Strategie di attacco avanzate

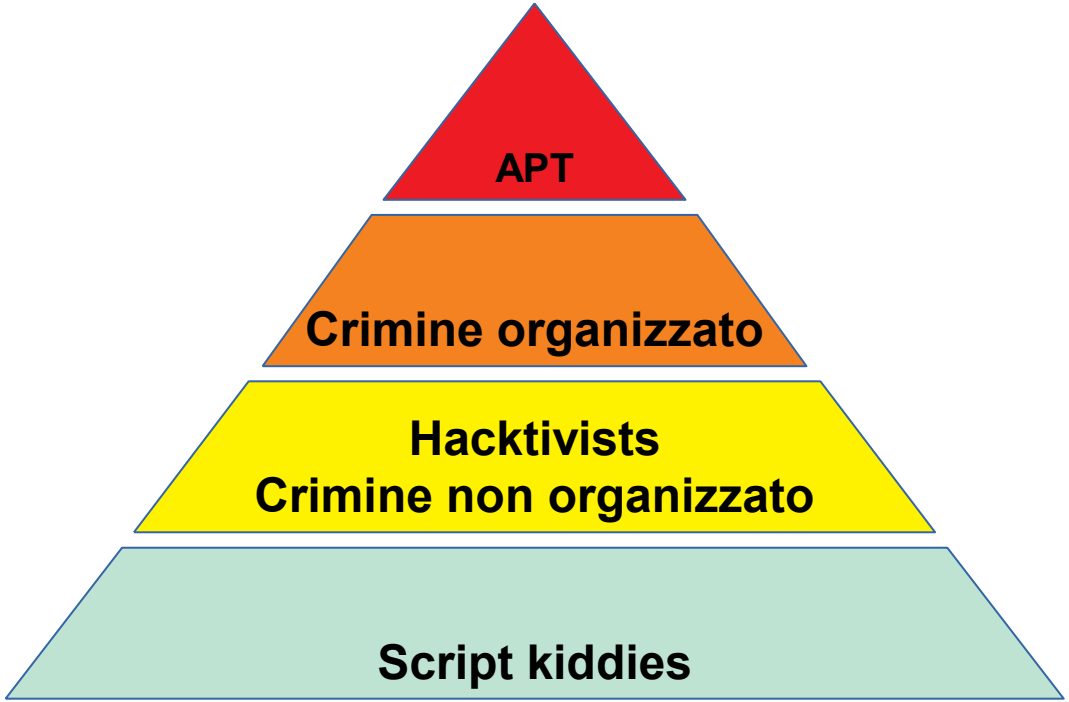
Indice

- Attacchi DoS e DDoS
- Botnet
- Attacchi di injection
- APT

Strategie di attacco

- Gli attaccanti possono essere caratterizzati in base ai loro **obiettivi** e alle loro **capacità**
- Obiettivo: quali dati/risorse l'attaccante vuole ottenere/compromettere
- Capacità: di quali strumenti/competenze l'attaccante dispone

Tipi di attaccanti



Impatto e mitigazione

- Possiamo assumere che un attaccante con adeguate capacità possa raggiungere qualsiasi obiettivo
- Caratterizzare correttamente gli attaccanti può comunque permetterci di mitigare le conseguenze e prendere contromisure più efficaci

Indice

- Attacchi DoS e DDoS
- Botnet
- Attacchi di injection
- APT

Denial of Service - DoS

- Un DoS è un attacco che ha lo scopo di degradare/interrompere un servizio esaurendone o compromettendone le risorse
  - CPU, memoria, capacità di banda, …
- I bersagli tipici sono le applicazioni web
  - Per costruzione queste devono gestire le richieste provenienti dalla rete

Flooding

Flooding

- Ogni applicazione si appoggia su una infrastruttura di rete
  - Interrompere (parzialmente o interamente) la rete rende le applicazioni irraggiungibili e causa danni materiali
    - Es. Perdita di ordini
- Saturare la banda con messaggi “spazzatura” può causare ritardi nella ricezione o la perdita di messaggi importanti

Tipi di flooding

- I tipi di flooding cambiano con il canale utilizzato
- Alcuni canali interessanti sono
  - Ping
  - HTTP
  - TCP

Ping flooding

- Un attaccante ha a disposizione un numero maggiore di risorse della vittima
  - banda, potenza di calcolo, terminali, etc.
- L'attaccante invia il maggior numero possibile di ping alle macchine della vittima per saturare il sistema di gestione dei pacchetti
- Però l'attaccante è individuabile: i pacchetti contengono gli indirizzi del mittente
  - il sistema bersaglio deve rispondere ai ping e può comunque ignorare la sorgente

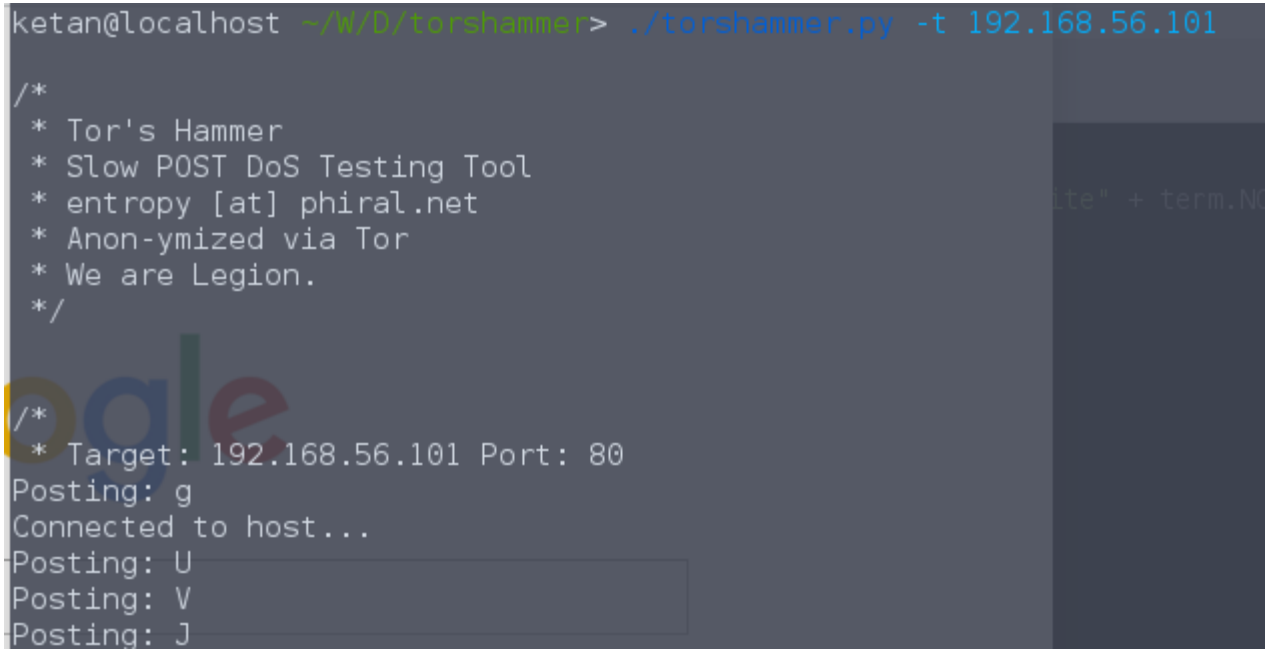
HTTP flooding

- Richieste HTTP (tipo GET o POST)
  - Protocollo a livello applicativo
  - Scopo del protocollo: trasferimento di interi documenti
  - Ogni singola richiesta genera un treno di pacchetti
  - La gestione da parte del server è più onerosa rispetto a quella per protocolli di livello più basso (es. ping)
- Si può usare solo se il bersaglio è una applicazione web

HTTP floading via Tor's Hammer

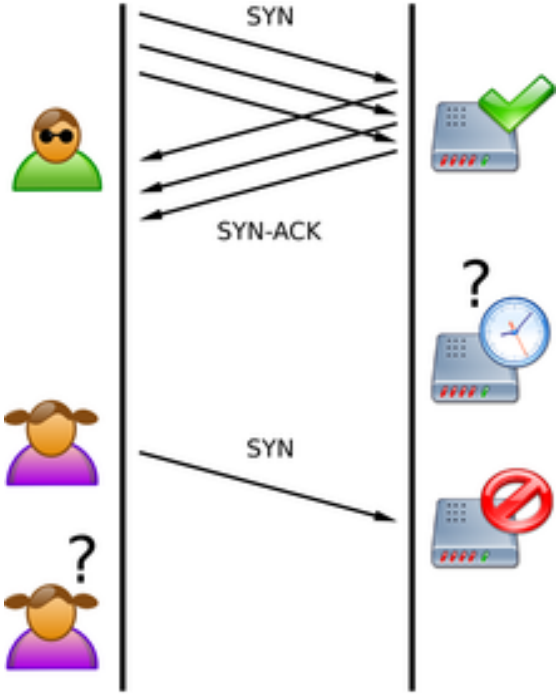
- https://sourceforge.net/projects/torshammer/
- Genera richieste HTTP anonime usando la rete TOR
- Sfrutta slow POST rate messages
  - Estende la durata delle sessioni inviando messaggi con bassa cadenza
  - Es. imposta content-length alto e poi invia un byte alla volta
- Strumenti simili:
  - HULK (HTTP Unbearable Load King) https://github.com/darkwarrior3/hulk
  - RUDY (R U Dead Yet?) https://sourceforge.net/projects/r-u-dead-yet/

HTTP floading via Tor’s Hammer



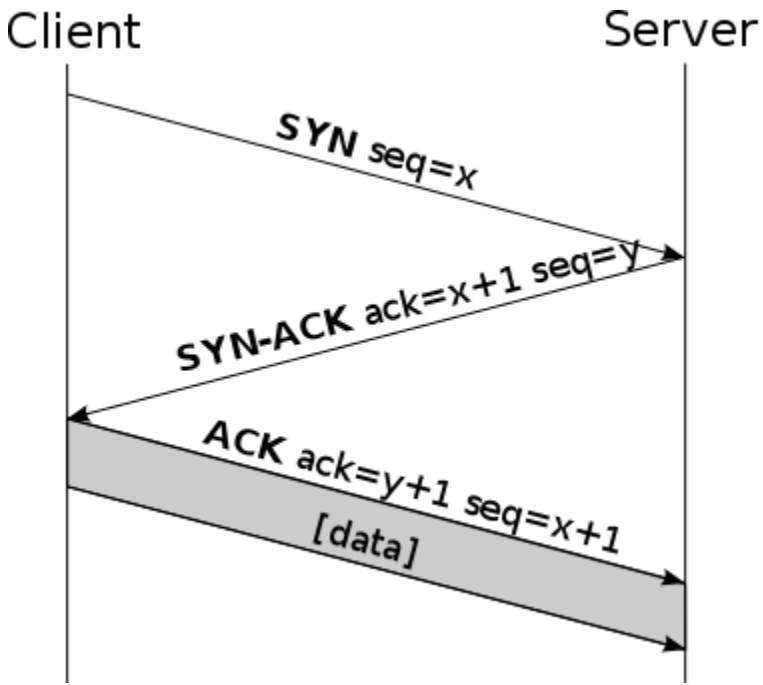
SYN flooding

- Le richieste di connessione TCP iniziano con una fase di handshaking
- Ogni macchina connessa alla rete usa TCP/IP
  - Nessuna limitazione sui bersagli
- A livello trasporto le richieste consumano meno risorse



TCP handshaking

- Client inizia la sessione inviando il messaggio SYN con i dettagli necessari per la connessione
  - Address, port, seq number, …
- Server aggiunge Client alla tabella delle connessioni e invia SYN-ACK ripetutamente fino a quando Client non invia ACK (c'è però un time out)



SYN flooding

- Vengono inviati solo SYN e non gli ACK
- Il Server tenta di stabilire la connessione inviando una sequenza di SYN-ACK
- Per i SYN l'attaccante usa l'IP di un'altra macchina (address spoofing), pertanto il Server invia SYN-ACK all'indirizzo spoofed che ignorerà i messaggi

Flooding con altri protocolli

- ICMP
  - Internet Control Message Protocol
  - Protocollo a livello IP usato ad esempio per traceroute
- UDP
  - Tipicamente sfruttato per flooding di specifici servizi
  - Es. streaming dati

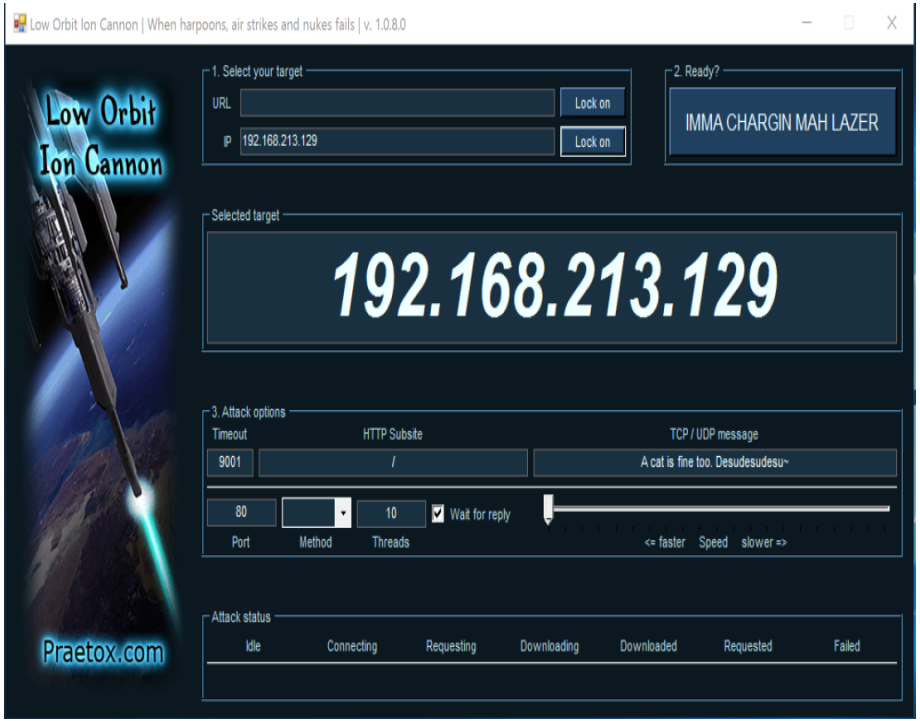
Distributed DoS

- Un attacco DoS implementato tramite una infrastruttura distribuita
  - Es. Low Orbit Ion Cannon
- L'infrastruttura può anche essere costruita utilizzando macchine compromesse
  - Tipicamente dette zombie o bot
- Gli zombie possono essere anche dispositivi IoT
  - Es. Mirai botnet (più dettagli in seguito)

Low Orbit Ion Cannon

- https://sourceforge.net/projects/loic/
- Strumento open source per TCP, UDP e HTTP flooding
  - Originariamente sviluppato per stress test
- Usato per alcuni attacchi notevoli
  - Es. DDoS a Universal music (2012)
  - https://money.cnn.com/2012/01/19/technology/megaupload_shutdown/index.htm

Low Orbit Ion Cannon



- Sviluppato per testare la risposta dei sistemi a grandi quantità di richieste
- Usato per attacchi DDoS: più utenti si coordinano per colpire un unico bersaglio e inondarlo di richieste

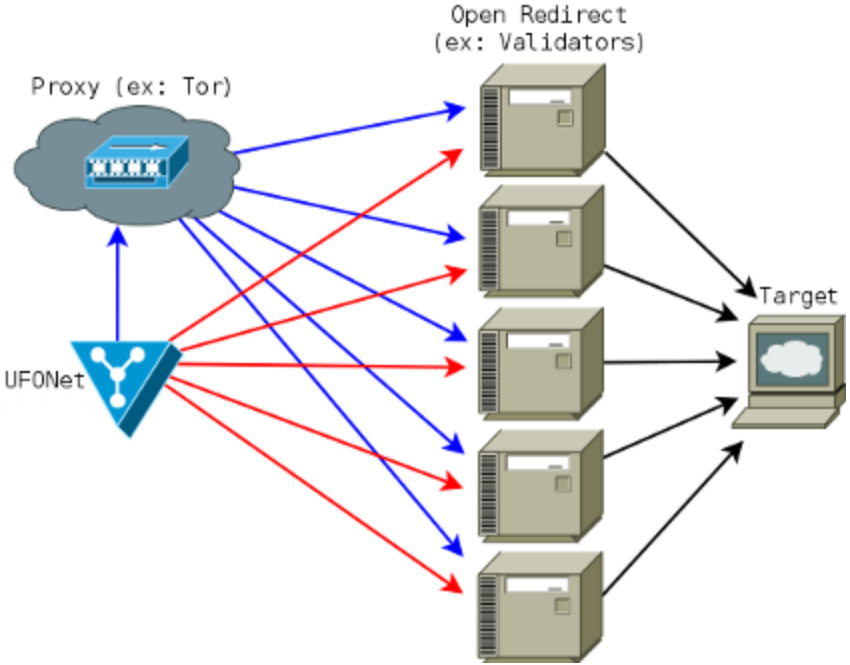
Tecniche DDoS

In mancanza di coordinamento tra attaccanti o di risorse sufficienti, si usano tecniche alternative:

- Reflection:
  - Sfruttare altre macchine, non controllate, per colpire i bersagli (giocando di sponda)
- Amplification:
  - Sfruttare il funzionamento della rete per aumentare il volume di traffico

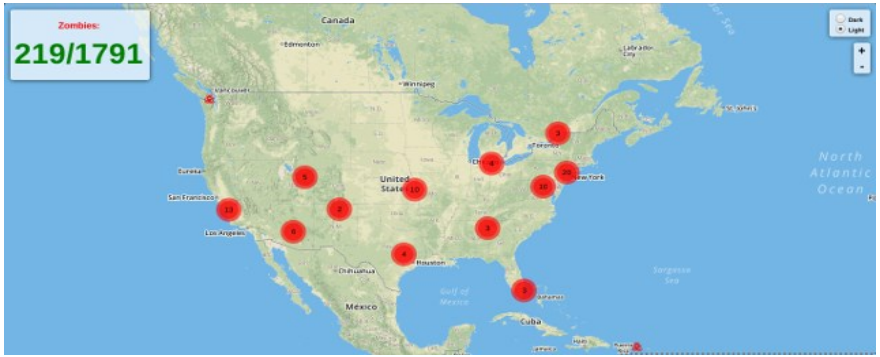
Reflection: UFONet

- Usa macchine “neutrali” per generare il volume di traffico necessario per il flooding
  - Aka confused deputy attack
- In alcuni casi vengono generati loop di risonanza
  - Forzando due macchine a interagire in maniera indefinita



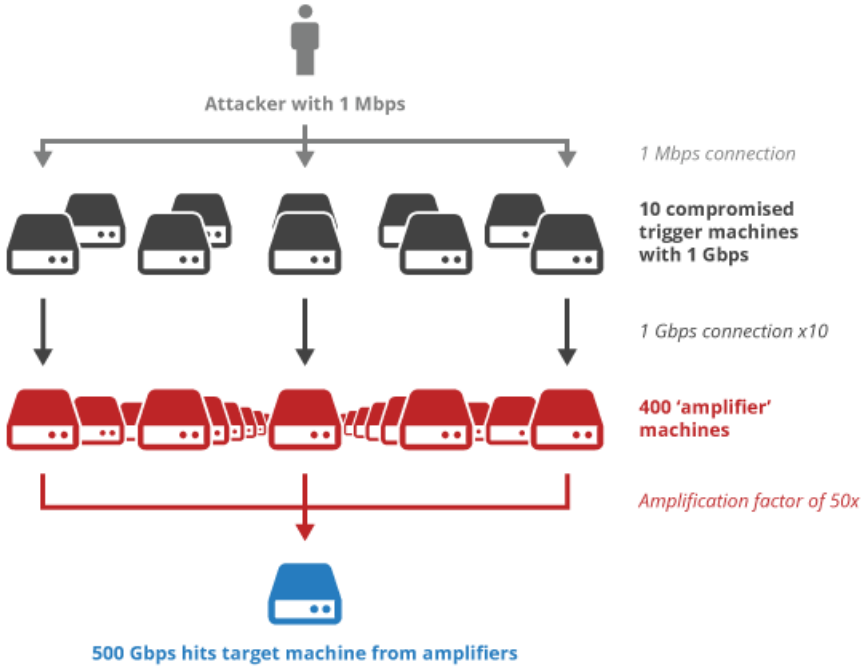
Reflection: UFONet

- Utilizza come zombie servizi che supportano redirect
  - https://validator.w3.org/check?uri=TARGET_URI
- Usa query specifiche sui motori di ricerca per trovare gli zombie
  - Aka dorks
  - Es. l'url contiene "redirect/?url="



Amplification

- Il traffico viene aumentato sfruttando protocolli che generano un volume di dati molto maggiore di quelli inviati con una singola richiesta
- Usabile anche in combinazione con reflection

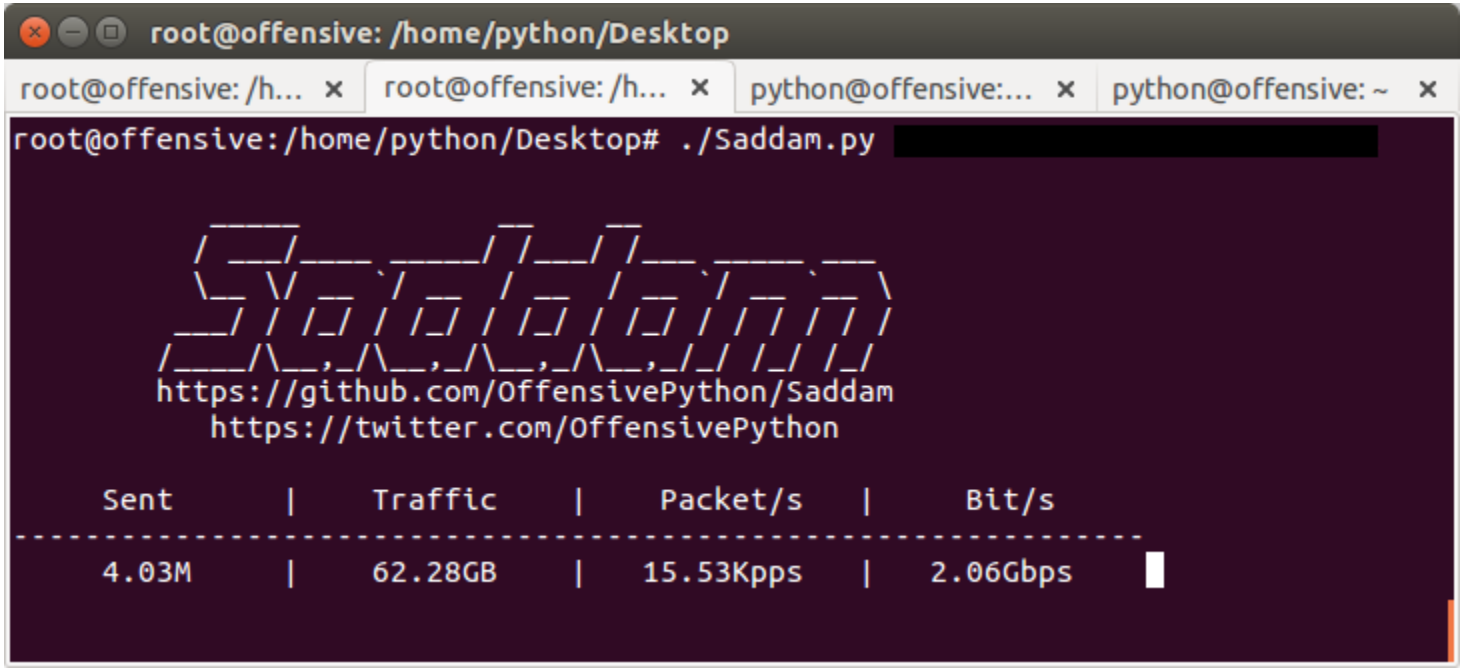


https://www.bansalonsecurity.com/

DNS Amplification

- Il lookup di un domain name è sostanzialmente asimmetrico
  - Es. una richiesta da ~80 bytes può generare 512 bytes di risposta
- Usando l'indirizzo della macchina obiettivo si possono generare volumi di traffico DNS ingenti
- Saddam è un tool che utilizza questa tecnica

Saddam


root@offensive: /home/python/Desktop
root@offensive: /h...
root@offensive: /h...
python@offensive:...
python@offensive: ~
root@offensive:/home/python/Desktop# ./Saddam.py
https://github.com/OffensivePython/Saddam
https://twitter.com/OffensivePython
Sent | Traffic | Packet/s | Bit/s
4.03M | 62.28GB | 15.53Kpps | 2.06Gbps

Riferimenti

- https://github.com/dotfighter/torshammer
- https://ufonet.03c8.net/
- https://sourceforge.net/projects/loic/

Indice

- Attacchi DoS e DDoS
- Botnet
- Attacchi di injection
- APT

Botnet

- Un insieme di macchine compromesse, sotto il controllo di un attaccante
- Orchestrate secondo uno schema gerarchico:
  1. Bot Master
  2. Command & Contro (C&C) server
  3. Bots

Utilizzo

- Le botnet sono una risorsa distribuita a disposizione dell'attaccante
  - Potenza di calcolo, storage, connettività
- Possono essere utilizzate per effettuare attacchi
- Possono anche essere messe a disposizione del migliore offerente

Mercato delle botnet

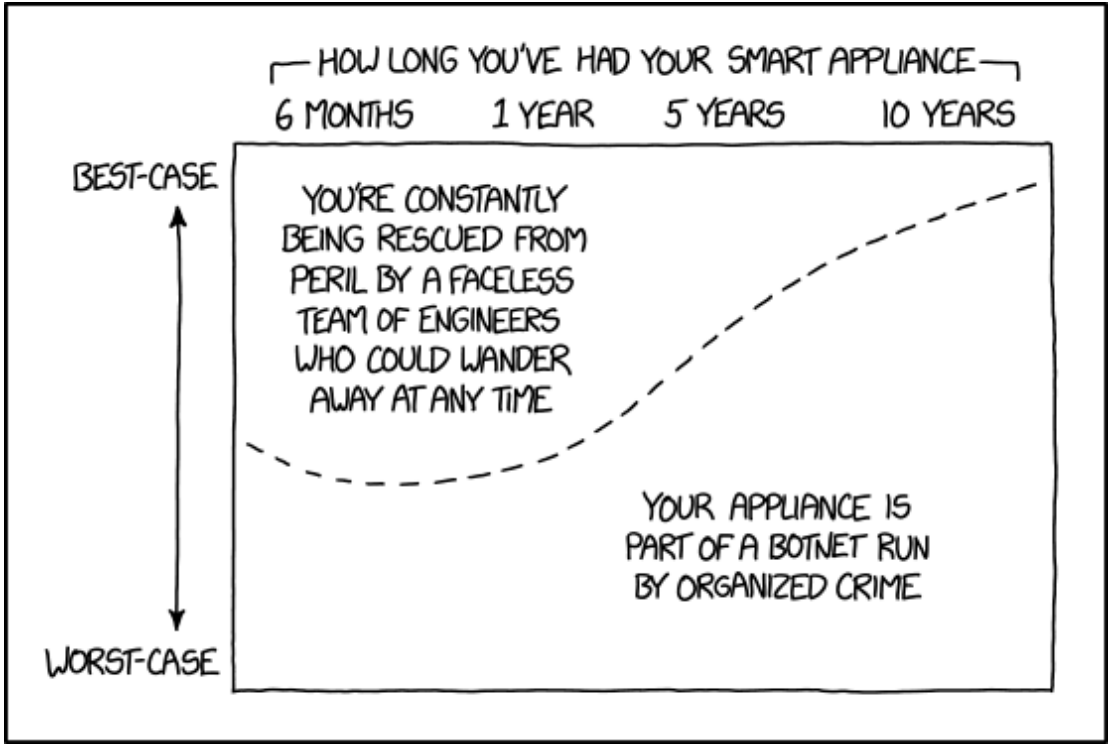
PrivateLayer
638 iscritti

For buy add me on discord: Down#9591 |Telegram:@allowedspoof
Monthly - 20£
Lifetime - 40£
Reseller - 80£ (You keep profit)
BOTNET,#BOTNET

Come si crea una botnet

- Compromettendo macchine vulnerabili
  - L'attacco dipende dal tipo di macchina
  - Per esempio sfruttando vulnerabilità note in sistemi obsoleti
- Lo scopo è il deployment di un client che esegua i comandi ricevuti dal C&C server

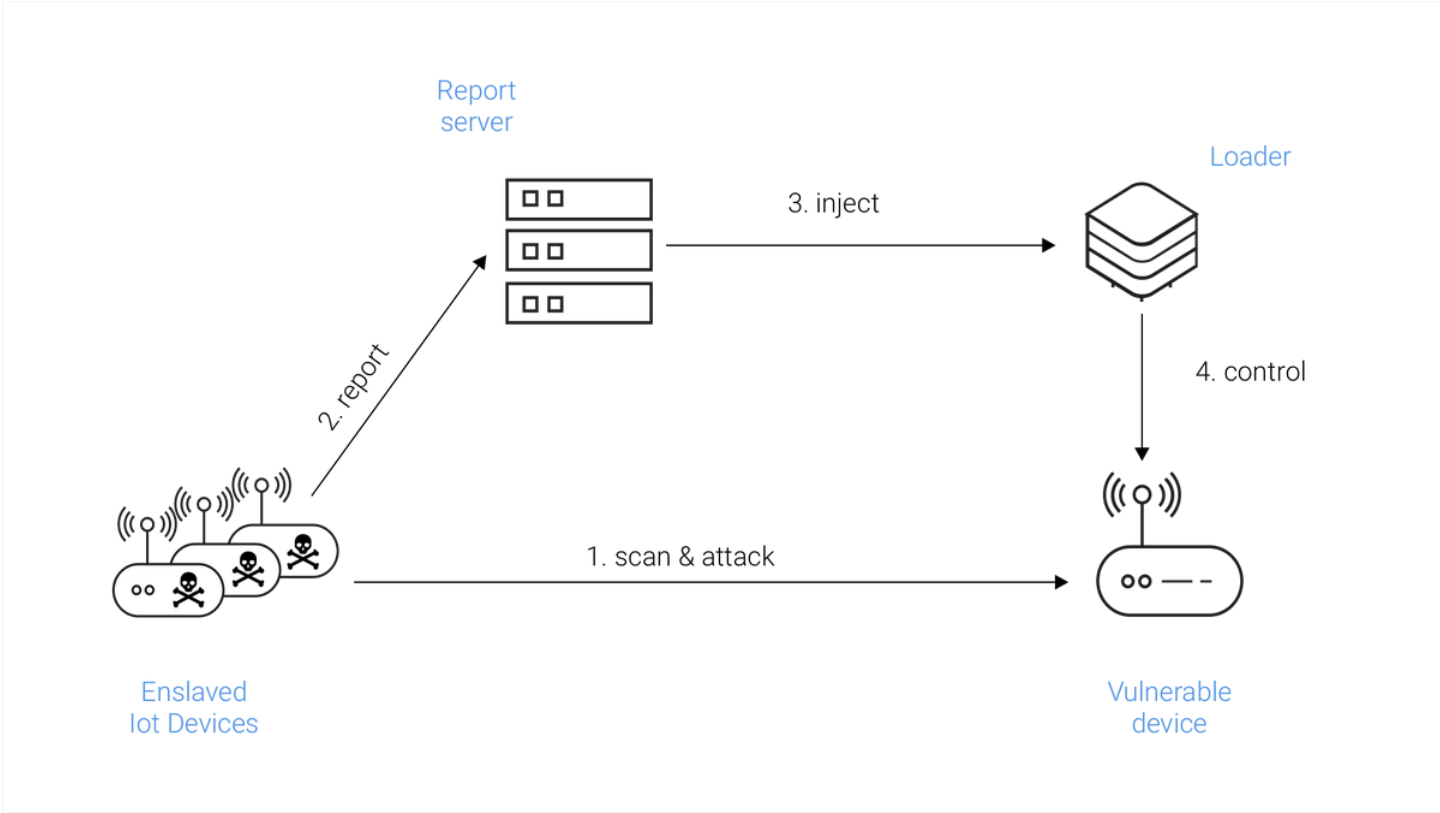
Internet of Bots



Mirai: una botnet per IoT

- Molti dispositivi IoT non hanno meccanismi di sicurezza built-in
  - usano autenticazione base tramite username e password
- Mirai
  - usa una lista di 61 credenziali comuni (admin/admin, root/pass, …)
  - Sonda la rete in cerca di dispositivi che accettano connessioni telnet

Mirai: comportamento



Altre botnet

- Dendroid: trojan per C&C di dispositivi Android
  - Primo malware a bypassare Bouncer (il malware scanner usato da Google Play)
- Kraken: probabilmente la botnet più grande
  - 400K dispositivi
  - Trojan che infetta macchine Win e Mac
  - Usata per diffusione spam

Riferimenti

- https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf
- https://www.usenix.org/legacy/event/nsdi09/tech/full_papers/john/john_html/

Indice

- Attacchi DoS e DDoS
- Botnet
- Attacchi di injection
- APT

Attacchi di injection

- Ogni sistema accessibile da remoto riceve input e genera output
  - Es. web applications, database, web services, ...
- Gli input vengono analizzati e processati
- Un attacco di injection consiste nel fornire input che alterano il comportamento del sistema

Injection: fattori abilitanti

- Gli attacchi di injection sono possibili quando gli input confluiscono in un comando eseguito da un interprete
- Implicitamente questo trasforma un dato (input) in un comando da eseguire

Alcuni tipi di injection

- Code injection
- Cross-site scripting (XSS)
- SQL injection
- NoSQL injection

Code injection

- Caso base: l'interprete è il Sistema Operativo
- In PHP l'injection può avvenite tramite varie funzioni che passano comandi direttamente al Sistema Operativo:
  - exec(…),
  - shell_exec(…),
  - passthru(…)
  - …

Code injection: esempio

exec("cat ./".$_GET['user'], $output);

- Se user = 'bob' viene eseguito cat ./bob
- Se user = '../../../etc/passwd' allora viene eseguito
  cat ./../../../etc/passwd
- Se user = 'bob;ls' viene eseguito
  cat ./bob;ls

Mitigare code injection

- Garantire che gli input siano della forma attesa (ciò che deve essere una stringa rimanga una stringa!)
  - È utile per difendersi da ogni attacco di injection
  - Si ottiene tramite validazione o manipolazione (usando ad esempio exec(escapeshellcmd(...))
- Applicare politiche di controllo degli accessi
  - Tramite whitelist di comandi e file accessibili

Cross-site scripting (XSS)

- L’interprete dei comandi è presente nel browser
  - Es. javascript
- Il codice viene iniettato in una pagina web generata dinamicamente
  - Le vittime dell’attacco sono i client che ricevono la pagina iniettata

XSS: esempio

Welcome <?php echo $_POST[‘user’]; ?>!

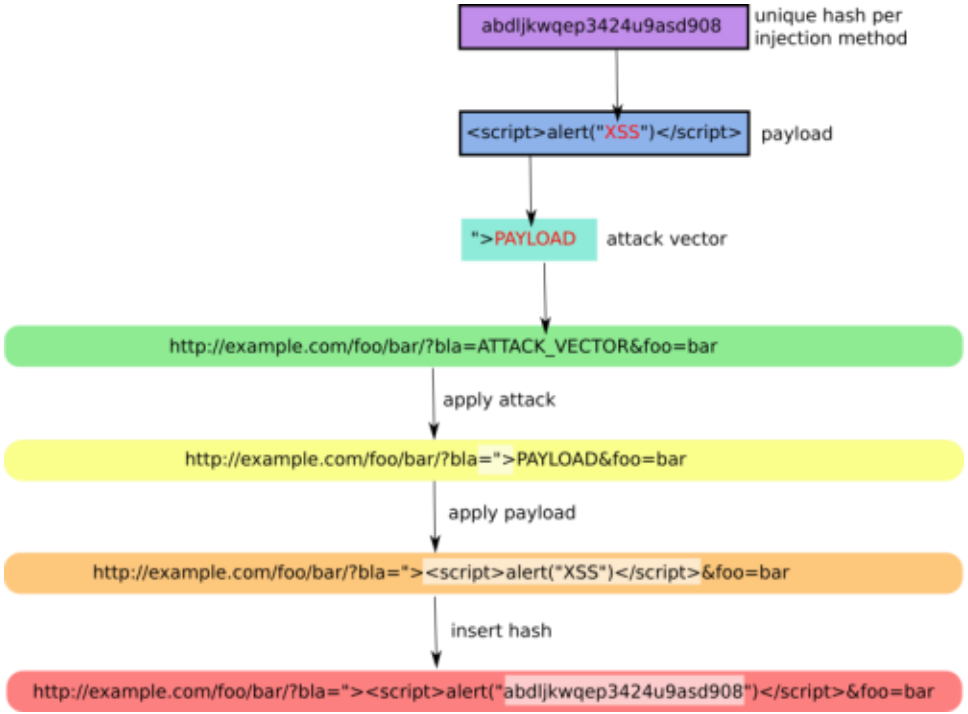
- Se user = ‘bob’ genera Welcome bob!
- Se user = ‘<script>alert(1)</script>’ genera Welcome <script>alert(1)</script>!
  - Esegue il comando alert in javascript

Cross-site scripter: Un tool per XSS

È un tool per il testing automatico di vulnerabilità XSS

- Include +1300 vettori di attacco (attack vector)
- Usa i vettori per individuare i campi potenzialmente iniettabili
- Inserisce gli input (payload) con il codice javascript nei vettori per individuare i campi effettivamente iniettabili e quindi le vulnerabilità

Cross-site scripter: Esempio di test



Tipi di XSS

- **Reflected**: il codice iniettato appare solo nella pagina vulnerabile in presenza di una richiesta specifica
  - (tipicamente usato per phishing tramite pagine vulnerabili, es. inviando link alla pagina iniettabile)
- **Stored**: il codice iniettato viene memorizzato e colpisce tutti gli utenti che si collegano alla pagine

Database injection

- Moltissime applicazioni web leggono/scrivono dati su database
  - Es. tabelle per utenti, ordini, prodotti, ...
- L'interazione avviene inviando comandi a un Database Management System (DBMS)
  - Interprete del linguaggio di query
- Le query possono essere iniettabili

SQL injection

- Structured Query Language (SQL) è il linguaggio di riferimento per i Database (DB) relazionali (MySQL, SQLite, Oracle DB, …)
- La query viene costruita come una stringa e inviata al DBMS tramite apposite funzioni
- Forzando l'esecuzione di query arbitrarie l'attaccante può
  - Accedere evitando l'autenticazione
  - Estrarre informazioni dal DB (e.g., user enumeration)
  - Modificare i dati del DB

SQL injection: esempio

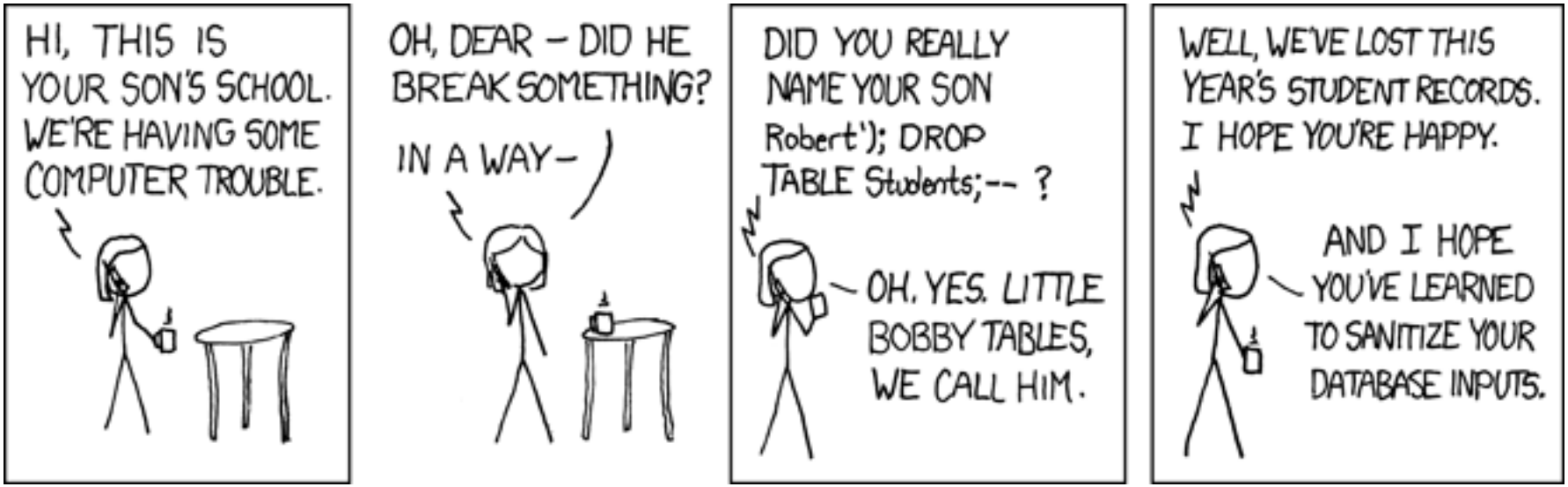
$sql = “SELECT * FROM users WHERE name='”.$name.“' AND pwd='”.md5($pwd).“’”;

$result = $db->query($sql); #print $result

Se name = ‘bob’ e pwd = ‘123’ si ottiene
$sql = “SELECT * FROM users WHERE name='bob' AND pwd='ba1f...’”;
print info about bob

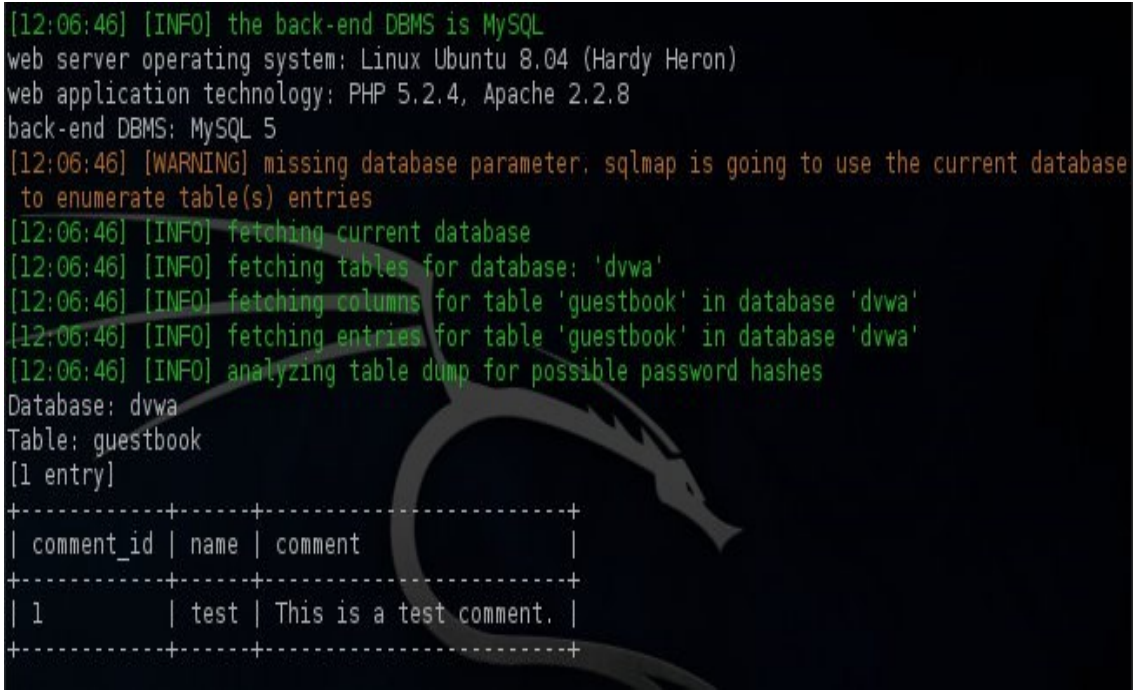
Se name = ‘bob' OR 1=1 -- ’ e pwd = ‘123’ si ottiene
$sql = “SELECT * FROM users WHERE name='bob' OR 1=1 -- ' AND pwd='ba1f...’”; # print info about all users

SQL injection



SQL injection

SQLmap



- SQLmap è uno strumento di penetration testing open source che automatizza il processo di rilevazione delle vulnerabilità dei DB SQL, inclusa SQL injection

Blind SQL injection

- In alcuni casi non è necessari estrarre direttamente i dati dal DB tramite SQL injection
- E’ sufficiente poter capire quando una query è valutata vera o falsa (es. dal contenuto che viene mostrato in risposta o dal tempo di gestione della richiesta)
- Se la query viene manipolata dall’attaccante si possono estrarre dati un bit alla volta
- Es. “esiste un utente il cui nome inizia per ‘m’?”

NoSQL injection

- NoSQL si riferisce ai DB non relazionali che (tipicamente) non usano SQL come linguaggio di query
- Le query vengono sottomesse al DBMS in formato JSON
  - db->users->find({name:‘bob’, pwd:‘123’})
- Il formato JSON non previene la possibilità di injection
  - il rischio deriva dal modo in cui viene costruita la query, come per SQL

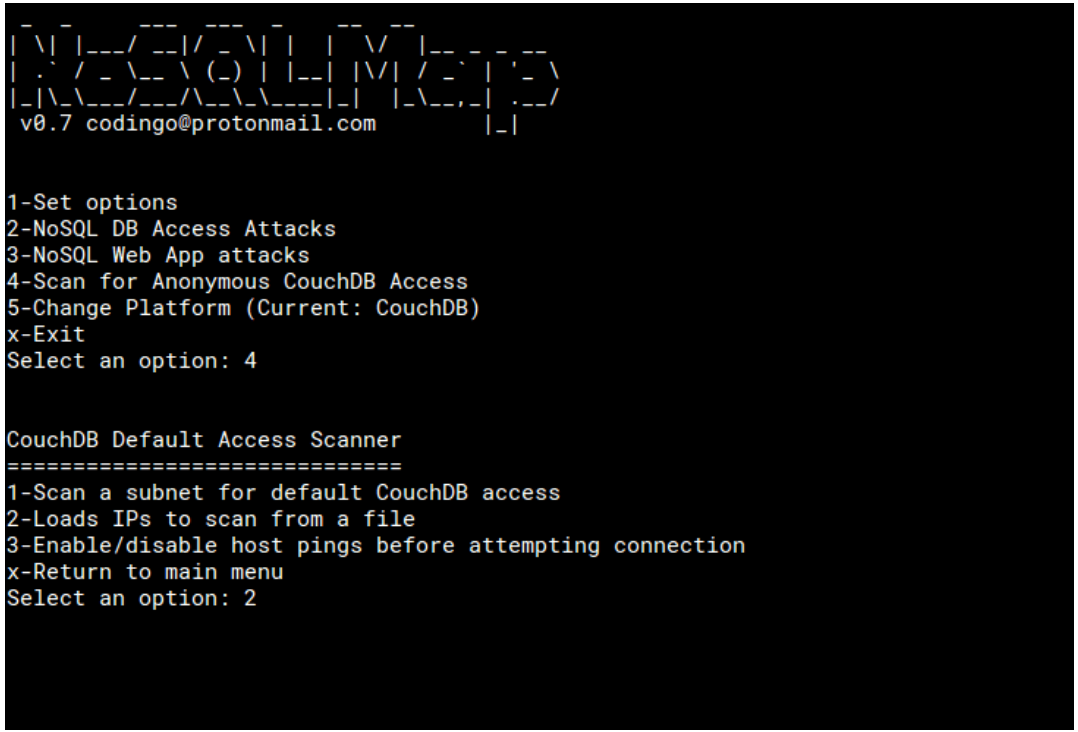
NoSQL injection: esempio

$json = “{name:'”.$user.“', pwd:'”.$pwd.“'}”

- Se user = ‘bob', $or: [{}, { 'a':'a’ e pwd= “'}], $comment:'’
- $json = “{name:'bob', $or: [{}, { 'a':'a', pwd=''}], $comment:''}”
- Equivalente a

… WHERE name = ‘bob’ AND (TRUE OR (‘a’= ‘a’ AND pwd=‘’))

NoSQLmap



- NoSQLMap è un tool progettato per individuare vulnerabilità NoSQL
- Orientato a testare MongoDB e CouchDB

Riferimenti

- http://sqlmap.org/
- https://github.com/codingo/NoSQLMap
- https://www.owasp.org/index.php/SQL_Injection
- https://xsser.03c8.net/
- No SQL, No Injection? Examining NoSQL Security https://arxiv.org/abs/1506.04082

Indice

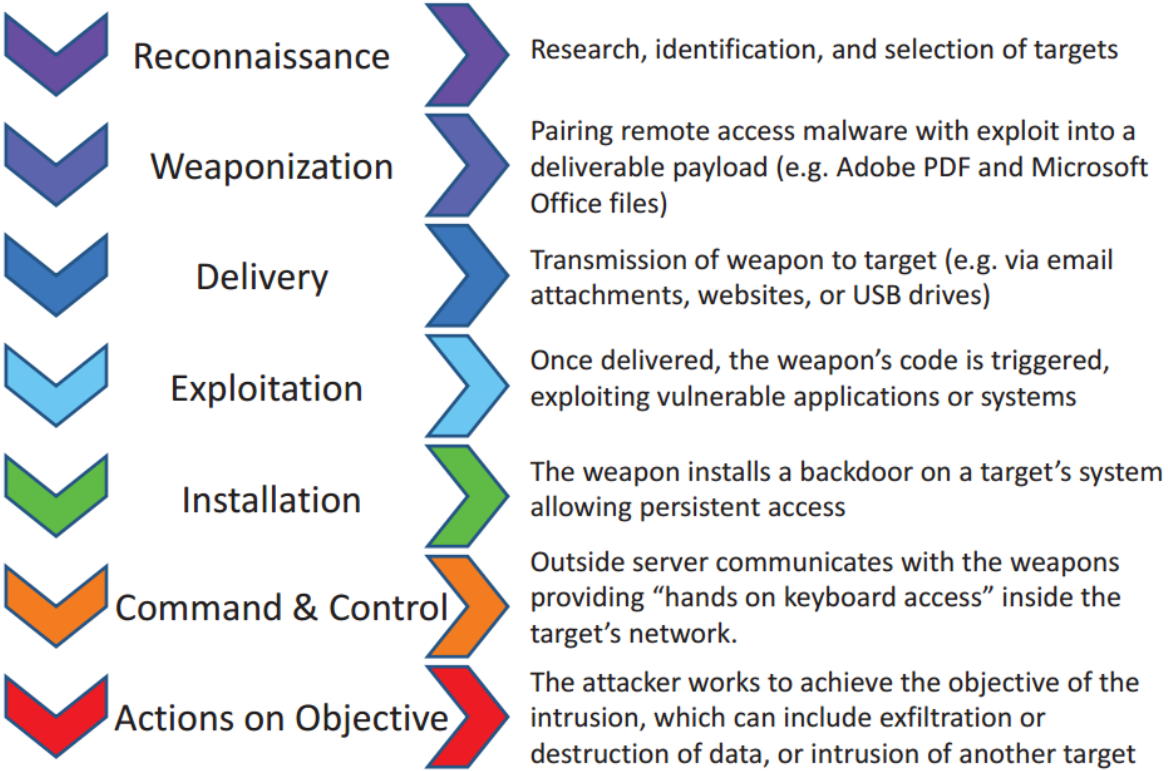
- Attacchi DoS e DDoS
- Botnet
- Attacchi di injection
- APT

Identificazione delle minacce

APT: definizione

- Advanced: Disponibilità di considerevoli risorse e competenze tecniche
- Persistent: Operazioni strutturate, durature nel tempo e guidate da obiettivi strategici
- Threat: Attacchi pianificati tramite il coordinamento intelligente di agenti umani e di strumenti automatici

Intrusion kill chain di un APT spiegare



Motivazioni

Bersagli

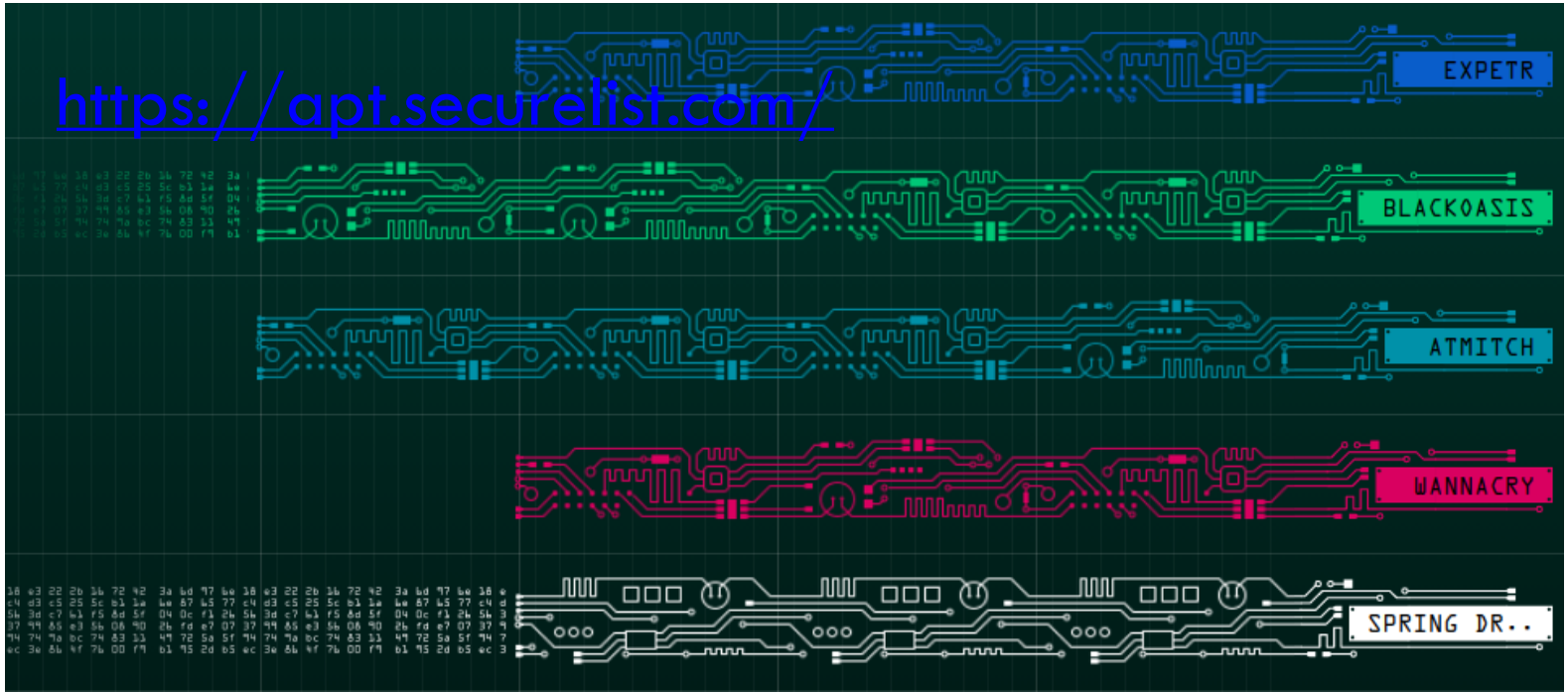
- Infrastrutture nazionali
- Segreti e informazioni sensibili
  - Es. progetti militari
- Servizi finanziari
- Privati cittadini
- ...

Attori

- Sviluppatori di malware
- security analysts & pentesters
- Vendors
- Domain experts
- Social analysts
- ...

Liste degli APT noti

https://apt.securelist.com/



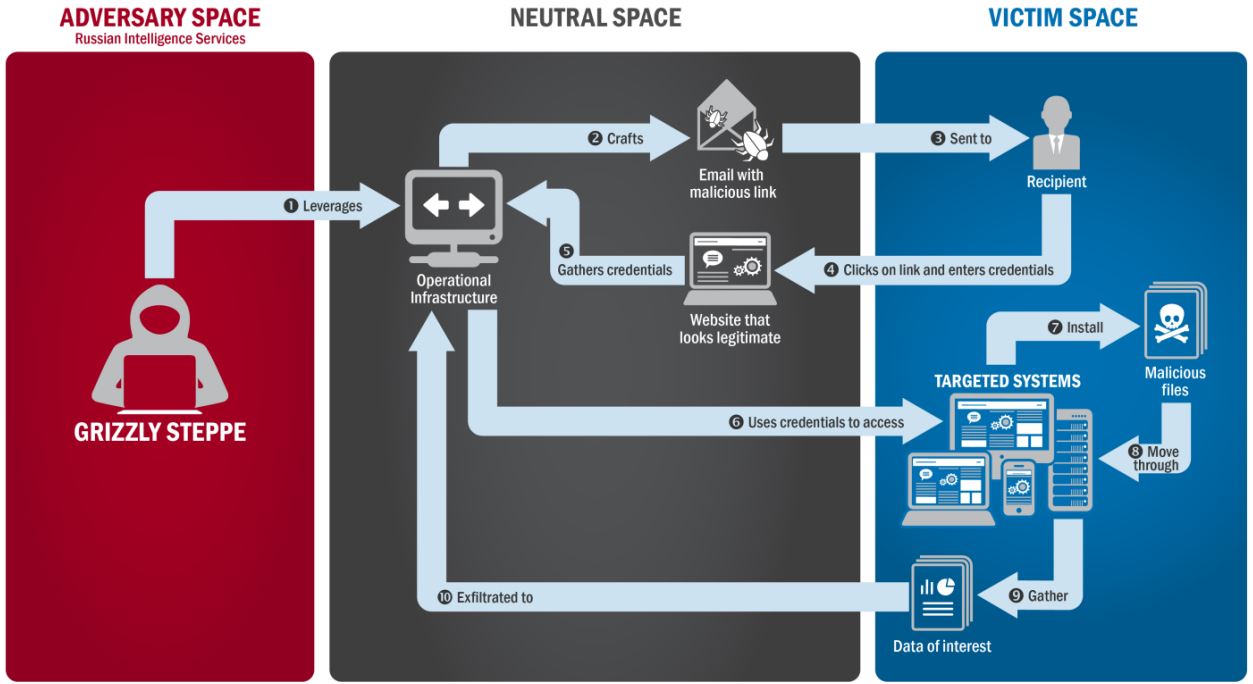
Come individuare un APT

- Indicatori di compromissione (IoC) ovvero informazioni estratte dopo o durante un attacco per costruire una fingerprint dell'attaccante
  - Indirizzi IP e nomi di dominio
  - Hash dei file utilizzati
  - Vulnerabilità sfruttate
  - Bersagli secondari e movimenti laterali
  - ...

APT28: Grizzly steppe/Fancy bear/...

- Ritenuto responsabile di diversi attacchi a istituzioni internazionali
  - Sistemi militari, servizi governativi, media, ...
- Attribuito ai servizi militari russi (GRU)
- I bersagli principali sono situati in paesi NATO

IoC di APT28: spear phishing



IoC di APT28: malware e indirizzi

- X-AGENT: C&C trojan per Win, Mac e Linux

IP	Dominio
139.5.177.205	malaytravelgroup.com
89.34.111.107	undseats.com
…	…

File	Hash
chost.exe	46e2957e699fae6de1a212dd98ba4e2bb969497d
msoutlook.dll	c53930772beb2779d932655d6c3de5548810af3d
…	…

X-Agent: infection (Mac)

- Alla vittima viene inviata una mail con un falso PDF in allegato
  - roskosmos_2015-2025
- L'allegato è il trojan Komplex che apre un falso PDF e si occupa dell'installazione di X-Agent

X-Agent: moduli e comportamento

- BootXLoader: inizializza il malware, si assicura che sia persistente ed eseguito al boot.
  - Verifica che non sia attivo un debugger
- MainHandler: gestisce i comandi del server C&C

Codice	Comando
'g'	apre una shell remota sulla macchina infetta
'h'	ottiene la lista delle applicazioni installate
'n'	cattura una schermata e la invia al server

X-Agent: moduli e comportamento

- HTTPChannel: gestisce le comunicazioni con il server C&C
- Password: estrae password da alcuni applicativi, es. firefox
- KeyLogger: registra gli input da tastiera
- Cryptor: cifra le comunicazioni con il server C&C

- Una versione per Android ha infettato dispositivi mobili usati nel conflitto tra Russia e Ucraina
  - Distribuito tramite una app per puntamento dell'artiglieria D-30
- L'esercito Ucraino potrebbe aver perso tra il 15% e il 20% di D-30 a causa di X-Agent

Riferimenti

- https://www.crowdstrike.com/wp-content/brochures/FancyBearTracksUkrainianArtillery.pdf
- https://apt.securelist.com/
- https://download.bitdefender.com/resources/files/News/CaseStudies/study/143/Bitdefender-Whitepaper-APT-Mac-A4-en-EN-web.pdf

Approfondimento 1: mitigazione DoS

- Gli attacchi DoS e DDoS hanno lo scopo di sovraccaricare di traffico una rete in modo da interromperne o degradarne gravemente le funzionalità. Per raggiungere questo risultato devono generare un volume di dati e richieste superiore a quello che la rete può gestire normalmente.
- Tali volumi di traffico sono spesso molto differenti da quelli normalmente ricevuti dalla rete e, per questo, possono essere identificati come “anomali” da strumenti di anomaly detection. Il traffico anomalo può quindi essere filtrato o rallentato allo scopo di diminuire la portata dell’attacco.
- Ulteriori tecniche di difesa includono black/white listing e l’utilizzo di architetture scalabili. Una white list è un elenco di indirizzi che possono legittimamente collegarsi alla rete e inviare richieste. Avere una white list è una soluzione ideale dato che difficilmente un attaccante potrà generare traffico proveniente da questi indirizzi. Purtroppo non sempre una white list è compatibile con i servizi erogati dalla rete (es. si pensi a un sito web pubblico). L’utilizzo di una black list, invece, previene connessioni da indirizzi notoriamente pericolosi. Questo metodo è compatibile con la maggior parte dei sistemi, ma poco efficace data la difficoltà di avere una lista accurata degli indirizzi usati dagli attaccanti. Infine, se il sistema utilizza una architettura scalabile, potrà supportare picchi di traffico maggiore (es. grazie a logiche di load balancing) e avere una alta resilienza, cioè tempi molto rapidi di ripristino.

Approfondimento 2: prevenire injection

- Gli attacchi di injection avvengono tramite l'invio, da parte dell'attaccante, di un messaggio contenente il codice malevolo (payload). Le tecniche più comuni per evitare a questo tipo di attacchi di raggiungere la vittima sono basate sul riconoscimento e sulla sanitizzazione.
- Le tecniche di riconoscimento possono essere specifiche o generiche. Le tecniche specifiche traggono vantaggio dalla conoscenza di dominio degli input attesi dal sistema. Per esempio, regole sintattiche possono essere definite per filtrare il traffico in entrata. Invece, le tecniche generiche mirano a riconoscere i payload notoriamente pericolosi. Anche questi possono essere riconosciuti tramite regole sintattiche o black list. Le operazioni di riconoscimento dei payload sono tipicamente affidate ai Web Application Firewall (WAF), cioè sistemi che analizzano e filtrano il traffico di rete a livello applicativo
- La sanitizzazione degli input viene tipicamente implementata tramite funzioni built-in nei framework di programmazione web. La sanitizzazione avviene, per esempio, sostituendo alcuni caratteri all'interno di un input.

Approfondimento 3: altre forme di XSS

- Il Cross-site scripting è ancora una delle principali vulnerabilità in termini di impatto e di diffusione. Uno dei motivi è che esistono diverse varianti dell'XSS. Ogni variante ha sempre lo scopo di iniettare codice nelle pagine web, ma sfutta tecniche diverse anche interagnedo conl la logica del funzionamento dalla applicazione target.
- Lo **Stored XSS**, per esempio, sfrutta il comportamento di alcune applicazioni web che memorizzano alcuni input degli utenti e li utilizzano nella generazione delle pagine. Un esempio comune sono I campi dedicati ai commenti tipici di forum e blog online. In questo caso, l'injection consiste nell'inviare il payload malevolo al server in modo che questo venga inserito nelle pagine inviate agli utenti dell'applicazione web. In questo modo si ottengono due risultati: tutti gli utenti legittimi vengono attaccati tramite il payload e l'attacco persiste nel tempo (fino a quando non viene rimosso).
- **Reflected XSS** si riferisce invece all'utilizzo di applicazioni web iniettabili per colpire una vittima tramite un link malevolo (ad esempio consegnato tramite mail di phishing). Il link contiene un payload che sfrutta la vulnerabilità e quando viene attivato la vittima accede alla pagina iniettata.

Approfondimento 4: blind SQL injection

- In molti casi un attacco che sfrutta SQL injection ha lo scopo di esfiltrare dati dall'applicazione bersaglio. Per esempio, un attaccante può essere interessato a enumerare gli utenti registrati nel sistema. Raramente però una applicazione web vulnerabile a SQL injection restituisce direttamente I dati presenti sul database. Ad esempio, se il database degli utenti è utilizzato per verificare le credenziali di accesso, il suo contenuto non viene tipicamente mostrato a video.
- Per aggirare questa limitazione è possibile ricorrere a **blind SQLi**, cioè attacchi di injection che mirano a estrarre informazione, un bit alla volta, osservando le risposte generate dall'applicazione web. Per poter funzionare, è necessario che l'attaccante possa sottomettere una query che esegue un test sul database e che sia in gradp di distinguerne il risultato (positivo o negativo) dal compostamento della applicazione. Per esempi, l'applicazione potrebbe restituire un errore (e.g., 404 not found) quando la query fallisce. In questi casi, l'attaccante può implementare un algoritmo di ricerca componendo I test (e.g. "esiste un utente il cui nome inizia per 'A'?").
- Lo stesso meccanismo si può applicare se il test genera effetti laterali osservabili. Per esempi, se il fallimento genera un ritardo nella risposta, si parla di **time-based blind SQLi**.