Contents

1 Cos'è l'hacking

Bisogna innanzitutto fare distinzione fra DDoS ed Hacking:

- DDoS nega un servizio ad una qualche società/compagna, esaurendone le risorse. Il modo più gettonato è eseguire un elevato numero di connessioni verso i server che offrono i servizi, in modo da interrompere il servizio. Questo non è però hacking, non ha tecniche interessanti da sfruttare
- Hacking: tecniche per ottenere accesso non autorizzato ala macchina (ma in realtà servirà) con lo scopo finale di proteggere la macchina su cui abbiamo ottenuto l'accesso.
  Accedo come utente root, riporto i passi che mi hanno permesso di accedervi per far si che il cliente che ha richiesto il test possa sistemare la sicurezza

Come possono due entità comunicare in modo sicuro, in modo che ci sia integrità e confidenzialità? Si introducono appositi protocolli e tecniche crittografiche etc... in modo da rendere la sicurezza sicura. Uso TLS e mi sento abbastanza sicuro, in teoria: chi però implementa il protocollo in software può introdurre delle funzionalità nuove che si pensa siano innocue ma poi sono devastanti.
esempio: ricorda l'hearthbleed di openSSL, la funzionalità aggiunta

era la possibilità di hearthbeat per TLS/DTLS. Vulnerabilità affligge tutti i dispositivi che utilizzano la libreria, era possibile leggere tutti i dati all'interno del dispositivo: password, cookies, etc... Non era un malware, ma una vulnerabilità derivante da ciò che si pensava essere una funzionalità. In cosa consisteva la vulnerabilità: buffer overflow, invia più caratteri di quelli necessari andando a pescare contenuti di aree di memoria adiacenti.
Take home message: tutte le vulnerabilità derivano dal fatto che l'utente può inserire delle informazioni in un sistema informatico che possono:

- avere valore sbagliato
- avere dimensione sbagliata
- input può essere da utente autorizzato ma malevolo
- input da utente non autorizzato, sia malevolo che non

Problema è che input di utenti in generale vanno sempre validati in forma e contenuto, le funzioni di sicurezza non devono mai basarsi su input non validato. Questa cosa va fatta nelle sezioni più critiche del sistema: se server avesse controllato la lunghezza della parola data con quella fornita dall'utente, non ci sarebbe stato problema.
In linea di principio è complicato avere tutto sott'occhio, specialmente perché le configurazioni sono fatte da persone. La teoria non è uguale al mondo reale: un protocollo che in teoria funziona bene, può essere implementato male.

1.1 Hacker vs penetration testers

C'è una delineazione molto chiara e generale su quelle che sono le figure, con annessa legalità/illegalità. "Cappelli": figura che si è sviluppata negli anni:

- script kiddies: prendono programmi dalla rete e li utilizzano per attaccare le reti per "farsi un nome"

- attivisti: motivati da scopi politici o religiosi che effettuano attacchi informatici con questi scopi
- white hat hackers: i "buoni", operavano comunque nell'illegalità. Esperti di sicurezza, il cui scopo era di dimostrare vulnerabilità di sistemi affinché queste venissero fixate.
- black hat hackers: operano nell'illegalità al fine di ottenere accesso non autorizzato per interessi personali: ottenere un nome nella comunità, soldi, vendetta, cerare bot-net etc... Ne fanno parte:
  - gruppi sponsorizzati dallo stato
  - terroristi
  - spie

Tutto ciò visto fin ora opera nell'illegalità, i penetration tester invece no: il pentester ha l'autorizzazione da parte del cliente, mentre l'hacker no. Lo scopo dei pentester è quello di aumentare la consapevolezza in ambito di sicurezza, ottenendo anche permessi per fare test.
Non esistevano questi confini legali, ad esempio nel caso dei white hat, se qualcuno trova una vulnerabilità in un'azienda è come dirglielo: potrebbe farci causa, ignorarla etc... C'è sempre stata un "area grigia", oggi si è quasi arrivati ad un punto fisso che è quello dei bug bounties: viene dato premio in denaro dall'azienda a chi trova la vulnerabilità. Vengono dati dei limiti entro cui poter agire, se si trova la vulnerabilità si ottiene un premio in denaro.

2 Fasi di un penetration test

Lo scopo sarà quello di effettuare un penetration test su un sistema. Il test andrà strutturato, ci sono delle fasi prestabilite, lo scopo del test è ottenere accesso come utente con i permessi più elevati nel sistema. Il test potrebbe essere automatizzato, oppure essere fatto

a mano: alcuni tool rendono veloci degli steps, ma altri step vanno fatti a mano (occorre pensare in modo creativo). 4 macro-step:

- gathering information
- identificare possibili entry point
- tentativo di accesso
- report di ciò che è stato trovato

Il vulnerability assessment può essere automatizzato, ci sono dei tool che permettono di farlo ma può essere molto poco affidabile e produce alto rate di falsi positivi: non c'è la certezza che il sistema sia vulnerabile ad una certa vulnerabilità.
Invece, il penetration test ha un'accuratezza molto alta e produce un risultato binario: successo o insuccesso, quindi o il sistema non è sicuro o potrebbe essere sicuro.
Gli ambiti del penetration test sono vari:

- target recon: sfruttare software vulnerabile
- social engineering: sfruttare interazione con le persone per ottenere informazioni riservate
- physical facilities adult
- ...

Un penetration test può portare a risultati che un semplice vulenaribility assessment non può, l'azienda può sistemare tutte le vulnerabilità trovate (anche in termini di persone). Nel test è fornita l'autorizzazione, ma nel contratto stipulato con l'azienda poterebbe essere possibile non accedere a determinate parti del sistema. Nel caso di un attacco hacker, le fasi sono le stesse del pentest ma in più ci sono fasi di mantenimento di acceso (dopo averlo ottenuto) e di copertura tracce.

2.1 The killchain model: APT

Il modello che si usa nel caso di un attacco informatico. Le fasi sono di più, ma ognuna è mappabile su quelle viste nel pentest:

- reconnaissance: information gathering
- weaponization: cerco arma con cui ottenere l'accesso
- delivery: mando payload malevolo per accedere
- exploitation: fase in cui si esegue un exploit
- installation: per mantenere l'accesso, posso installare malware sul PC per raccogliere informazioni nel tempo
- command and control: l'attaccante installa un agent, che comunica con un mio server per riceve comandi al fine di ottenere controllo della macchina. È l'agent che manda pacchetti verso il server e non il viceversa
- exfiltration: esporto informazioni utili dalla macchina

3 Hunder the hood of applications

Cosa accade "dietro le quinte" quando provo ad accedere ad una qualche applicazione che sta nel web: ho il mio client ed il server, di mezzo l'Internet.
Supponiamo di considerare un'applicazione web: per accedere ad un sito web si usa nella maggior parte dei casi uno URL per indicare la risorsa del web a cui accedere.
HTTP: protocollo costituito da messaggi human-readable, è possibile ispezionare i pacchetti di rete che vanno dal mio client verso il server e viceversa.
Lo stesso vale per SMTP e come per HTTP di default non è inclusa alcuna autenticazione (oggi è possibile configurare server SMTP per rifiutare e-mail non autenticate), ma è possibile trovar e alcuni server

in cui è possibile mandare e-mail nascondendo il mittente.
telnet: software che permette il collegamento con un server e l'invio di messaggi, ad esempio posso richiedere una pagina web (vedo il sorgente).
Quello che accadeva qualche tempo fa era la possibilità di mandare mail senza specificare il mittente (no auth).

4 Linux overview - privilegi e comandi

Permessi ad ogni file o directory di Linux è associato un utente proprietario che avrà determinati privilegi di lettura, scrittura ed esecuzione su questo file, inoltre ci saranno dei privilegi per il gruppo e per gli others.
I permessi vengono visti come dei bit ed è possibile cambiarli con il comando chmod:

- convertendo i bit in base 10 (101 = 4 0 2) e facendo la somma, si ottiene un valore che è possibile assegnare a user, group ed others (ad esempio chmod 666 <file>)
- usando i flag "ugo" (user, group, owner), con il "+" o "-" a seconda se si vuole aggiungere o togliere il privilegio, ed il privilegi/o (rwe ad esempio)

Sudoers la lista degli utenti nel sistema si trova nel file /ect/passwd e con il comando id <utente> è possibile avere informazioni ulteriori sullo specifico utente.
Con il comando su è possibile effettuare il log in con un altro utente, ed è anche possibile entrare come root. *L'utente di root è pericolosissimo*: può eseguire qualsiasi comando senza richiedere password e senza ottenere warning. Con il comando sudo è possibile impersonare altri utenti, utilizzando la password dell'utente corrente, la configurazione del programma è nel file /etc/sudoers, che se mail configurato può portare ad avere gravi vulnerabilità: avendo ad esempio un

utente generico con configurazione ALL=(root) NOPASSWD /bin/cat * è possibile per l'utente eseguire il programma cat seguito da qualsiasi altra cosa, in quando la wildcard "*" può essere sostituita con qualsiasi altra stringa, senza necessità di password e come utente root.

SETUID/SETGID i due flag SETUID/SETGID, se impostati, permettono di cambiare l'esecuzione di un file, e quindi del relativo processo che viene creato, rispettivamente all'utente proprietario del file o al gruppo. Questo cambio di associazione può avvenire senza necessità di password, quindi anche qui è possibile avere gravi conseguenze nel caso in cui il processo fosse vulnerabile (ad esempio ad attacchi di tipo hearthbleed / stack o heap overflow). Per poter settare i bit, è necessario settare il bit s con chmod, in questo modo sarà possibile cambiare il proprietario o il gruppo del processo (perché non funziona di default). Per poter scoprire quali file permettono di cercare file con bit attivi, mediante il comando find + flags.

Mount comando per montare/smontare partizioni e vedere le partizioni disponibili. Su Linux qualunque cosa è un file, per vedere quali sono le partizioni si può controllare il file /etc/fstab, contiene le partizioni da montare di default. Utilità nel pentest: riesco ad ottenere l'accesso ad una macchina terza, per fare l'enumeration una volta avuto l'accesso nella macchina, ossia per poter ottenere più possibili informazioni sulla macchina, col mount possiamo avere informazione sui vari dischi, su cui poi andare a cercare i file.

Compression il formato comune è .tar, l'obietto nel pentest è quello di dare la possibilità di raccogliere le informazioni insieme in modo da poterli portare sulla propria macchina per analizzarli. È possibile passare alcuni flag, la cartella in cui comprimere e cosa andare a comprimere.

/usr/share/wordlists contiene un tar.gz chiamato rockyou, che contiene circa 15M di password risultate da un dataleak.

Processes la lista dei processi è visibile a tutti gli utenti[1], con ps otteniamo una tabella in cui abbiamo diverse informazioni utili

Cron ogni Sistema Operativo è dotato di un job scheduler gestito dall'utente, i comandi o processi di cron vengono schedulati dal sistema in base a determinate impostazioni, definite in file appositi, il file principale è /etc/crontab: è editabile solo da root, ma leggibile da tutti gli utenti del sistema. È utile in quanto spesso, in caso di pentesting o CTF, potremmo ritrovarci nel caso in cui gli admin hanno messo degli specifici comandi nel crontab, in modo da poterli sfruttare.

SSH usando coppia di pub/pr key (RSA) può essere possibile sfruttare delle mal configurazioni per poter ottenere la chiave privata, nel caso in cui non sia stata salvata correttamente. SSH è un tool molto versatile, che permettono oltre che amministrare un server, usare la macchina target per fare molte altre cose.

[1] anche i non privilegiati