1. GTFOBins

GTFOBins (Get The F**k Out Binaries) è un progetto open-source che raccoglie un elenco di binari preinstallati nei sistemi UNIX che possono essere sfruttati da un attaccante per bypassare le restrizioni di sicurezza e ottenere l'escalation dei privilegi. Questi binari sono generalmente considerati innocui, ma possono essere utilizzati in modi imprevisti per eseguire comandi malevoli, ottenere una shell o accedere a file riservati.

Uso e Funzionalità:

GTFOBins si concentra sui binari che possono essere usati per:

- **Escalation dei privilegi**: Alcuni binari possono essere usati per ottenere i permessi di root o altri privilegi elevati.
- **Bypass dei permessi**: Utilizzo di binari per leggere o modificare file protetti.
- **Esecuzione di comandi remoti**: In alcuni casi, possono essere eseguiti comandi arbitrari o aperte backdoor.
- **Persistenza**: Creare meccanismi per mantenere l'accesso persistente a un sistema compromesso.

Esempio:

Un esempio famoso è l'uso del binario tar, che può essere utilizzato per ottenere una shell interattiva con privilegi elevati in determinati contesti:

bash

Copia codice

tar -cf archive.tar --checkpoint=1 --checkpoint-action=exec=/bin/sh

Come usarlo:

Gli attaccanti e i penetration tester utilizzano **GTFOBins** per trovare metodi di exploit su sistemi con binari male configurati. Puoi visitare il sito ufficiale: GTFOBins per cercare binari specifici e le relative vulnerabilità.

---

2. PwnKit (Polkit Vulnerability - CVE-2021-4034)

PwnKit è una vulnerabilità critica identificata nel pacchetto **Polkit**, un componente chiave nei sistemi Linux che gestisce i privilegi di accesso a livello di sistema. La vulnerabilità è identificata come **CVE-2021-4034** ed è stata scoperta nei primi mesi del 2022.

Descrizione:

Polkit (PolicyKit) è uno strumento che consente di definire e gestire i privilegi concessi ai processi. La vulnerabilità in Polkit riguarda il binario pkexec, un comando che consente agli utenti non privilegiati di eseguire comandi come superutente. La vulnerabilità **PwnKit** permette agli attaccanti locali di ottenere i privilegi di root senza autenticazione.

Dettagli Tecnici:

Il problema è un classico **buffer overflow** o una **race condition**. Attraverso l'uso improprio del binario pkexec, gli attaccanti possono manipolare l'input e causare l'esecuzione di comandi arbitrari con privilegi di root, permettendo l'escalation di privilegi.

Esempio di Exploit:

L'exploit tipico di PwnKit consiste nell'eseguire pkexec senza argomenti o con argomenti malformati, sfruttando il modo in cui gestisce la memoria o le variabili d'ambiente, permettendo agli attaccanti di bypassare le protezioni di sicurezza.

Soluzione:

- Aggiornare polkit alle versioni patchate che risolvono la vulnerabilità.
- Utilizzare soluzioni temporanee come la rimozione di pkexec fino all'aggiornamento del sistema.

Per ulteriori dettagli tecnici, puoi consultare il sito di sicurezza relativo alla vulnerabilità: CVE-2021-4034.

---

3. knockd

knockd è un demone che implementa una tecnica chiamata **port knocking**, utilizzata per migliorare la sicurezza delle connessioni a servizi remoti. Il port knocking è un meccanismo in cui un server firewall blocca completamente le porte, ma le apre temporaneamente se riceve una sequenza specifica di richieste sulle porte chiuse.

Come Funziona:

L'idea è che un utente remoto "bussi" (knock) su una serie predefinita di porte in un certo ordine, inviando pacchetti TCP, UDP o ICMP senza alcun contenuto significativo. Il demone **knockd** in ascolto sul server riconosce questa sequenza come un segnale per aprire una porta, ad esempio la porta 22 (SSH), permettendo così l'accesso.

Configurazione:

Il file di configurazione di knockd definisce le sequenze di porte e i comandi associati. Un esempio di configurazione può sembrare il seguente:

bash

Copia codice

[openSSH]

sequence = 7000,8000,9000

tcpflags = syn

command = /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

In questo caso, quando knockd riceve una sequenza di pacchetti TCP SYN sulle porte 7000, 8000 e 9000, esegue un comando che apre la porta 22 per l'indirizzo IP che ha inviato la sequenza.

Pro e Contro:

- **Pro**:
    - Aggiunge uno strato di sicurezza nascondendo i servizi critici dietro porte chiuse.
    - Previene scansioni automatiche delle porte.
- **Contro**:
    - Il port knocking può essere vulnerabile a replay attack se la sequenza non cambia.
    - Se qualcuno intercetta la sequenza, può facilmente sfruttarla per accedere al sistema.

Utilizzo:

Un attaccante o un amministratore di sistema potrebbe usare knockd per nascondere i servizi sensibili, come un server SSH, da potenziali attacchi automatizzati. Per i penetration tester, la sfida consiste nel rilevare il port knocking, usando strumenti come nmap per tentare di scoprire quali porte possono essere sbloccate.

Riassunto:

1. **GTFOBins**: È una raccolta di binari utilizzabili per eseguire exploit come l'escalation di privilegi o l'accesso a risorse protette sui sistemi UNIX.
2. **PwnKit**: Una vulnerabilità critica nel binario pkexec di Polkit che permette l’escalation di privilegi su sistemi Linux.
3. **knockd**: Un demone per implementare port knocking, una tecnica di sicurezza per aprire porte specifiche in base a una sequenza di richieste.

Conclusioni

In conclusione, gli strumenti e le vulnerabilità che abbiamo esplorato — **GTFOBins**, **PwnKit** e **knockd** — evidenziano l'importanza di una corretta configurazione e gestione della sicurezza nei sistemi operativi. **GTFOBins** dimostra come binari comuni possano essere utilizzati in modo malevolo per ottenere privilegi elevati, rendendo fondamentale il monitoraggio di questi file e l'uso di policy di accesso rigorose. **PwnKit** mette in luce la criticità delle vulnerabilità zero-day nei componenti di sistema come Polkit, che possono avere conseguenze devastanti se non risolte tempestivamente con patch di sicurezza. Infine, **knockd** mostra come tecniche creative come il port knocking possano aumentare la sicurezza nascondendo porte sensibili, anche se la loro efficacia dipende da una corretta implementazione.

Questi strumenti e vulnerabilità dimostrano che la sicurezza è un equilibrio tra protezione attiva, monitoraggio e aggiornamenti regolari. Il miglior approccio rimane una strategia difensiva su più livelli, combinando pratiche di sicurezza rigorose con la consapevolezza delle ultime minacce e tecniche di attacco