, e utilizza tali dati in una query SQL, è possibile dirottare la query per modificare il database o ottenere informazioni critiche, quali ID e password.
- **Cross-Site Scripting (XSS):** se l'input di un utente in un modulo web non è adeguatamente filtrato, è possibile iniettare codice in una pagina web, come ad esempio HTML o JavaScript. Questo permette a malintenzionati di iniettare codice dannose in una certa pagina, nei confronti della quale gli utenti nutrono una certa fiducia.
- **Inclusione di PHP:** un errore comune è quello di includere una pagina tramite una variabile URL. Il valore della variabile può essere cambiata per far sì che punti a una pagina remota che dovrebbe essere eseguita in locale.
- **Fuga di informazioni:** i file di configurazione e gli elenchi di utenti e password possono essere lasciati leggibili su un server web. A volte, è possibile ingannare un'applicazione web al fine di visualizzare file.
- **Privileges escalation:** alcune applicazioni scritte male possono permettere a chiunque di aumentare i propri privilegi privi attraverso variabili non documentate. Queste variabili nascoste spesso possono essere facilmente trovate e sfruttate da malintenzionati.

2.4.1 Nikto

Nikto è uno strumento di valutazione di web server. Si tratta di uno scanner open source, progettato per scovare non solo configurazioni e programmi su qualsiasi tipo di web server, ma anche file di default non sicuri. Può contare su un database di oltre 6.400 file potenzialmente pericolosi, controlli per le versioni obsolete di oltre 1.000 server e problemi specifici legati alla versione

su oltre 270 server. Nikto esamina un server web per trovare potenziali problemi e eventuali vulnerabilità di sicurezza, tra cui:

- errori di configurazione del server e dei vari software;
- file e programmi predefiniti;
- file e programmi non protetti;
- server e programmi obsoleti.

Nikto è basato su LibWhisker2 e può funzionare su qualsiasi piattaforma avente un ambiente Perl. Quindi è possibile eseguirlo correttamente su Windows, Mac OSX e varie distribuzioni Linux.

Supporta SSL, proxy, autenticazione dell'host ecc. Può essere aggiornato automaticamente da riga di comando.

2.4.2 HP WebInspect

HP WebInspect è uno scanner proprietario, inizialmente commercializzato con il nome di WebInspect da SPI Dynamics, azienda che è stata acquisita nel portafoglio gestionale IT dell'HP. La caratteristica principale è quella di eseguire test su server ad ogni livello. Come Nikto, infatti, controlla vulnerabilità note, per poi esaminare in maniera approfondita il sito web per analizzarne la struttura, tutti i file disponibili, i parametri utilizzati nell'URL e le form web. Tali informazioni vengono utilizzate da HP WebInspect per creare il traffico proveniente sia da vulnerabilità note che da altre tipologie di attacchi, per l'applicazione web testata.

Lo scanner è disponibile solamente per Windows e richiede un sistema con la seguente configurazione: deve essere presente la versione XP o superiore, il .NET Framework e Microsoft SQL Server Express.

Al momento di lanciare una scansione, un wizard ci guida a impostare le principali opzioni:

- URL: rappresenta l'indirizzo da scansionare. La porta da testare è, di default, la 80; se il server utilizza un'altra porta è possibile specificarla indicandola alla fine dell'URL.

- Limitazione della cartella: è possibile limitare la scansione a una cartella e/o relative sottocartelle.
- Metodo di valutazione: di default, le operazioni di scansione e auditing del sito web vengono effettuate allo stesso modo. È possibile selezionare una specifica opzione affinché, ogni qualvolta HP WebInspect trova una form, vengono richiesti i dati di accesso. Questo è particolarmente utile se si utilizza un modulo web per l'autenticazione e si vuole dare a HP WebInspect accesso ai contenuti privati del sito web in scansione.
- Policy: come per Nessus, è necessaria selezionare di una policy per poter lanciare la scansione.
- Autenticazione di rete: HP WebInspect gestisce quattro tipi di identificazione: HTTP Basic, NTLM, Digest e Kerberos. Si può rilevare automaticamente il tipo di autenticazione che viene utilizzato dal sito web.
- Proxy di rete: è possibile specificare un server proxy facoltativo da utilizzare per la connessione.

2.5 Vulnerability Exploitation

La fase successiva all'individuazione delle vulnerabilità è il loro sfruttamento. Le prossime sezioni vedranno l'analisi degli strumenti ritenuti migliori.

2.5.1 Metasploit Project

Il Metasploit Project è un progetto open-source, inerente la sicurezza informatica, il cui scopo è quello di fornire informazioni sulle vulnerabilità e semplificare le operazioni di penetration testing. Il progetto fornisce una suite di strumenti che possono essere sfruttati per ridurre significativamente le difficoltà che si incontrano nel processo di sviluppo e di lancio di un exploit. Comprende tre versioni: Metasploit Pro, Metasploit Express e Metasploit framework; solo quest'ultima è gratuita. Per la comprensione approfondita

di questi strumenti, e con particolare attenzione alle funzionalità del framework, si rimanda la lettura del terzo capitolo, interamente dedicato al Metasploit Project.

2.5.2 Core Impact

Core Impact Pro è una soluzione commerciale per il penetration testing, sviluppata da Core Security Technologies. Può contare su un portafoglio clienti vastissimo, nel quale spiccano molti nomi di multinazionali, nonché numerose agenzie governative; tutto a evidenziare l'assoluta qualità del prodotto. Core Impact consiste in una suite di programmi automatici, che permettono di rilevare e sfruttare vulnerabilità di sicurezza nelle reti informatiche, terminali, applicazioni web e reti wireless. Il suo più grande vantaggio è la sua facilità d'utilizzo. Esso consente agli amministratori di rete di eseguire tutto quell'insieme di attività base del penetration testing, senza la necessità di codificare linguaggio assembly o persino di compilare di compilare un exploit. All'interno della suite, è compreso anche un framework per il lancio di exploit. L'interfaccia grafica del framework non è così flessibile, come riscontrato in strumenti simili; tuttavia, nonostante ciò, una volta muniti di strumenti che soddisfino i giusti requisiti hardware richiesti, non si avranno problemi nel suo utilizzo.

Core Impact è basato su una tecnologia cosiddetta ad agente: prevede l'installazione, appunto, di un agente all'interno di un sistema compromesso. Questo assicura una penetrazione in profondità nella rete e un buon controllo del target sia per le operazioni in locale sia per quanto riguarda eventuali attacchi che possono essere lanciati dalla macchina compromessa. In altre parole, una volta che un sistema è stato compromesso durante un penetration test, Core Impact permette all'amministratore di rete di utilizzare tale sistema come un trampolino, dal quale lanciare ulteriori attacchi contro altri sistemi della stessa rete, replicando i tentativi di un utente malintenzionato di muoversi attraverso i sistemi vulnerabili per guadagnare livelli più profondi di accesso agli ambienti di rete.

Detto ciò, è possibile capire come i progettisti che sviluppano software di

questo tipo, cerchino di immedesimarsi in un attaccante, focalizzando la loro attenzione sugli aspetti più vulnerabili:

- sfruttare eventuali vulnerabilità dei sistemi di rete, applicazioni web, sistemi lato client, reti wireless e altri dispositivi di rete.
- elevare il proprio privilegio a quello di root (amministratore); normalmente definito come "privilege escalation".
- prevenire le conseguenze di un attacco in termini di accesso, furto, manipolazione di dati da parte di un utente malintenzionato nei sistemi compromessi.

Tramite queste ed altre strategie, Core Impact Pro permette di effettuare valutazioni sulla sicurezza, testando più ambienti possibili, anche tra diversi strati della tecnologia.

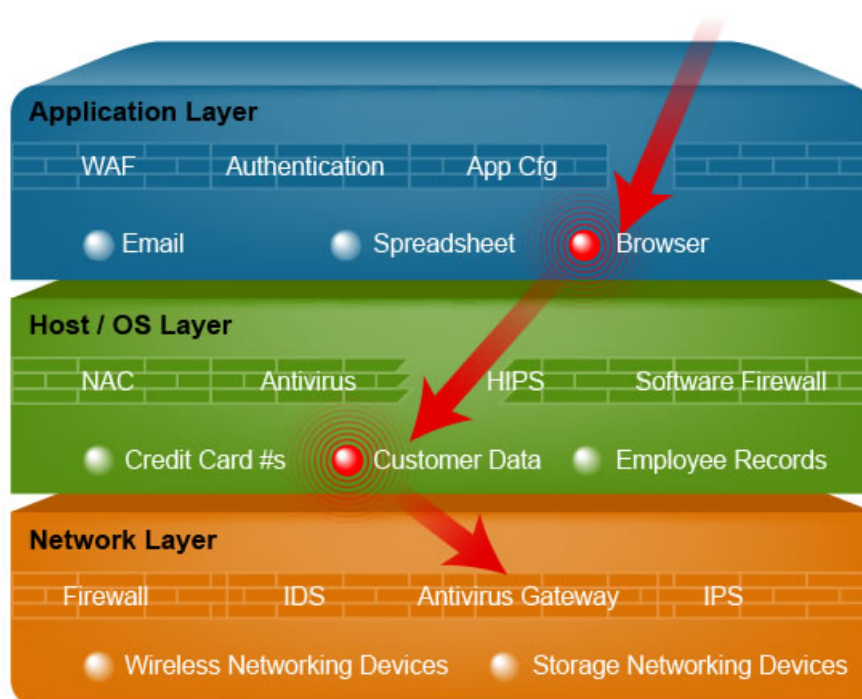


Figura 2.14: Esempio di valutazione di sicurezza tra diversi strati.

Una metodologia alla base del funzionamento di Core Impact è il RPT: *Rapid Penetration Testing*, che semplifica il testing di server, sistemi desktop,

terminali, applicazioni web, reti wireless e dispositivi di rete, automatizzando le attività che tradizionalmente richiederebbero molto tempo, impegno e competenza nell'esecuzione. Il RPT riesce quindi ad automatizzare la fase di penetration testing attraverso fasi fondamentali, come sarà possibile osservare nelle prossime immagini. Si tratta di un metodo di testing semplice e basato su un'interfaccia a wizard, anche se in certi casi, con il suo traffico, quasi satura la rete. Questo è un problema noto, ma d'altronde, è normale in questa fase. Core Impact fornisce le funzionalità RPT in cinque categorie di testing:

- Network RPT: replica i comportamenti di un utente malintenzionato nel lanciare exploit remoti sulla rete.
- Client-Side RPT: scansione delle vulnerabilità legate al phishing, spam e altri attacchi di ingegneria sociale² contro gli utenti finali.
- Web Application RPT: attacchi del tipo SQL injection o relativi ad un inserimento remoto di file contro l'e-commerce, servizio clienti, ERP e altre applicazioni web.
- Wireless Network RPT: vulnerabilità legate a tentativi di scoprire access point Wi-Fi, il cracking delle chiavi di crittografie e l'accesso a reti esposte.
- Network Device RPT: tentativi di accedere a reti e intercettare dati, individuando e sfruttando le vulnerabilità di router e switch.

Gli approcci ai cinque differenti test si differenziano nella fasi di Information Gathering e Attack and Penetration.

²Con il termine ingegneria sociale si intende lo studio del comportamento umano al fine di carpirne informazioni. Cerca di conquistarsi la fiducia del soggetto al fine di manipolarla, aggirando così quello che il buon senso suggerirebbe.

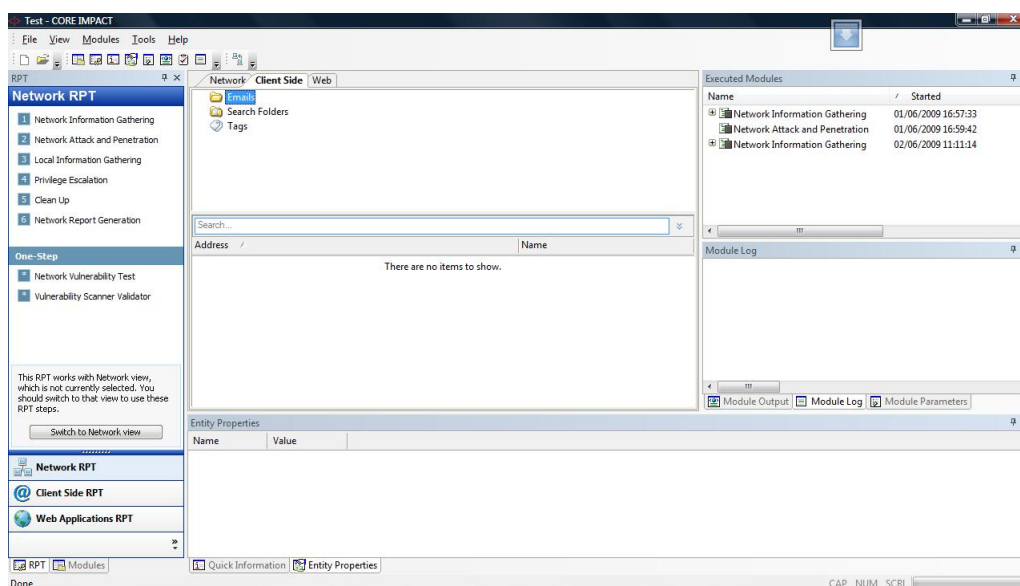


Figura 2.15: Schermata per il Network RPT.

Nonostante il RPT è un metodo semplice per effettuare una valutazione della rete, non può certo sostituire un esperto *penetration tester*. RPT fornisce una veloce dimostrazione delle potenzialità offerte da Core Impact e la sua facilità d'uso, ma test di questo tipo non dovrebbero essere utilizzati nella fase vera e proprio di penetration testing; fase nella quale l'obiettivo rimane quello di lasciare meno minacce possibili. In altre parole, una volta eseguito un certo modulo RPT non si riscontrano delle minacce, non ci si può sentire eccessivamente sicuri, in quanto il risultato ottenuto non significa necessariamente che la rete sia sicura. In certi ambiti è sempre meglio affidare strumenti così potenti, a mani altrettanto esperte, onde evitare valutazioni troppo ottimistiche o problemi, invece che soluzioni, alla rete.

2.5.3 Canvas

Canvas è uno strumento di valutazione della sicurezza, sviluppato dalla Immunity, progettato più come uno strumento di sviluppo di exploit e test di difesa, piuttosto che come un tool completo per il penetration testing, come Core Impact. La forza di Canvas risiede nel fornire uno dei framework più

flessibili e potenti per lo sfruttamento di vulnerabilità e per il rilevamento delle intrusioni in un dispositivo. Canvas è rilasciato con una licenza commerciale, ma supporta anche una licenza per un singolo utente che prevede: un quarto degli aggiornamenti mensili standard e del supporto, nessuna limitazione sull'indirizzo IP della macchina target e il rilascio completo del codice sorgente.

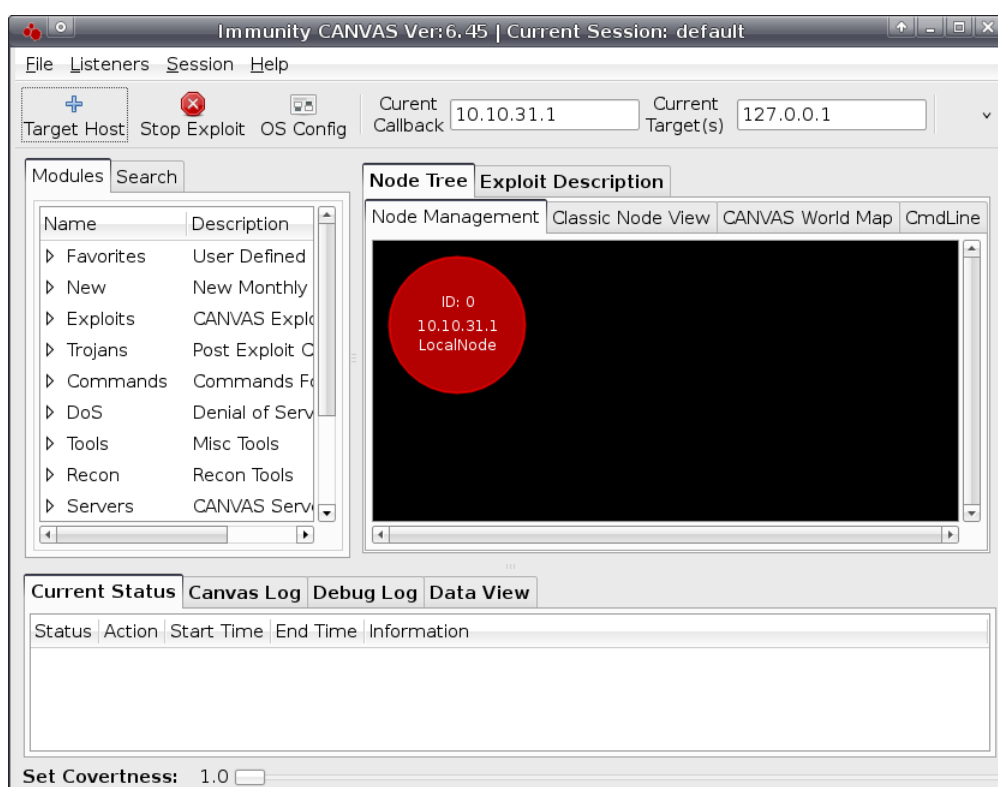


Figura 2.16: La schermata di default dell'interfaccia grafica di Canvas.

Per quanto concerne le piattaforme supportate, il software è disponibile per i seguenti sistemi operativi: Windows (richiede l'installazione di Python e PyGTK), Linux e MacOSX (richiede PyGTK). Sono inoltre supportati tutti gli altri ambienti Python come ad esempio telefoni cellulari. All'interno di Canvas sono attualmente compresi oltre 370 exploit che vengono aggiornati tramite un rilascio, mediamente, di 4 nuovi exploit al mese. Exploit che vengono rilasciati via web, non appena risultino stabili, e che interessano tutte le più comuni piattaforme e applicazioni. Immunity seleziona attentamente

le vulnerabilità da includere nel database Canvas, dando una priorità più alta alle vulnerabilità maggiormente critiche, come quelle remote o quelle riguardanti software utilizzati su vasta scala. Il supporto comprende anche la possibilità di contattare via posta elettronica il team di sviluppo di un certo exploit, il quale, in ogni caso, realizza filmati flash che dimostrano lo sfruttamento dell'exploit.

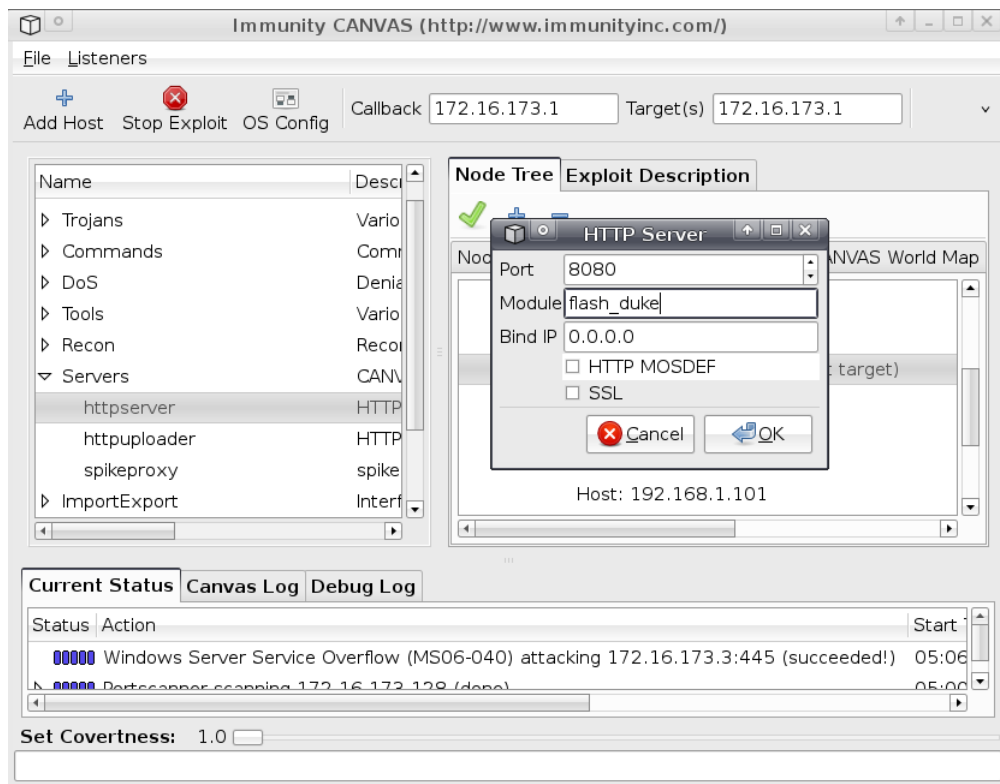


Figura 2.17: Avvio di exploit lato client tramite il server HTTP compreso.

L'interfaccia grafica di Canvas, permette di eseguire la maggior parte delle attività di penetration testing. L'area è divisa in più riquadri che forniscono informazioni e consentono di eseguire diversi compiti. I report generati dall'applicazione raccolgono tutte le informazioni acquisite durante il test. Inoltre, è possibile installare dei plugin di terze parti per estendere le funzionalità standard di Canvas.

2.5.4 W3af

W3af è il diminutivo per "Web Application Attack and Audit Framework", rilasciato con licenza GPL 2 e sviluppato in python. Come suggerisce il nome, *w3af* è sia uno scanner sia un framework per il lancio di exploit, entrambi nei confronti di applicazioni web. Consiste quindi in un ambiente completo per la verifica e l'attacco di applicazioni web.

W3af ha diverse tipologie di plugin, ecco le tipologie fondamentali:

- *discovery*: hanno il compito di trovare nuovi URL, form e altri cosiddetti "injection point". Un classico esempio di un plugin di discovery è un "web spider": richiede un URL come input restituendo uno o più punti di iniezione. Quando un utente abilita più di un plugin di questo tipo, essi lavorano in un ciclo: se un plugin trova un nuovo URL nel primo ciclo, *w3af* invierà tale URL al successivo plugin. Se poi anche quest'ultimo trova un nuovo URL, sarà inviato al primo plugin. Questo processo andrà avanti fino a quando tutti i plugin sono stati eseguiti e nessun'altra informazione riguardo l'applicazione web scansionata può essere trovata, tramite i plugin di discovery abilitati precedentemente.
- *evasion*: vengono usati per provare a eludere l'eventuale sorveglianza di un IDS (Intrusion Detection System).
- *audit*: una volta che i plugin di discovery hanno individuato tutti i punti di iniezione, i plugin di audit inviano dati appositamente predisposti a tutti loro, al fine di trovare vulnerabilità. Un classico esempio di un plugin audit è quello che cerca vulnerabilità di tipo SQL injection.
- *grep*: sono utilizzati per analizzare ogni risposta che il server restituisce, in cerca di qualcosa di interessante; non importa quale plugin abbia avviato la richiesta.
- *attack*: il suo obiettivo è quello di sfruttare le vulnerabilità trovate dai plugin di audit. Di solito restituisce una shell sul server remoto, o un dump di tabelle remote in caso di un exploit con un SQL injection.

- *output*: usati per scrivere l'output degli altri plugin e del framework stesso in un formato pratico come txt o html.

W3af mette a disposizione dell'utente due interfacce grafiche, quella a console *consoleUI*, e una GUI *gtkUi*. Al momento attuale, la console è molto più testata e completa rispetto all'altra.

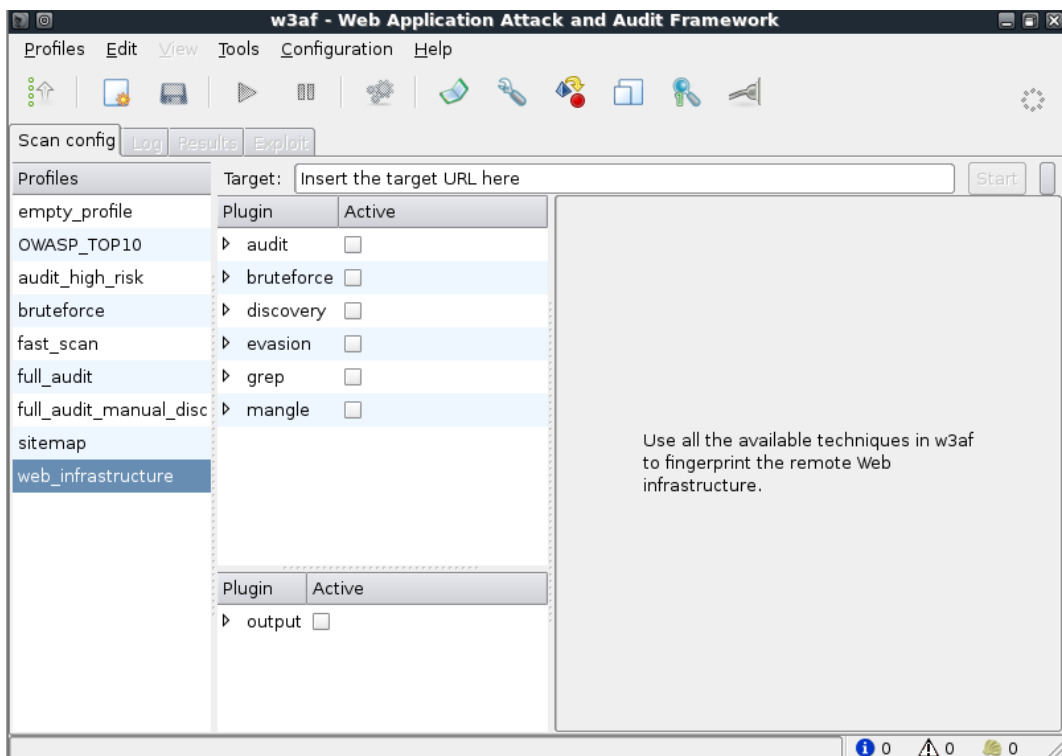


Figura 2.18: La gui di w3af.

Capitolo 3

Metasploit Project

3.1 Panoramica

Il Metasploit Project, come accennato nel precedente capitolo, è un progetto riguardo la sicurezza informatica, il cui scopo è quello di fornire quante più informazioni possibili sulle vulnerabilità e semplificare le operazioni di penetration testing. Il progetto fornisce una suite di strumenti che possono essere sfruttati per ridurre significativamente le difficoltà che si incontrano nel processo di sviluppo e di lancio di un exploit.

Metasploit è stato creato da HD Moore nel 2003 come uno strumento di rete portatile, scritto quasi interamente in Perl. L'obiettivo era proprio quello di diventare una risorsa pubblica per valorizzare la ricerca o lo sviluppo di codice. Con il passare degli anni, ha acquisito sempre più una rinomata fama a livello mondiale, fino al divenire uno degli strumenti più utilizzati nel settore della sicurezza informatica. La piattaforma si rivolge sia agli esperti generici di sicurezza con la necessità di testare alcune semplici vulnerabilità, sia ai penetration tester con esigenze molto più avanzate.

Vediamo le caratteristiche generali del Metasploit Project:

Come già accennato, Metasploit è nato principalmente per fornire un framework ai penetration tester per sviluppare exploit. Il tipico ciclo di vita di una vulnerabilità e del suo sfruttamento è il seguente:

1. Scoperta. Un ricercatore o uno sviluppatore scopre una vulnerabilità

critica di sicurezza nel software.

2. **Divulgazione.** Il ricercatore può comunicare la sua scoperta in due modi: aderendo a una divulgazione ufficiale o comunicandola su una mailing list pubblica. In entrambi i casi, ovviamente, il venditore ha la necessità di trovare una patch per la vulnerabilità scoperta.
3. **Analisi.** Il ricercatore o altri soggetti sparsi in tutto il mondo iniziano l'analisi della vulnerabilità per determinare la sua sfruttabilità o meno. Nel caso in cui sia sfruttabile, si studia la possibilità di eseguirla da remoto. Questa fase coinvolge anche il debug dell'applicazione vulnerabile: nella porzione vulnerabile di codice viene iniettato un input dannoso.
4. **Sviluppo.** Una volta trovate le risposte alle domande chiave, il processo di sviluppo dell'exploit può avere inizio. Questa fase potrebbe essere considerata, per alcuni, come una sorta di magia nera, in quanto è richiesta una conoscenza approfondita dei registri del processore, codice assembly, offset e payload.
5. **Test.** Questa è la fase in cui il programmatore testa il codice dell'exploit contro diverse piattaforme, service pack o patch, e possibilmente anche per diversi processori (ad esempio Intel, Sparc e così via).
6. **Rilascio.** Dopo che l'exploit è stato positivamente testato, ed i parametri specifici necessari per una sua esecuzione a buon fine sono stati determinati, lo sviluppatore rilascia l'exploit, privatamente o su un forum pubblico. Spesso, l'exploit è ottimizzato in modo che non funzioni al di fuori della sandbox¹. Questo è solitamente fatto per dissuadere "simpatici soggetti perditempo" che una volta scaricato l'exploit, avrebbero l'intenzione di lanciarlo contro un sistema vulnerabile.

In generale ecco le caratteristiche che concretizzano le vaste potenzialità di Metasploit Project:

¹Il sandbox è un meccanismo di sicurezza per separare diversi programmi in esecuzione. Questo è spesso usato per eseguire codice non testato o, comunque, programmi non attendibili provenienti da fonti incerte.

- ampio database pubblico di exploit testati: Metasploit sfrutta una vasta collezione pubblica di exploit e payload² garantiti, rendendo i penetration test realistici nel simulare attacchi verso un sistema informatico.
- l'interfaccia grafica disponibile rende il flusso di lavoro semplificato.
- gli obiettivi del penetration test possono essere: le applicazioni Web standard e personalizzate, dispositivi di rete, server di database, sistemi di endpoint e gli utenti di posta elettronica, ampliando così la gamma di vettori del test.
- Metasploit comprende funzionalità avanzate di attacco: incorpora altri strumenti che consentono di lanciare flussi di lavoro riguardo la scansione di rete, attacchi di tipo bruteforce³, di ingegneria sociale consentendo di muoversi più in profondità nella rete.

Il progetto è composto da tre versioni: Metasploit Framework, Metasploit Express e Metasploit Pro; solo la prima è gratuita. Vediamo subito nella seguente tabella le caratteristiche che differenziano le varie versioni, per poi analizzare meglio alcuni aspetti.

²Qualsiasi stream di byte che viene trasmesso su una rete, comprende sia i dati sia le informazioni che identificano la sua origine e la sua destinazione. Il payload consiste nel dato reale contenuto nelle intestazioni del pacchetto o frame. Con riferimento a uno exploit, il payload rappresenta le conseguenze generate da un virus o altro codice malevolo eseguito dall'exploit sulla macchina target. Il payload di un virus può causare lo spostamento, la modifica o la cancellazione di determinati file oppure altre attività non desiderate.

³Il termine bruteforce indica un attacco "forza bruta", a volte indicato anche come "metodo della ricerca esaustiva della soluzione". È una strategia che consente di verificare sistematicamente tutte le chiavi teoricamente possibili fino a quando la chiave corretta non è stata individuata. Nel peggiore dei casi, si tratterebbe di attraversare l'intero spazio di ricerca.

	Metasploit Framework	Metasploit Express	Metasploit Pro
progettato per	comunità open source	team di sicurezza IT	consulenti di penetration testing
prezzo	gratis	3.000 \$	15.000 \$
ultimi aggiornamenti	si	si	si
raccolta automatica informazioni rete	no	si	si
raggiro di IDS, IPS, anti-virus	no	no	si
target dell'attacco server, desktop, web server, database, periferiche	si	si	si
scoperta, controllo e sfruttamento vulnerabilità di applicazioni web	no	no	si
campagne lato client di ingegneria sociale	no	no	si
importazione report XML di vulnerability scanner	si	si	si
collaborazione in team per simulazioni di attacchi combinati	no	no	si
report esportabili (HTML, PDF, XML)	no	si	si
interfaccia a console	si	no	no
console avanzata	no	no	si
integrazione con il gestore della vulnerabilità NeXpose	si	si	si
ricerca globale in tutti i progetti e gli host	no	si	si
supporto comunità	si	si	si
supporto clineti specifico	no	si	si

Tabella 3.2: Comparativa tra le tre versioni di Metasploit.

Una breve osservazione della tabella fa realizzare che le caratteristiche fondamentali del progetto sono garantite per tutta la suite di prodotti. Chiaramente si vede come il Metasploit Framework sia uno strumento versatile con enormi potenzialità. Tuttavia, risulta altrettanto evidente come, in ambiti aziendali o per chi fa del penetration testing la propria professione, ci si dovrà rivolgere alle versioni a pagamento.

3.2 Metasploit Pro

Metasploit Pro è un software a livello aziendale, rivolto ai professionisti della sicurezza che si specializzano in test di penetrazione e richiedono una soluzione avanzata per attacchi multi-livello che consentono di analizzare la rete in maniera molto curata ed efficiente. Esso consente agli utenti di indentificare rapidamente, valutare e sfruttare le applicazioni web vulnerabili. Utilizzando il pivoting per le VPN, è possibile lanciare il vulnerability scanner NeXpose attraverso il server web compromesso per scoprire una vulnerabilità sfruttabile in un certo database, che ospita i dati dei clienti o informazioni riservate ai dipendenti dell'azienda. I membri del team possono quindi sfruttare i dati acquisiti per fare dell'ingegneria sociale sotto forma di una campagna di phishing, aprendo nuovi vettori di attacco sulla rete interna. Metasploit Pro permette poi di personalizzare i report da esportare.

I requisiti minimi di sistema per questa versione sono: processore 2 GHz+, 2GB RAM disponibile, 500MB+ di spazio disponibile sul disco fisso. I sistemi operativi supportati in maniera ufficiale sono: l'intera famiglia Windows, Red Hat e Ubuntu, sia a 32 che a 64 bit.

3.3 Metasploit Express

Metasploit Express è un software ottimizzato per gli esperti di sicurezza con una vasta gamma di responsabilità che necessitano di una accessibile soluzione per eseguire penetration testing e verificare i risultati di uno vulnerability scanner, senza avere una formazione approfondita. Esso si basa sulle po-

tenzialità di Metasploit Framework, ma aggiungendo l'automatizzazione di molte attività comuni durante un penetration test e offrendo la possibilità di lanciare attacchi avanzati, senza comunque avere l'onere di sviluppare script personalizzati. Tramite Metasploit Express è possibile fornire al team che si occupa della sicurezza, una solida base per poter rendere sicura l'infrastruttura IT di una organizzazione.

I requisiti minimi di sistema sono gli stessi della versione Pro, come anche i sistemi operativi ufficialmente supportati.

3.4 Metasploit Framework

Metasploit Framework è una piattaforma di sviluppo open source per la creazione di strumenti di sicurezza e, in particolar modo, di exploit. Il framework viene utilizzato da un vasto numero di figure, in ambito lavorativo o accademico. In generale, Metasploit Framework viene usato dai professionisti della sicurezza di rete per eseguire test di penetrazione, dagli amministratori di sistema per verificare le installazioni di patch, dai fornitori di prodotti per effettuare test di regressione⁴ e da ricercatori di sicurezza in tutto il mondo. Il framework è scritto nel linguaggio di programmazione Ruby, ma include anche componenti scritti in C, Java e assembler.

Per installare Metasploit Framework, d'ora in avanti MSF, è necessario avere una macchina con una installazione di Ruby funzionante. Questo è un altro grosso vantaggio del framework in quanto è così installabile su tutte le seguenti piattaforme: tutta la famiglia Linux, quella Windows, Mac OS X, iPhone, Android e Maemo. Per maggiori dettagli sulla processo di installazione e configurazione si rimanda il lettore ad una successiva sezione, appositamente dedicata a descrivere i vari passaggi per avere un'installazione funzionante di MSF.

⁴dopo aver introdotto dei cambiamenti in un certo prodotto, il fornitore esegue nuovamente dei test anche sulle vecchie funzionalità per assicurare che la qualità complessiva del prodotto non sia compromessa.

3.4.1 La struttura

Come mostrato nella seguente figura, gli elementi base dell'architettura del MSF sono le librerie, le interfacce, i moduli e i plugin.

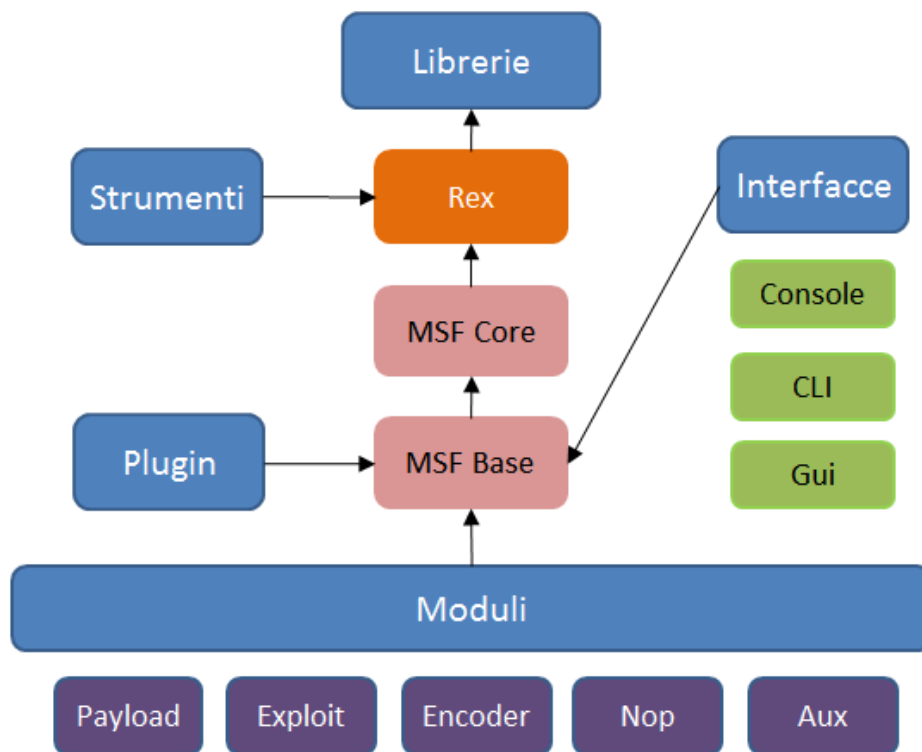


Figura 3.1: Architettura del Metasploit Framework.

Una documentazione completa di tutte le classi e le API si possono trovare nella documentazione presente sul sito web di Metasploit. In ogni caso, analizziamo brevemente i componenti. Il file system di MSF è disposto in maniera intuitiva ed è organizzato in directory:

- *lib*: in questa cartella è raccolto il codice fondamentale che fa funzionare il MSF.
- *data*: file modificabili utilizzati da MSF.
- *tools*: vari strumenti utilizzabili da riga di comando.

- *modules*: i moduli effettivi di MSF.
- *plugins*: plugin che possono essere caricati al momento dell'esecuzione.
- *scripts*: Meterpreter e gli altri script.
- *external*: codice sorgente e librerie di terze parti.

REX è il componente più importante dell'intera architettura del framework; il suo nome sta per "Ruby Extension Library". Esso è essenzialmente un insieme di classi e moduli che possono essere utilizzate dagli sviluppatori per sviluppare progetti o strumenti riguardo MSF.

MSF Core rappresenta il nucleo del framework ed è costituito di vari sottosistemi, come la gestione dei moduli, delle sessioni e degli eventi. Questa parte fornisce anche un'interfaccia tra i moduli e i plugin e il framework stesso. Seguendo l'approccio orientato agli oggetti di tutta l'architettura, anche il framework in sé può essere considerato una classe, la quale può essere istanziata e venire usata come qualsiasi altro oggetto. MSF Core si compone di:

- *Datastore*: si tratta di un hash dei valori che possono essere utilizzati sia da moduli riferiti dal programmatore sia da valori impostati dall'utente. Le variabili d'ambiente sono una categoria di tali valori, le quali vengono utilizzate sia dai moduli di un exploit sia dal framework per determinare il comportamento atteso.
- *Event Notifications*: il MSF consente agli sviluppatori di reagire a eventi specifici del framework e di poter eseguire azioni arbitrarie di conseguenza. Questo funziona con la stessa logica di come vengono gestiti gli eventi in Windows, e richiede ad ogni istanza del framework di avere un collegamento con il gestore di tali eventi. Alcuni degli eventi che possono operare sono quelli legati al framework in generale, ad un exploit, a eventi di riconoscimento (ad esempio quando un nuovo host o servizio viene scoperto) e infine a quelli di sessione.

- Framework Managers: come accennato in precedenza, il framework è composto da sottosistemi critici, responsabili della gestione dei moduli, plugin, sistemi di riconoscimento, le sessioni e altre attività.

MSF Base è implementato al di sopra del MSF Core e fornisce le interfacce per poter interagire con il nucleo. Alcune di queste sono:

- Configuration: mantenere una configurazione persistente e ottenere informazioni sulla struttura di un impianto, come ad esempio la directory root di installazione e altri attributi.
- Logging: supporto di autenticazione.
- Sessions: il nucleo mantiene informazioni e controlla il comportamento delle sessioni dell'utente.

Il framework fornisce anche classi e metodi per semplificare le interazioni con esso, come ad esempio quando si tratta di exploit, NOP, payload e moduli di riconoscimento.

I *moduli* del framework sono:

- Exploit: il motivo dell'esistenza del framework. Tutti gli exploit contenuti nel MSF sono raggruppati in due categorie: attivi e passivi. Gli exploit attivi sfrutteranno vulnerabilità in uno specifico host, fino al loro completamento, e poi usciranno. Gli exploit passivi, invece, restano in attesa di qualche host per poi sfruttarli come si connettono.
- Payload: se il test dell'exploit ha successo, si dispone di una vasta gamma di operazioni che è possibile eseguire sul sistema target. Queste attività includono l'aggiunta di un utente, l'esecuzione di un comando specifico, la generazione di una shell, l'iniezione di DLL VNC per un accesso da remoto con interfaccia grafica e moltissimo altro ancora.
- NOP generator: spesso, la posizione esatta del salto non può essere conosciuta e i NOP necessitano di essere preposti all'exploit in fase di test. Per evitare il riconoscimento da parte di un IDS bisogna attivare l'offuscamento delle sequenze NOP.

- Encoder: come per i NOP, anche l'esecuzione di payload potrebbe essere rilevata da un IDS. Questo può essere evitato dalla codifica dei payload in modo tale da passare inosservati nella rete per poi venire decodificati una volta oramai giunti al bersaglio.
- Auxiliary module: un'aggiunta molto importante dalla release 3.0 sono stati i moduli ausiliari, che forniscono migliori funzionalità al penetration tester in termini di impronte digitali e scansione delle vulnerabilità. Per esempio, uno dei moduli ausiliari permette la connessione a un MS SQL Server, mentre un altro modulo tenta di indovinare la versione e il service pack del sistema operativo Windows in base al comportamento del protocollo SMB.

Per quanto riguarda i *plugin*, sono una componente introdotta dalla versione 3.0. Rispetto ai moduli i plugin sono progettati per modificare il framework stesso. Per esempio, un plugin può essere sviluppato per aggiungere un nuovo comando all'interfaccia console. Mentre plugin avanzati possono avere la capacità di automatizzare alcune sequenze di attività frequenti. Tutto dipende dalla creatività dello sviluppatore.

Meterpreter

Quando si tenta di sfruttare un sistema remoto, un utente malintenzionato ha un obiettivo specifico in mente, di solito quello di ottenere la shell dei comandi, e quindi di poter eseguire comandi arbitrari su questo sistema target. L'attaccante vorrebbe farlo nella maniera più furtiva possibile, certamente sfuggendo ad eventuali IDS (Intrusion Detection Systems). Se l'exploit va poi effettivamente a buon fine, ma la shell dei comandi non funziona o non ha privilegi di root, le opzioni dell'attaccante sarebbero gravemente limitate. Questo comporterebbe il lancio di un nuovo processo sul sistema remoto, che porterebbe ad una situazione ad alta visibilità, nella quale un buon amministratore o un analista forense noterebbe con facilità il processo sospetto. Inoltre, l'attaccante ha solitamente una sola opportunità di lanciare una shell o comunque di eseguire un comando arbitrario.

È proprio in questo contesto che si inserisce Meterpreter, abbreviazione di Meta-Interpreter. Meterpreter è uno dei più avanzati payload disponibili in MSF. Tuttavia, il modo di analizzare quello che Meterpreter effettivamente riesce a fare, non può limitarsi ad una sua visione come un semplice payload; piuttosto come una piattaforma di exploit che viene eseguita sul sistema. Meterpreter ha la sua propria shell, consentendo l'attaccante l'esecuzione di una vasta gamma di attività che possono poi essere eseguite sulla macchina bucata. Inoltre, Meterpreter consente agli sviluppatori di scrivere le proprie estensioni sotto forma di file DLL, per una successiva iniezione. Pertanto, qualsiasi linguaggio di programmazione in cui i programmi possono essere compilati in DLL possono essere utilizzati per sviluppare estensioni per Meterpreter. Ma la vera potenzialità di Meterpreter è che funziona iniettandosi nel processo vulnerabile in esecuzione sul sistema remoto, una volta che sia stato compromesso. Tutti i comandi lanciati tramite Meterpreter, vengono eseguiti nel contesto del processo in esecuzione. In questo modo, si è in grado di evitare il rilevamento da parte dei sistemi antivirus basilari e esami forensi non approfonditi. Un esperto forense, infatti, avrebbe bisogno di effettuare un'analisi diretta della memoria dei processi in esecuzione, al fine di poter determinare il processo di iniezione. E anche questo sarebbe tutt'altro che semplice.

3.4.2 Le interfacce

Esistono diverse interfacce di MSF, ciascuna con propri punti di forza e di debolezza. In quanto tale, non esiste una interfaccia perfetta per l'utilizzo con MSF, anche se la console è un ottimo strumento per chi lo usa quotidianamente. In ogni caso è consigliabile che ogni utente utilizzi l'interfaccia con la quale si trovi a proprio agio. Finita l'analisi delle interfacce, analizzeremo brevemente un'altra GUI, rilasciata a fine 2010, che è risultata molto versatile, "Armitage".

console di MSF. Ciò aumenta notevolmente la capacità di interagire con il framework, offrendo anche delle funzionalità per aiutare l'utente ad eseguire il debug dello script.

- *jobs*. Visualizza e gestisce tutte le attività inerenti al framework. Una delle novità introdotte con la versione 3 di MSF è la possibilità di pianificare le attività direttamente dall'interfaccia a console. Questo comando permette, oltre ad elencare le varie attività, anche la facoltà di eliminarle.
- *loadpath*. Aggiunge uno o più percorsi di ricerca del modulo. L'utente può utilizzare anche moduli che non sono localizzati nelle directory standard di MSF.
- *route*. Questo comando permette di instradare il traffico attraverso una sessione: indirizza il traffico per una determinata subnet attraverso una sessione il cui ID è conosciuto.

Msfcli

L'interfaccia *msfcli* ("MetaSploit Framework Command Line Interface") permette di eseguire l'esecuzione di exploit direttamente dalla linea di comando Unix o Windows, senza la necessità di aver precedentemente lanciato *msfconsole*. L'inconveniente di questa interfaccia è rappresentato dal fatto che non è ben supportato come *msfconsole* e che, soprattutto, può gestire solamente una shell alla volta, il che rende *msfcli* inutilizzabile per gli attacchi sul lato client. Non supporta, inoltre, nessuna delle caratteristiche avanzate di automazione di *msfconsole*. Nonostante questi svantaggi, *msfcli* può rivelarsi molto utile per certe attività specifiche e per apprendere la fase di configurazione di un exploit, prima di lanciarne l'esecuzione.

Questo tipo di interfaccia si addice maggiormente a circostanze in cui si ha la necessità di lanciare rapidamente un exploit specificando direttamente tutti i parametri richiesti come argomenti della riga di comando. È inoltre particolarmente utile quando un gran numero di sistemi devono essere testati per la medesima vulnerabilità. In quel caso può essere scritto un semplice

script di shell, che scorra un intervallo di indirizzi IP utilizzando msfcli per eseguire exploit contro ognuno dei sistemi target.

Come di consueto l'opzione "-h" ci permette di visualizzare tutte le opzioni disponibili.

```

root@bt:~# msfcli -h
Usage: /usr/local/bin/msfcli <exploit_name> <option=value> [mode]
=====
Mode           Description
----           -
(H)elp         You're looking at it baby!
(S)ummary     Show information about this module
(O)ptions     Show available options for this module
(A)dvanced    Show available advanced options for this module
(I)DS Evasion Show available ids evasion options for this module
(P)ayloads    Show available payloads for this module
(T)argets     Show available targets for this exploit module
(A)ctions     Show available actions for this auxiliary module
(C)heck       Run the check routine of the selected module
(E)xecute     Execute the selected module

```

Figura 3.3: Una interfaccia di MSF: msfcli.

In sostanza, useremo il seguente esempio per illustrare il modo più semplice per eseguire un exploit utilizzando l'interfaccia msfcli:

- informazioni su un exploit selezionato:

```
./msfcli <exploit_name> S
```

- mostra payload disponibili:

```
./msfcli <exploit_name> P
```

- sceglie il payload con questo exploit e visualizza le opzioni che devono essere impostati:

```
./msfcli <exploit_name> PAYLOAD = <payload_name> O
```

- elenco target disponibili:

```
./msfcli <exploit_name> PAYLOAD=<payload_name> T
```

- impostare le opzioni richieste in option=value form and execute with the E mode.

Mfsgui

Questa interfaccia è una Gui scritta nel linguaggio Java. Rappresenta un buon strumento per chi non vuole proprio saperne di utilizzare la console.

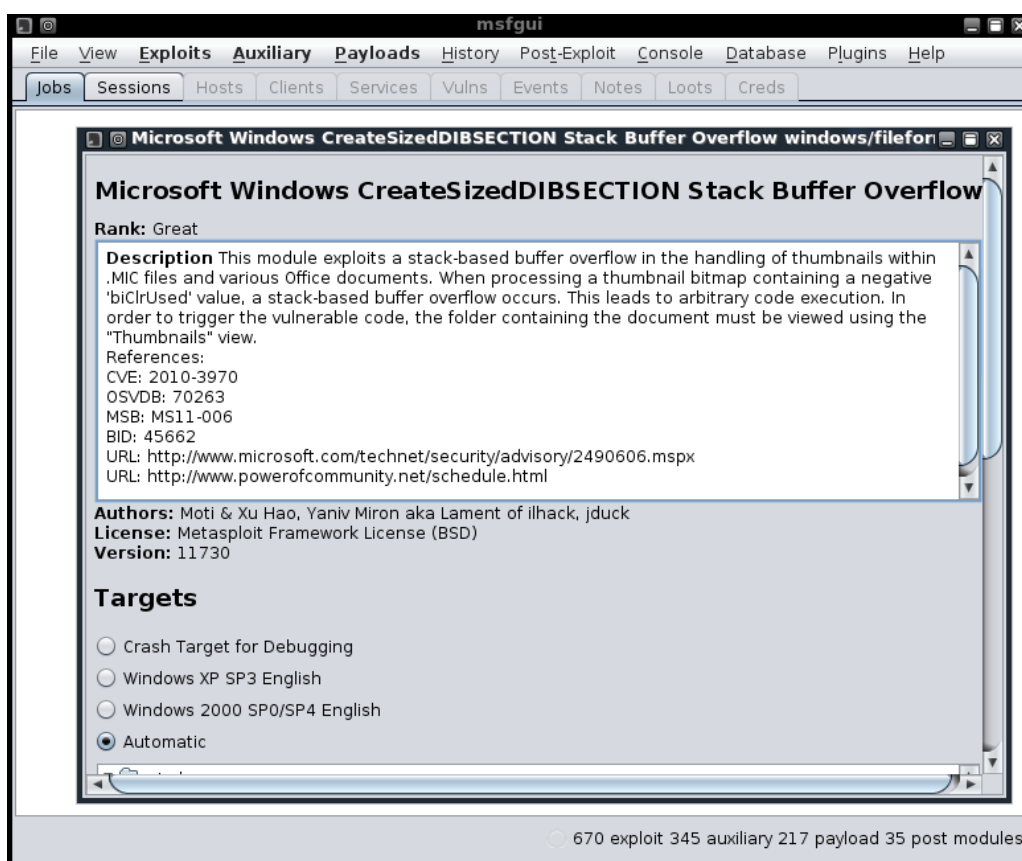


Figura 3.4: La Gui in java del Metasploit Framework.

La Fig. 3.4 mostra una schermata di mfsgui con un exploit selezionato, in attesa che vengano ulteriormente specificate altre opzioni. In alto, per mezzo della barra dei menu, possiamo visualizzare tutti gli exploit, auxiliary e payload disponibili con una certa versione di MSF. Ma è anche possibile accedere a tutte le altre funzioni riguardo le azioni da eseguire dopo l'esecuzione di un exploit andato a buon fine o, piuttosto, la gestione dei plugin.

All'interno del pannello di configurazione dell'exploit, vi sono campi richiesti e altri non. Prima di lanciare l'esecuzione di un determinato exploit, è necessario impostare i parametri richiesti.

3.4.3 Armitage

Armitage è un strumento grafico di gestione di attacchi per MSF in grado di visualizzare i vari target e raccomandare exploit, mettendo in evidenza anche le funzionalità avanzate del framework. Questa Gui è stata sviluppata da Raphael Mudge, per il team "Fast and Easy Hacking", verso la fine del 2010, ed è rilasciata sotto licenza BSD. Armitage rende sicuramente più facile l'interazione tra l'utente alle prime armi e MSF; ciò nonostante, anche gli utenti più esperti potranno apprezzarlo per quanto riguarda funzionalità come la gestione remota delle istanze del framework e la possibilità di collaborazione tra più team.

Armitage organizza le funzionalità di MSF riguardo l'intero processo di hacking possibile. Ci sono infatti caratteristiche per la scoperta, l'accesso e il post-sfruttamento. Per la fase di scoperta, Armitage mette a disposizione le diverse caratteristiche che il framework offre per quanto riguarda la gestione degli host. È possibile importare degli host da una precedente scansione, oppure avviarne una ex novo direttamente dall'interfaccia di Armitage, al fine di popolare un database di target. Database che potrà poi essere visualizzato in qualsiasi momento, tenendo sotto controllo su quali host si sta lavorando e sui quali si hanno sessioni aperte.

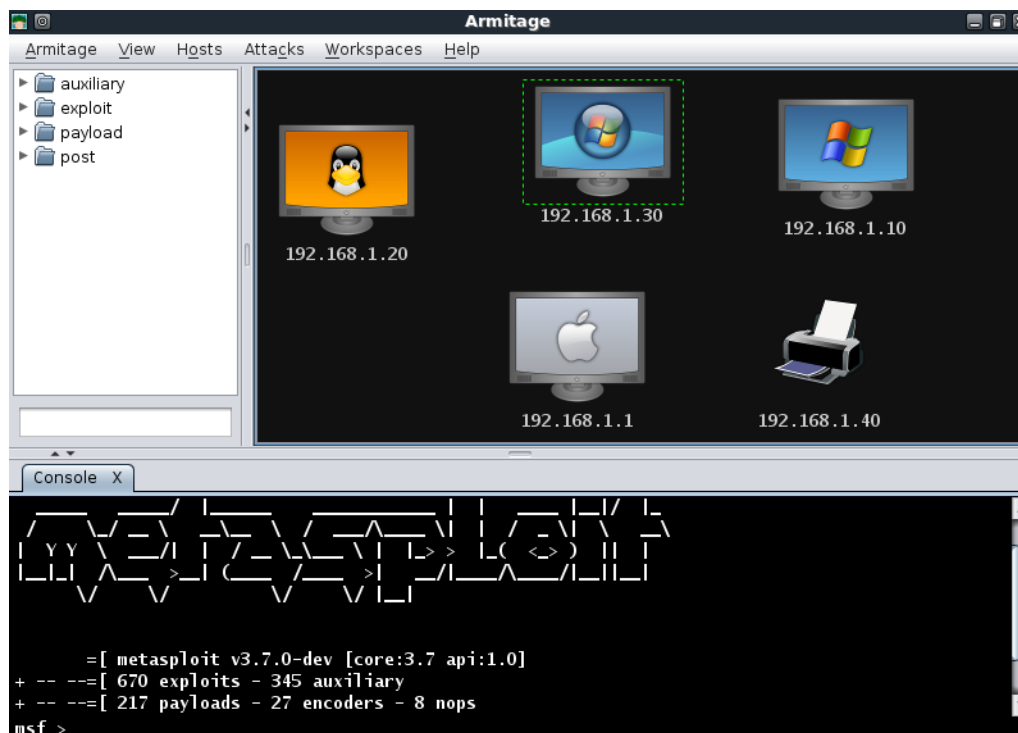


Figura 3.5: La schermata principale di Armitage, con alcuni host già inseriti.

Nella Fig. 3.5 è possibile vedere come gli host importati tramite un report di un qualche vulnerability scanner, oppure da una scansione fatta internamente al framework con Nmap ad esempio, siano rappresentati graficamente nell'area dei target, ognuno con una intuitiva immagine indicante il tipo e il sistema operativo dell'host.

Per lanciare Armitage è necessario prima far partire il demone di MySQL, con il seguente comando:

```
/etc/init.d/mysql start
```

Opzionalmente, si può lanciare un demone RPC (Remote Procedure Call) che può accettare connessioni locali e remoti da parte di Armitage. Per far partire il demone, è necessario scrivere questo comando:

```
sudo msfrpcd -S -U msf -P test -f
```

Una volta che questo demone è avviato, facciamo partire Armitage:

```
cd /pentest/exploits/framework3
./armitage
```

Comparirà una finestra, illustrata nella seguente figura, dove si dovranno compilare i dati richiesti e poi cliccare su "Connect" se prima è stato lanciato il demone, altrimenti su "Start MSF".

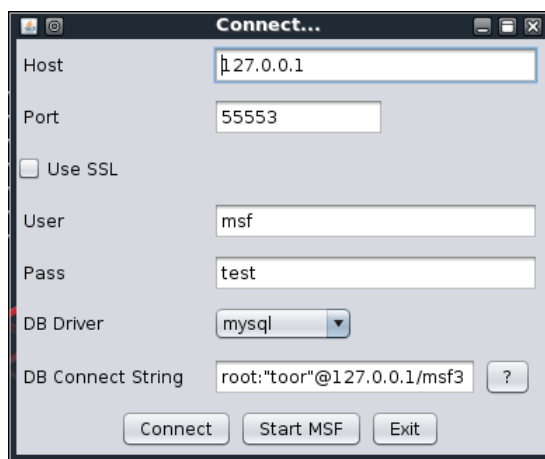


Figura 3.6: La finestra di connessione di Armitage.

3.4.4 Installazione

In questa parte del documento, si vedranno i passi chiavi per installare Metasploit Framework.

Linux

Iniziamo con l'installazione su un sistema equipaggiato con una versione di Ubuntu. Tale processo sarà utilizzabile anche su Kubuntu, Xubuntu e Debian. Prima di tutto è necessario installare le dipendenze necessarie di Ruby:

```
sudo apt-get install ruby libopenssl-ruby libyaml-ruby
libdl-ruby libiconv-ruby libreadline-ruby irb ri rubygems
```

Poi è necessario installare il client Subversion, ovvero uno dei più affidabili sistema centralizzato di controllo della versione di una applicazione. Tramite Subversion, potremo poi aggiornare MSF ogni qual volta lo riterremo necessario.

```
sudo apt-get install subversion
```

Infine, per far sì che anche le estensioni native del framework, possano funzionare egregiamente dovremo installare anche i seguenti pacchetti:

```
sudo apt-get install build-essential ruby-dev libpcap-dev
```

Una volta che tutte le dipendenze sono state soddisfatte, è possibile scaricare il tarball Unix dalla pagina ufficiale e dare i seguiti comandi:

```
tar xf framework-3.X.tar.gz
sudo mkdir -p /opt/metasploit3
sudo cp -a msf3/ /opt/metasploit3/msf3
sudo chown root:root -R /opt/metasploit3/msf3
sudo ln -sf /opt/metasploit3/msf3/msf* /usr/local/bin/
```

Dopo questi semplici passi, avremo un'installazione perfettamente funzionante del framework. Per poterlo sfruttare al meglio non ci resta che configurare correttamente anche l'aspetto dei database: Postgres o MySQL.

Windows

Sotto Windows, le cose sono più semplici in quanto vi è un semplice installer che si occupa di installare tutti i programmi necessari. Nel pacchetto disponibile sul sito web, sono compresi anche i seguenti programmi: Console2, Ruby, PostgreSQL, Java JDK, Subversion, VNCViewer, WinVI32 e Nmap.

Mac OS X

Prima di iniziare l'installazione di Metasploit Framework, sincerarsi di avere Xcode e MacPorts installati ed aggiornati. Il primo passo è quello di installare le librerie Ruby:

```
sudo port install ruby19 +nosuffix
```

Tale comando installerà Ruby nella cartella `"/opt/local"`. Per un elenco completo di tutti i file che sono stati installati, eseguire:

```
port contents ruby19
```

Infine, è necessario verificare che il path sia impostato correttamente in modo tale che `"/opt/local/bin"` sia elencata prima di `"/usr/bin"`.

```
$ echo $PATH
/opt/local/bin:/opt/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin
$ which ruby gem
/opt/local/bin/ruby
/opt/local/bin/gem
```

Come in ambiente Linux, per poter utilizzare tutte le funzioni di MSF è necessario installare un sistema di database. Lo staff di Metasploit raccomanda l'utilizzo di Postgres; in alternativa sqlite3 o MySQL.

3.4.5 Aggiornamento

L'aggiornamento rappresenta un'azione fondamentale per un penetration tester, un'opzione quasi maniacale per avere gli ultimi aggiornamenti rilasciati da Metasploit. La possibilità di testare gli ultimi exploit disponibili, contribuisce a trovare nuove vulnerabilità per poter essere sanate, in modo da avere la proprio infrastruttura sempre con un buon livello di sicurezza.

In ambiente Linux, è sufficiente eseguire il seguente comando:

```
/opt/framework/app/msfupdate
```

```
root@bt:~# msfupdate
[*]
[*] Attempting to update the Metasploit Framework...
[*]

U   lib/msf/core/exploit/smb.rb
A   lib/msf/core/exploit/ntlm.rb
U   lib/msf/core/exploit/mssql.rb
U   lib/msf/core/exploit/mixins.rb
U   lib/rex/proto/smb/client.rb
U   lib/rex/proto/smb/utils.rb
U   lib/rex/proto/ntlm/utils.rb
A   modules/auxiliary/scanner/oracle/isqlplus_sidbrute.rb
A   modules/auxiliary/scanner/oracle/isqlplus_login.rb
D   modules/exploits/linux/http/dr_b_syscall_linux_32.rb
A   modules/exploits/linux/misc/dr_b_remote_codeexec.rb
U   modules/exploits/windows/browser/adobe_flashplayer_avm.rb
U   modules/exploits/windows/browser/vlc_amv.rb
Updated to revision 12167.
```

Figura 3.7: Esecuzione di msfupdate su una macchina con BackTrack 4.

Sotto Windows, una volta che il Metasploit Framework è stato installato, può essere aggiornato tramite il collegamento "Metasploit Update". Sulle versioni di Windows che utilizzano l'UAC (User Account Control) o nelle situazioni in cui l'utente non dispone di privilegi di amministratore, il collegamento per l'aggiornamento deve essere eseguito come amministratore.

Capitolo 4

Conclusioni

Metasploit Framework è uno strumento molto potente quanto versatile. Tra i suoi maggiori vantaggi ho riscontrato:

- **multiplatforma:** Metasploit Framework è disponibile per tutti i sistemi operativi maggiormente utilizzati, compresi dispositivi mobile quali iOS o smartphone equipaggiati con il sistema di casa Google, Android.
- **vastissima comunità alle spalle,** capace di sostenere con ottimi risultati il progetto. Un importante numero di persone lavorano assiduamente allo sviluppo di nuovi exploit. Ciò si traduce in un'altissima frequenza di rilascio di aggiornamenti, sia di exploit ma anche per quanto riguarda miglioramenti al framework stesso e relative interfacce. Spesso, più volte al giorno viene rilasciata una nuova Subversion.
- **open source:** diversamente da quanto avviene nello sviluppo di software commerciale/proprietario, dove tipicamente è il marketing a definire le caratteristiche del prodotto, chi definisce le caratteristiche e l'evoluzione del software open source sono gli utenti della comunità di sviluppo. Questo significa che il framework, come altre applicazioni open source, è molto concreto in quanto deve rispondere a requisiti specifici segnalati dagli utilizzatori stessi del programma. Nel modello open source, inoltre, non esiste il vincolo di dover inserire delle nuove funzionali-

tà per giustificare una nuova versione del software a pagamento o un eventuale aumento del costo delle licenze.

- facilità di utilizzo: le varie interfacce messe a disposizione riescono a soddisfare le più disparate tipologie di utenti.
- requisiti hardware modesti: non è necessario disporre di una macchina con hardware aggiornato di recente per poter far girare Metasploit Framework.

Per quanto riguarda le limitazioni, è necessario comprendere che Metasploit Framework non è rivolto ai professionisti del settore, cioè a chi fa del penetration testing il loro pane quotidiano. Per tale tipologia di utenti, infatti, Rapid7 mette a disposizione le versioni a pagamento; rivolte proprio alle organizzazioni dove ci sono vari team di esperti, che collaborano per assicurare un buon livello di sicurezza all'intera infrastruttura. MSF è rivolto, sia a chi conosce veramente poco sui temi della sicurezza informatica, sia a chi è ben introdotto nella tematica, risultando un framework in grado di soddisfare diverse esigenze.

In particolare, credo che ogni amministratore di rete, a prescindere dalle dimensioni della infrastruttura che controlla, potrebbe sfruttare a pieno le funzionalità presenti in Metasploit Framework; specie se la fase di sicurezza è, come spesso accade, tralasciata. Le insidie sono assai numerose ed in costante aumento. Avere quindi la possibilità di disporre di un tale strumento, e per di più open source, non deve mai rappresentare un aspetto la cui importanza non sia considerata attentamente, trattando la questione in maniera del tutto superficiale.

Specie in un periodo di crisi economica mondiale come quello attuale, l'utilizzo di strumenti open source potrebbe fare effettivamente la differenza sul bilancio delle piccole e medie organizzazioni. Ribadisco, ancora una volta, che la questione della sicurezza informatica non va assolutamente sottovalutata, ma presa in considerazione rendendosi conto di quante opportunità le comunità open source ci rendono disponibili.

Appendice A

Demo 1

In questa prima dimostrazione, si mostreranno le potenzialità di Metasploit Framework, unite alle funzioni offerte da Armitage.

Come prima dimostrazione si mostrerà come sfruttare una vulnerabilità che affligge le versioni di Internet Explorer 6, 7 e 8. Il nome del modulo è "Microsoft Internet Explorer CSS Parsing Remote Memory Corruption Vulnerability" e l'exploit è localizzato nel seguente path:

```
windows/browser/ms11_003_ie_css_import
```

Questo modulo sfrutta una vulnerabilità di corruzione della memoria nel mshtml, ovvero il motore HTML di Microsoft. Mentre viene effettuata l'analisi di una pagina HTML contenente un import ricorsivo CSS, un oggetto C++ viene cancellato e poi riutilizzato. Questo fatto permette di eseguire codice arbitrario. Viene poi sfruttato una dll del .NET framework; per questo nel sistema target deve essere installata almeno la versione 2.0.50727 di tale framework.

La nostra macchina target per la demo, è una Windows XP con installato il Service Pack 3.

Come si può vedere nella seguente figura, dopo aver fatto partire il demone di MySQL, si lancia Armitage. In questo caso, non avendo fatto partire precedentemente il demone msfrpcd, si clicca su "Start MSF".

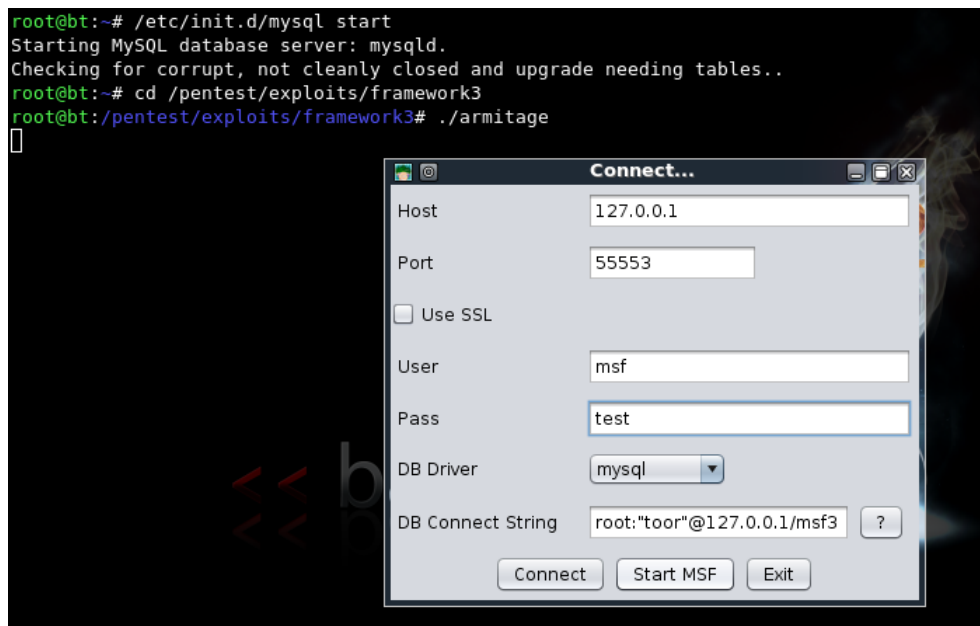


Figura A.1: Lancio di Armitage.

Il passo successivo è selezionare il suddetto modulo.

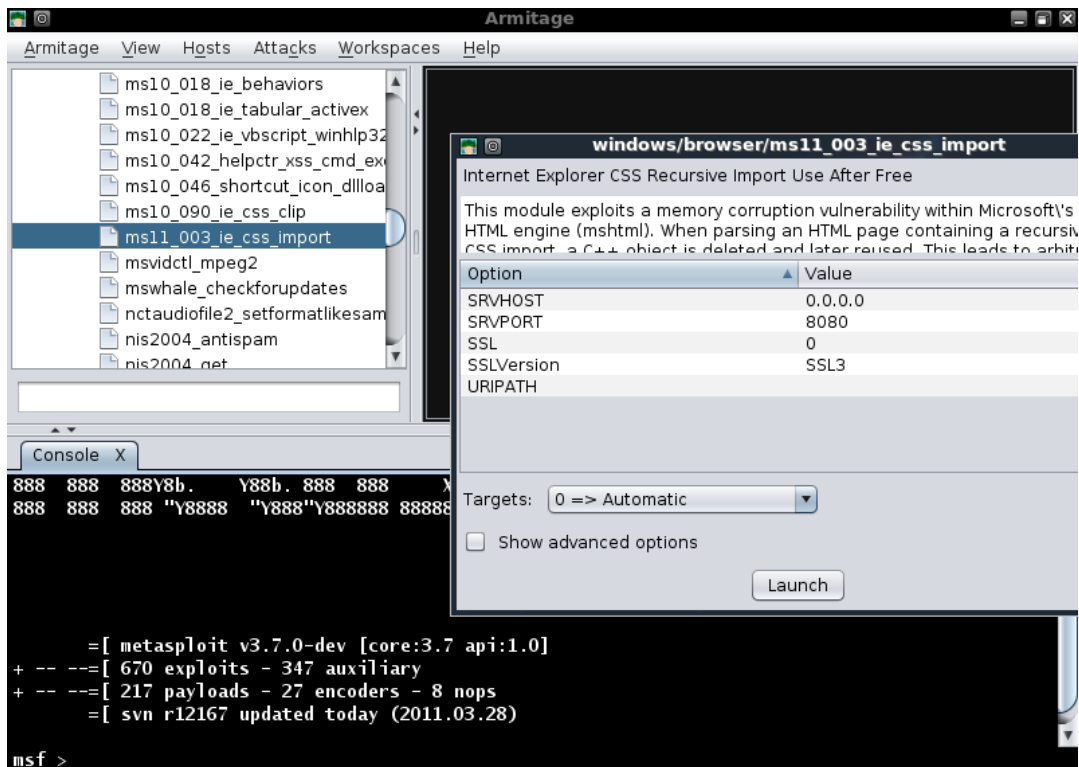


Figura A.2: Selezione dell'exploit.

Nella finestra di configurazione dell'exploit, si dovrà impostare l'indirizzo IP del server, la porta e il path.

Successivamente, un utente cliccherà sull'url creato da Metasploit Framework.

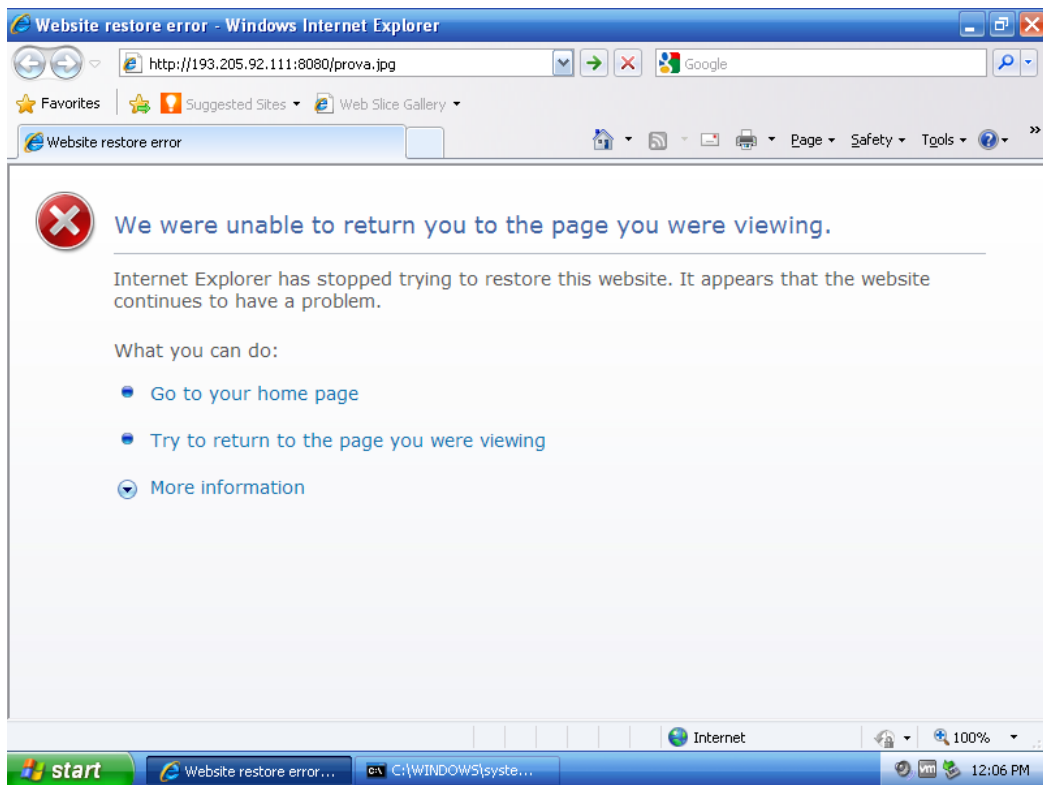


Figura A.3: Un utente clicca su di un link contraffatto.

L'exploit viene eseguito con successo. La macchina target è penetrata: Armitage rappresenta questo fatto graficamente con la immagine con bordo rosso e delle saette.

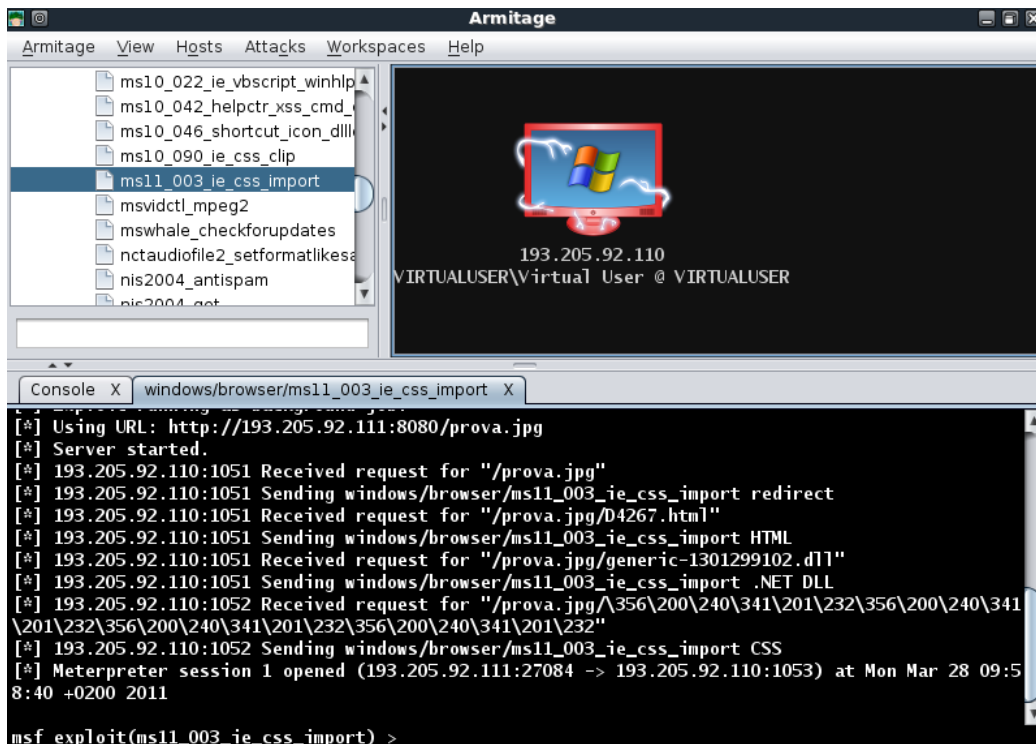


Figura A.4: Esecuzione con successo dell'exploit.

Automaticamente Armitage apre una sessione di Meterpreter, con la quale è possibile interagire.

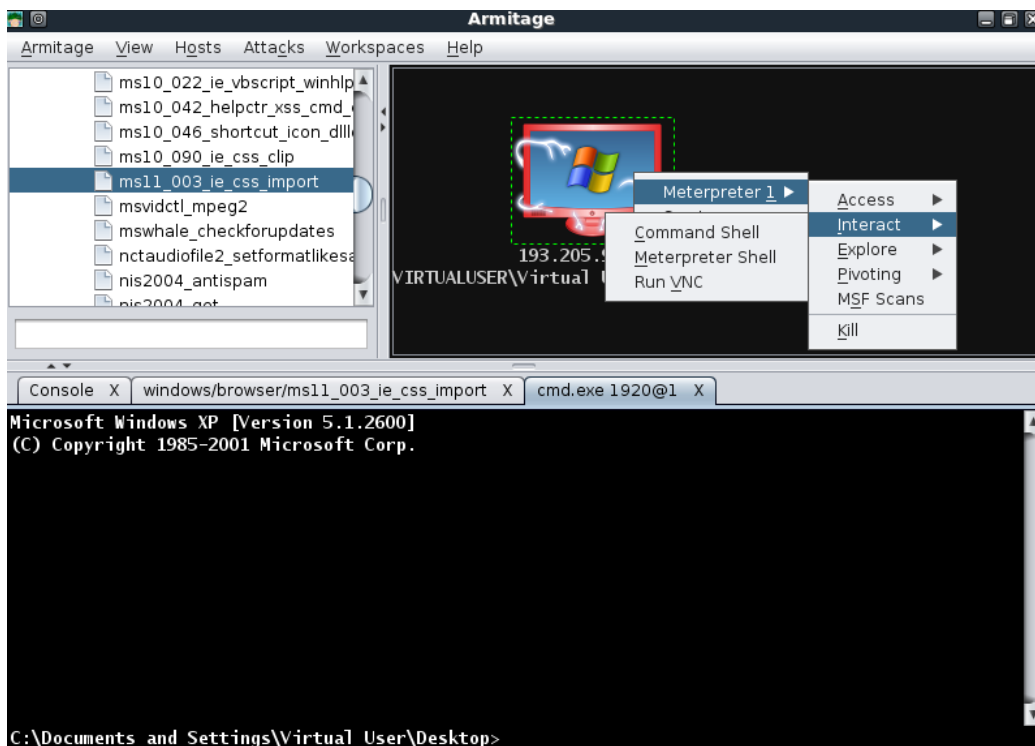


Figura A.5: Sono possibili vari modi per interagire con la sessione di Meterpreter.

In questo caso, si può interagire attraverso una shell dei comandi, una shell di Meterpreter oppure con una sessione di VNC.

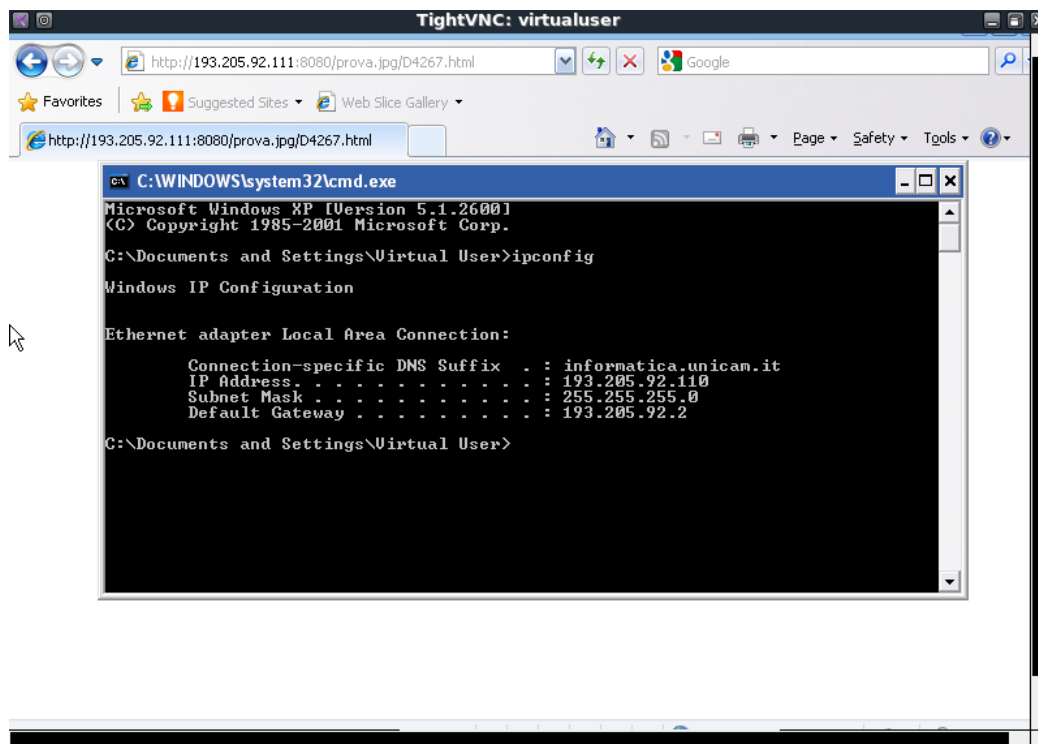


Figura A.6: Apertura di una sessione VNC tramite Meterpreter.

Appendice B

Demo 2

Nella prima dimostrazione, le modalità con cui procedere nel lancio dell'exploit erano precise. Quel caso specifico, rispecchia una circostanza in cui un ricercatore voglia testare una specifica vulnerabilità di un sistema. Nell'uso quotidiano, Metasploit Framework viene anche utilizzato per eseguire scansioni verso sistemi di cui molti aspetti sono ignari, come ad esempio il sistema operativo o la tipologia della macchina.

In questo contesto, tramite l'uso di Nmap, possiamo capire quali siano le porte aperte di un sistema target. Le seguenti figure spiegheranno questo procedimento e le successive fasi.

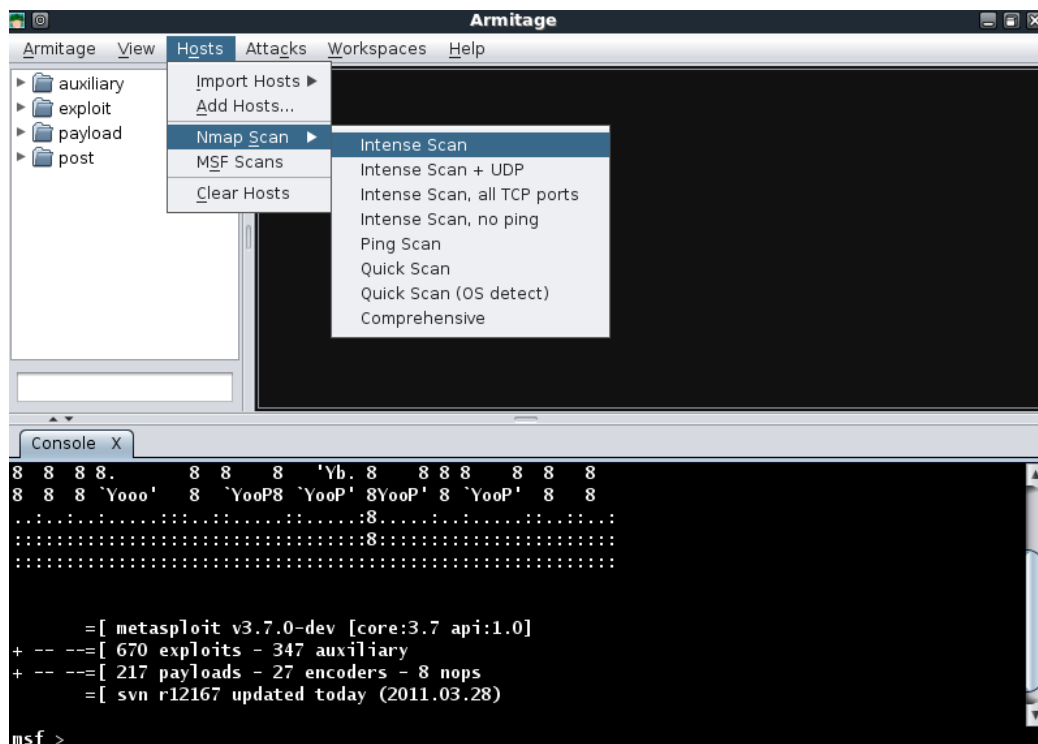


Figura B.1: Esecuzione di una scansione tramite Nmap, da Armitage.

A questo punto, è necessario inserire l'indirizzo IP o un'intera subnet da scansionare.

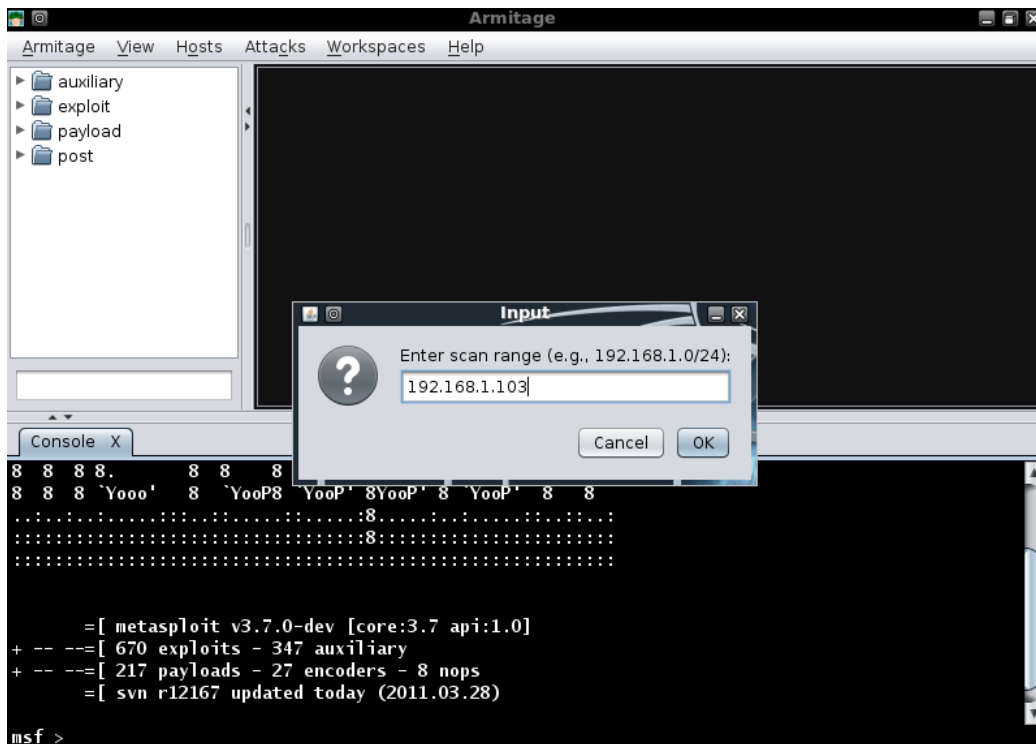


Figura B.2: Inserimento del range di indirri IP da scansionare.

Una volta completata la scansione, Armitage mostra l'host associando un'immagine per evidenziare il suo sistema operativo. Con un clic col tasto destro è possibile visualizzare i vari servizi attivi sulla macchina target. In questo esempio si tratta di una macchina Ubuntu con un sistema LAMP (Linux Apache MySQL Php) installato.

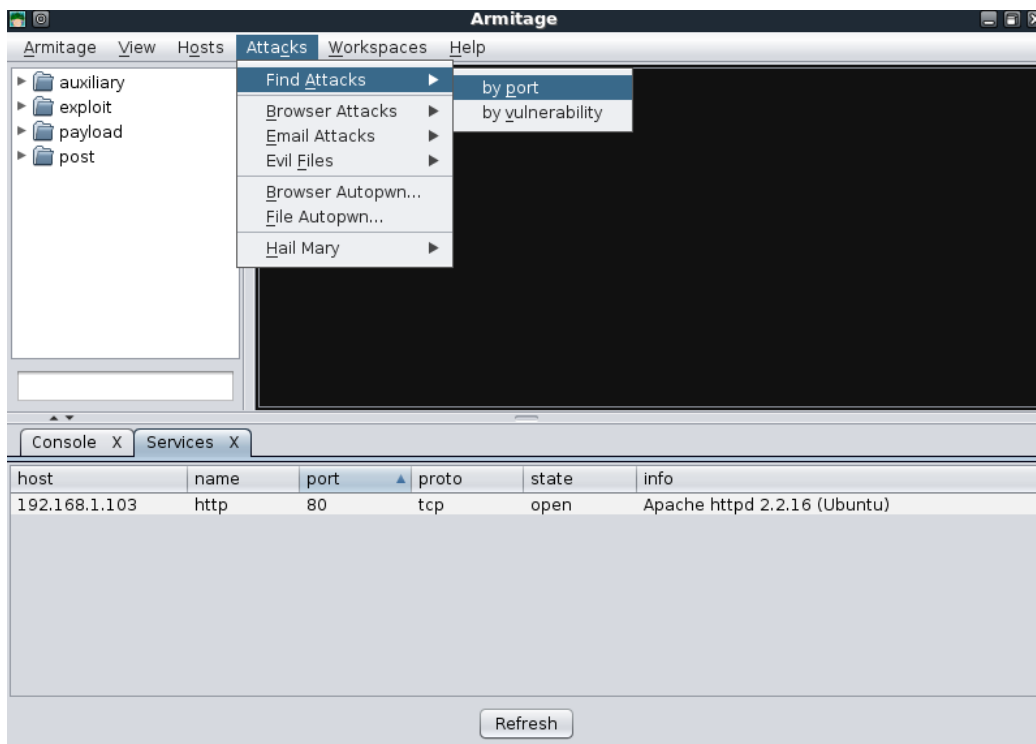


Figura B.3: Ricerca degli attacchi disponibili.

Armitage consiglia determinati exploit da poter testare. La ricerca degli attacchi può avvenire in base alle porte o alle vulnerabilità.

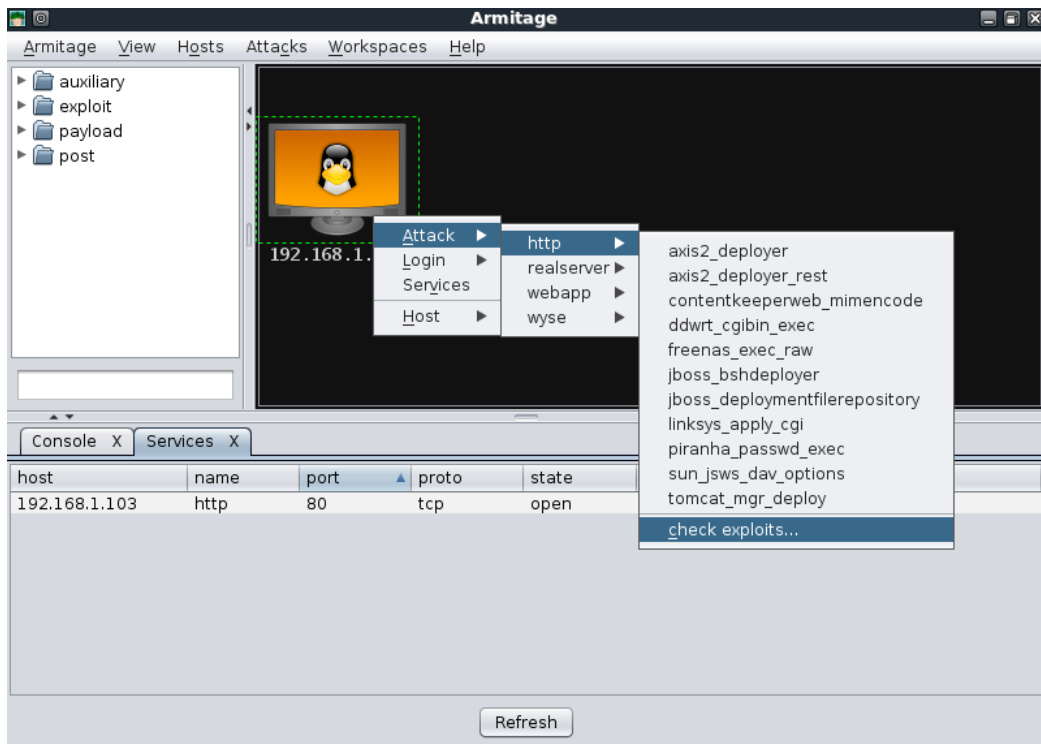


Figura B.4: Visualizzazione degli attacchi relativi.

Nella scheda "Attack" dell'host precedentemente, è possibile visualizzare l'elenco degli attacchi disponibili, in base ai servizi attivi dell'host stesso, classificati in base alla loro famiglia. Ora è possibile lanciare ogni singolo exploit oppure cliccare su "check exploits...". Tale funzione permette di eseguire un controllo per tutti gli exploit della relativa famiglia.

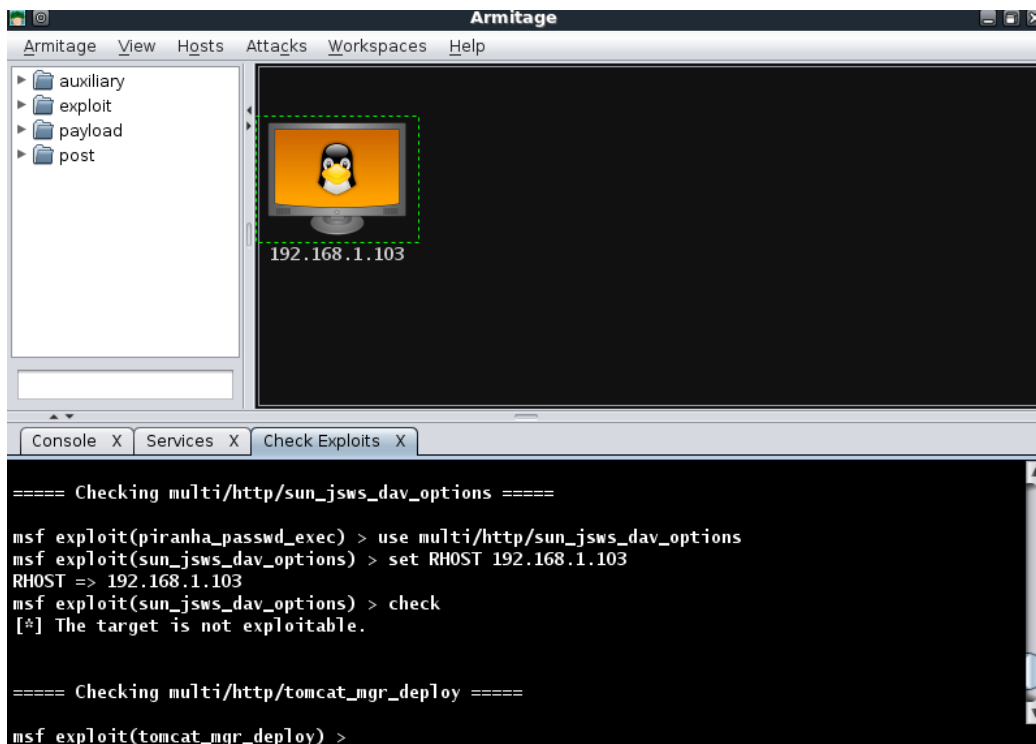


Figura B.5: La funzione "check exploits...".

Nella figura è possibile notare il controllo degli exploit selezionati. In questo caso il risultato è indicato dalla seguente dicitura:

```
[*] The target is not exploitable.
```

Purtroppo la funzione "check" è supportata solo da una piccola parte degli exploit presenti nel Metasploit Framework.

Ringraziamenti

Poche righe non possono bastare per esprimere la mia profonda gratitudine nei confronti di uomini e donne, che mi hanno accompagnato fino a questo importante giorno; che mi hanno portato ad essere *quello che sono*.

Tengo quindi a ringraziare la mia famiglia che, con il suo appoggio, mi ha
permesso di arrivare fino qui.

Un sentito ringraziamento per il professore Fausto Marcantoni e per la sua
costante fiducia, ricambiata dalla mia grande stima nei suoi confronti.

Infine, ma non meno importanti, tutti gli amici che da anni o da qualche
mese sono entrati, o usciti, dalla mia vita.

Bibliografia

- [1] Maynor D., Mookhey K. H., Cervini J., Roslan F., Beaver K., "Metasploit Toolkit for Penetration Testing Exploit Development and Vulnerability Research", 2007
- [2] Harper A., Harris S., Ness J., Eagle C., Lenkey G., Williams T., "Gray Hat Hacking - The Ethical Hackers Handbook" Terza Edizione, 2011
- [3] Burns B., Granick J.S., Manzuik S., Guersch P., Killion D., Beauchesne N., Moret E., Sobrier J., Lynn M., Markham E., Iezzoni C., Biondi P., "Security Power Tools", 2007
- [4] Long J., Bayles A. W., Foster J. C., Hurley C., Petruzzi M., Rathaus N., Wolfgang M., "Penetration Tester's - Open Source Toolkit"
- [5] Marsh N., "Nmap Cookbook - The Fat-free Guide to Network Scanning"
- [6] "Penetration testing guide", Corsaire - Experts at securing information, <http://www.penetration-testing.com/>
- [7] Fossi M., Egan G., Johnson E., Adams T., Blackbird J., Graveland B., McKinney D., "Symantec Report on Attack Kits and Malicious Websites", http://www.symantec.com/content/en/us/enterprise/other_resources/b-symantec_report_on_attack_kits_and_malicious_websites_21169171_WP.en-us.pdf
- [8] "NMap - Free Security Scanner for Network Exploration & Security Audits", <http://nmap.org/>

- [9] "Tenable Nessus - Tenable Network Security",
<http://www.nessus.org/products/nessus>
- [10] "OpenVAS - Open Vulnerability Assessment System Community Site",
<http://www.openvas.org/>
- [11] "Top 100 Network Security Tools", <http://sectools.org/>
- [12] "Top 3 Vulnerability Exploitation Tools",
<http://sectools.org/sploits.html>
- [13] "Top 10 Vulnerability Scanners", <http://sectools.org/vuln-scanners.html>
- [14] "WebInspect", <http://securityinnovation.com/security-report/October/tools/WebInspect.htm>
- [15] "Core Security", <http://www.coresecurity.com>
- [16] "w3af - Web Application Attack and Audit Framework",
<http://w3af.sourceforge.net/>
- [17] "Nikto2 - CIRT.net", <http://cirt.net/nikto2>