A06:2021 - Gestione e Difesa dei Componenti Vulnerabili e Obsoleti

Battiato Simone 1000031017

Contents

1 Identificazione del Problema **2**
1.1 Attacchi di downgrade 2
1.2 Sei vulnerabile se 3

2 Come ci difendiamo? **4**
2.1 Come funziona il Virtual Patching 4
2.2 tool per l'identificazione delle vulnerabilità 5

3 Dimostrazione **6**
3.1 Nmap 6
3.2 Metasploit 7
3.3 Difesa 8
3.4 Retire.js 9

4 Conclusione **10**

5 Bibliografia **10**

1 Identificazione del Problema

Il problema dei componenti vulnerabili e obsoleti è una sfida per la sicurezza e la stabilità dei sistemi informatici. Questi componenti, che possono includere software, librerie, firmware e hardware, rappresentano punti di vulnerabilità che possono essere sfruttati da attaccanti per compromettere i sistemi, in particolare nei sistemi odierni. Le implicazioni di questa vulnerabilità possono essere devastanti e includono:

- Esposizione a vulnerabilità di sicurezza già note e sfruttabili, che potrebbero portare a violazioni dei dati, furto di informazioni sensibili o attacchi DoS (deny of service).
- Rischi di conformità legati a normative e regolamentazioni che richiedono la gestione e l'aggiornamento regolare dei componenti software per garantire la sicurezza e la privacy dei dati.
- Complessità aggiuntiva nella gestione e nella manutenzione del sistema a causa della presenza di componenti eterogenei con esigenze di patching e aggiornamento differenziate.

Per comprendere appieno il problema e affrontarlo in modo efficace, è essenziale esaminare le ricerche precedenti, le best practice e le tendenze attuali riguardanti la gestione dei componenti vulnerabili e obsoleti. Questa analisi può includere:

- Revisione della letteratura accademica e delle pubblicazioni industriali che affrontano il tema della sicurezza dei sistemi informatici e della gestione dei componenti obsoleti.
- Studio delle linee guida e delle raccomandazioni fornite da organismi di standardizzazione e regolatori del settore sulla gestione dei rischi legati ai componenti vulnerabili e obsoleti, come il National Institute of Standards and Technology (NIST).
- Analisi delle tendenze emergenti nel panorama della sicurezza informatica e della gestione dei rischi, inclusi nuovi approcci e strumenti per affrontare la sfida dei componenti obsoleti e vulnerabili.

1.1 Attacchi di downgrade

Gli attacchi di downgrade sono una categoria di attacchi informatici che mirano a compromettere la sicurezza di una comunicazione, di un sistema o di un protocollo, costringendolo ad utilizzare una versione meno sicura o obsoleta di un protocollo o di un algoritmo di cifratura. Questi attacchi sono particolarmente pericolosi perché sfruttano vulnerabilità note presenti nelle versioni precedenti di protocolli di sicurezza. Ecco una panoramica dei principali tipi di attacchi di downgrade:

- **SSL/TLS:** Uno degli esempi più noti è il downgrade degli attacchi SSL/TLS, come l'attacco POODLE (Padding Oracle On Downgraded Legacy Encryption). In questo attacco, un aggressore forza una connessione HTTPS a utilizzare SSL 3.0, che è meno sicuro rispetto alle versioni più recenti di TLS. Una volta che il protocollo meno sicuro è in uso, l'aggressore può sfruttare le sue vulnerabilità per decrittografare le comunicazioni.

1.2 Sei vulnerabile se

- **Mancanza di Conoscenza delle Versioni dei Componenti:** Se un'organizzazione non è consapevole delle versioni di tutti i componenti utilizzati, sia lato client che lato server, potrebbe non essere in grado di individuare le vulnerabilità presenti in tali componenti. Questo può lasciare il sistema aperto a potenziali attacchi che sfruttano le vulnerabilità nelle versioni obsolete o non supportate dei componenti.
- **Componenti Vulnerabili, Non Supportati o Obsoleti:** Utilizzare software obsoleto o non supportato espone l'organizzazione a rischi significativi di sicurezza. Le vulnerabilità non corrette possono essere sfruttate dagli attaccanti per compromettere il sistema e mettere a rischio dati sensibili o interrompere i servizi critici.
- **Manca la Regolare Scansione delle Vulnerabilità:** Se un'organizzazione non esegue regolarmente la scansione per individuare le vulnerabilità nei componenti utilizzati e non è aggiornata con le notifiche di sicurezza, può non essere consapevole delle minacce che il suo sistema affronta. Questo può portare a una gestione reattiva delle minacce anziché a una prevenzione proattiva.
- **Manca l'Aggiornamento Tempestivo dei Componenti:** Ritardare l'applicazione delle patch e degli aggiornamenti per i componenti software può esporre l'organizzazione a rischi evitabili. Le vulnerabilità corrette dalle patch possono essere sfruttate dagli attaccanti se non vengono mitigate tempestivamente, lasciando il sistema aperto a possibili violazioni della sicurezza.
- **Mancata Verifica della Compatibilità dei Componenti:** Se gli sviluppatori non testano attentamente la compatibilità delle librerie aggiornate o patchate con il resto del sistema, potrebbero introdurre nuove vulnerabilità o problemi di stabilità. Questo può compromettere l'integrità e la sicurezza complessiva del sistema.
- **Mancanza di Sicurezza delle Configurazioni dei Componenti:** Configurare correttamente i componenti software è essenziale per ridurre al minimo i rischi di sicurezza associati a vulnerabilità di configurazione errate o non sicure. La mancata sicurezza delle configurazioni può esporre il sistema a una vasta gamma di minacce, comprese le violazioni della sicurezza e le perdite di dati.

2 Come ci difendiamo?

- **Applicazione di regole di firewall su determinate porte inutilizzate:** è importante applicare delle regole di firewall per impedire agli attaccanti di sfruttare delle debolezze che sfruttano vulnerabilità di determinate porte che magari sono pure inutilizzate ma aperte.
- **Rimozione di Dipendenze Inutilizzate:** È fondamentale rimuovere tutte le dipendenze, funzionalità, componenti, file e documentazione non utilizzati. Questo riduce la superficie di attacco e semplifica la gestione complessiva del sistema.
- **Inventario Continuo delle Versioni dei Componenti:** È necessario monitorare costantemente le versioni dei componenti sia lato client che lato server, insieme alle loro dipendenze. Questo può essere fatto utilizzando strumenti come versions, OWASP Dependency Check, retire.js, ecc. È importante anche monitorare fonti come il Common Vulnerability and Exposures (CVE) e il National Vulnerability Database (NVD) per individuare le vulnerabilità nei componenti utilizzati. L'automazione di questo processo attraverso strumenti di analisi della composizione del software è altamente consigliata.
- **Monitoraggio delle Librerie Non Mantenute o Obsolete:** Monitorare costantemente le librerie e i componenti per identificare quelli non mantenuti o che non forniscono patch di sicurezza per le versioni obsolete. Se non è possibile applicare una patch, è consigliabile considerare l'implementazione di una patch virtuale per monitorare, rilevare o proteggere contro la vulnerabilità individuata.
- **Virtual Patching:** La patch virtuale è uno strato di enforcement delle policy di sicurezza che impedisce lo sfruttamento di una vulnerabilità nota senza modificare il codice sorgente dell'applicazione. È utile quando non è possibile applicare subito una patch al codice sorgente, offrendo protezione temporanea e riducendo i rischi.

2.1 Come funziona il Virtual Patching

- **Rilevazione delle Vulnerabilità:** Identificazione delle vulnerabilità esistenti nel sistema o nell'applicazione tramite analisi di sicurezza o segnalazioni di terze parti.
- **Creazione della Patch Virtuale:** Configurazione di regole specifiche che blocchino o filtrino il traffico sospetto o dannoso, indirizzate a mitigare l'effetto della vulnerabilità identificata.
- **Implementazione:** Utilizzo di un Web Application Firewall (WAF), Intrusion Prevention System (IPS) o altri strumenti di sicurezza per applicare queste regole. Questi dispositivi monitorano il traffico in tempo reale

e applicano le regole per prevenire gli attacchi che sfruttano le vulnerabilità.

- **Monitoraggio e Aggiornamento:** Continuo monitoraggio del sistema per assicurarsi che le regole di virtual patching siano efficaci e aggiornamento delle st

  esse in base all'evoluzione delle minacce e delle vulnerabilità.

2.2 tool per l'identificazione delle vulnerabilità

In aggiunta alla valutazione delle metodologie, è essenziale considerare l'uso di strumenti specializzati per l'identificazione delle vulnerabilità. Tra questi:

- **Metasploit:** Una delle suite più conosciute per testare la sicurezza di un sistema, che include una vasta gamma di strumenti per scoprire, sfruttare e correggere vulnerabilità. Sfrutta gli exploit di un determinato software per entrare e sfruttare il sistema della vittima.
- **Nmap:** Uno strumento di scansione di rete che può essere utilizzato per scoprire host e servizi sulla rete, identificare configurazioni di sicurezza errate e individuare potenziali vulnerabilità.
- **Retire.js:** un estensione scaricabile in qualsisi browser per verificare potenziali vulnerabilità legate alla versione del sito web. L'uso di questi strumenti può fornire un supporto prezioso nella fase di identificazione delle vulnerabilità, consentendo una valutazione approfondita della sicurezza del sistema e facilitando l'implementazione di misure di protezione appropriate.

3 Dimostrazione

La mia dimostrazione si baserà su una vulnerabilità di una vecchia versione di ftp chiamata vsftpd in particolare la 2.3.4. l'exploit si chiama vstfpd 234 backdoor exploit che permette di ottenere l'accesso root della macchina vittima. Inanzitutto ci servirà fare scanning della rete per individuare l'ip della vittima da attaccare, sfrutterò nmap per farlo

3.1 Nmap

Nmap, è un potente strumento di scansione delle reti utilizzato per scoprire host e servizi su una rete informatica, creando una "mappa" della rete. È ampiamente utilizzato per la sicurezza della rete e la gestione delle risorse di rete.

principali funzionalità nmap:

- **scansione degli Host** Nmap può scansionare una rete per trovare dispositivi attivi (host). Questo include la rilevazione di dispositivi collegati, come computer, router, server, e qualsiasi altro dispositivo di rete.
- **scansione delle porte** Una delle funzionalità più conosciute di Nmap è la scansione delle porte. Nmap può determinare quali porte su un host sono aperte, chiuse o filtrate. Le porte aperte possono indicare servizi attivi su un host, come server web, server FTP, database, ecc.
- **identificazione dei Servizi** Nmap può tentare di determinare il sistema operativo in esecuzione su un host basandosi sulle risposte del network stack. Questo è chiamato "OS Fingerprinting."
- **Rilevamento del sistema operativo** Nmap può tentare di determinare il sistema operativo in esecuzione su un host basandosi sulle risposte del network stack. Questo è chiamato "OS Fingerprinting."
- **Scansioni avanzate**Nmap supporta diverse tecniche di scansione avanzate, come la scansione SYN, scansione UDP, scansione TCP connect, e molte altre, permettendo agli utenti di scegliere il metodo più adatto alla loro specifica situazione di rete e di sicurezza.

Per questa dimostrazione userò una macchina kali dotata di tool quali metasploit e nmap oltre a tanti altri strumenti utili nel campo della cybersecurity, e una macchina chiamata "metasploitable 2" usata a scopi di hacking etico dotata volontariamente di vulnerabilità e servizi outdated per pentesting. Per iniziare, il comando che userò è nmap "ip" -Pn(ping) -sV In particolare quello che ci restituirà sarà il nome del sistema, le varie porte aperte del sistema vittima insieme alle versioni dei servizi attivi su quelle porte ecco un esempio:

Come detto precedentemente sfrutteremo l'exploit della porta 21 Dopo aver individuato la versione del servizio, ci basterà dirigerci su metasploit

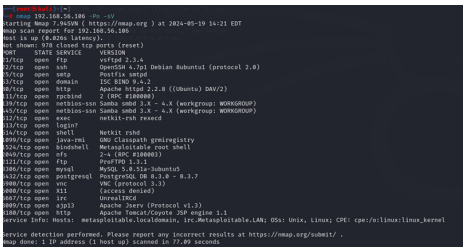

Figure 1: screenshot da nmap

3.2 Metasploit

Nella schermata di benvenuto di metasploit vengono presentate alcune cose tra cui:

- **Exploits**Gli exploit sono codici progettati per sfruttare vulnerabilità specifiche nei sistemi informatici. Questi exploit possono essere utilizzati per sfruttare vulnerabilità note nei sistemi target al fine di ottenere l'accesso non autorizzato o eseguire codice arbitrario.

- **Auxialiary Modules**I moduli ausiliari forniscono funzionalità aggiuntive che possono essere utilizzate durante un test di penetrazione. Questi possono includere scanner di vulnerabilità, strumenti di raccolta di informazioni, strumenti di fuzzing e altro ancora.

- **Post Modules**I moduli post sono progettati per essere eseguiti dopo un exploit o una compromissione riuscita per eseguire azioni specifiche sul sistema target. Questi moduli possono essere utilizzati per eseguire operazioni come la raccolta di informazioni, l'esfiltrazione dei dati, la modifica della configurazione del sistema, ecc.

- **Payloads**: I payloads sono codici che vengono eseguiti dopo che un exploit ha avuto successo e ha compromesso il sistema target. Questi possono includere shell revers, shell meterpreter, script per caricare altri exploit, ecc.

- **Encoders**Gli encoders sono utilizzati per codificare i payload in modo che possano passare attraverso i filtri di sicurezza senza essere rilevati.

- **Evasion Modules**I moduli di evasione sono utilizzati per eludere i sistemi di difesa e le tecnologie di rilevamento delle intrusioni. Questi possono includere tecniche come l'omissione delle firme, l'ingegneria inversa dei filtri, ecc.

Una volta entrati su metasploit ci servirà prima digitare "search vsftpd"

Ci restituira una serie di exploit insieme alla versione sfruttata, e altre informazioni come il rank e il check che ci indica se possiamo verificare se il sistema è

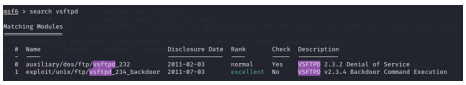

Figure 2: screenshot da Metasploit

vulnerabile senza effettivamente runnare l'exploit. Successivamente per scegliere l'exploit servirà digitare nel nostro caso, "use exploit/unix/ftp/vsftpd 234 backdoor" Ok, ora abbiamo caricato l'exploit, per individuare ciò che possiamo fare basterà digitare show options Ci mostrerà una serie di risultati:

Figure 3: screenshot da Metasploit

CHOST sta per local client address, è tipicamente utilizzato per settare una reverse connection indietro al tuo sistema Ma quello che ci interessa è RHOST ovvero il target in cui innietteremo il nostro exploit Quindi digitiamo "set RHOSTS IP" e una volta fatto ciò inseriremo il payload con "set payload cmd/unix/interact" digitiamo ancora "exploit" e una volta fatto ciò saremo dentro il sistema della vittima come root, per verificare basta digitare un whoaim

3.3 Difesa

Chiaramente la soluzione più semplice per mitigare il problema sarebbe aggiornare il servizio nella nostra macchina vittima però non sempre è l'unica soluzione disponibile, infatti in questo caso è possibile utilizzare un altro espediente per patchare l'exploit di seguito la dimostrazione; inanzitutto ci dovremmo dirigere nella cartella etc, e da li modificare il file vsftpd.conf scendiamo fino a trovare la linea con scritto anon upload enable e disattiviamola inserendo NO:

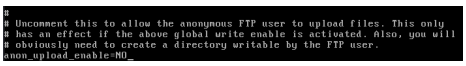

Figure 4: screenshot da bash di linux

Disabilitare l'upload anonimo serve a ridurre la superficie di attacco del server FTP. Quando l'upload anonimo è abilitato, chiunque può caricare file sul server senza autenticazione. Questo può essere sfruttato da attaccanti per: Caricare Malware: Un attaccante potrebbe caricare file malevoli che possono compromettere il sistema. Inondare il Server: Gli upload anonimi possono essere utilizzati per riempire lo spazio su disco, causando un denial of service

(DoS). Eseguire Attacchi di Phishing: File dannosi o di phishing possono essere caricati e serviti dal tuo server, ingannando gli utenti. Disabilitando l'upload anonimo, limiti le azioni che utenti non autenticati possono eseguire sul tuo server, migliorando la sicurezza complessiva. Salviamo il file Questo non basterà perché L'exploit vsftpd 234 backdoor sfrutta una vulnerabilità specifica nella versione 2.3.4 di vsftpd. Questa vulnerabilità è una backdoor inserita intenzionalmente nel software, che permette a un attaccante di ottenere accesso root utilizzando uno specifico username ":)". Quando un attaccante si connette con questo username, il servizio FTP apre una shell di root su una porta predefinita (solitamente la porta 6200). L'attacco non si basa su funzionalità di upload anonimo, ma su una speciale sequenza di login (username ":)") che attiva la backdoor. Questa sequenza è indipendente dalle normali configurazioni di accesso e upload. L'exploit richiede che il sistema apra una connessione su una porta specifica (6200) dopo il login con ":)". Anche se disabiliti l'upload anonimo, la backdoor continuerà a tentare di aprire questa connessione. Quindi oltre a ciò dovremmo anche fare un'altra cosa. Una tecnica per questo particolare tipo di attacco è usare iptables per bloccare le porte inutilizzate. In questo specifico caso la 6200. Per farlo basterà scrivere da linea di comando:

Figure 5: screenshot da bash di linux

Dopo questo runniamo nuovamente l'exploit e potremmo vedere che sta volta, non avrà successo: e nessuna sessione verrà creata

3.4 Retire.js

Un altro "tool" per la prevenzione di attacchi nel caso del web è RETIRE.JS. Retire.js è uno strumento progettato per rilevare l'uso di librerie JavaScript con vulnerabilità conosciute. Il suo scopo è aiutare gli sviluppatori a mantenere sicure le loro applicazioni web identificando versioni di librerie che potrebbero rappresentare un rischio. questo è un esempio di output di retire js su un sito

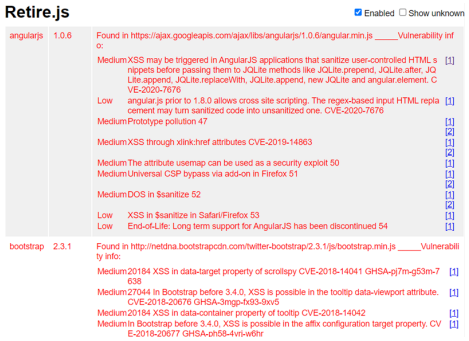

Figure 6: screenshot da bash di linux

4 Conclusione

La gestione dei componenti vulnerabili e obsoleti è una sfida critica per la sicurezza dei sistemi informatici. Attraverso l'adozione di misure proattive come la rimozione delle dipendenze inutilizzate, l'inventario continuo delle versioni dei componenti, il monitoraggio delle librerie non mantenute e l'implementazione della patch virtuale, le organizzazioni possono ridurre significativamente i rischi associati a queste vulnerabilità. È essenziale rimanere aggiornati sulle ultime minacce e tecniche di mitigazione per proteggere efficacemente i sistemi e i dati da potenziali attacchi.

5 Bibliografia

Sito owasp cve
Virtual Patching
retire.js
sito delle cve
sito del nist
cve associata all'attacco al servizio vsftpd