

30/10/2024

Obiettivo:

Lo studente effettuerà un Vulnerability Scanning sulla macchina Metasploitable utilizzando Nessus, concentrandosi sulle porte comuni.

Questo esercizio ha lo scopo di fare pratica con lo strumento Nessus, la configurazione delle scansioni, e di familiarizzare con alcune delle vulnerabilità note.

Fasi dell'Esercizio:

Configurazione della Scansione

- Target: Metasploitable
- Porte: Solo le porte comuni (es. 21, 22, 23, 25, 80, 110, 139, 443, 445, 3389)
- Tipo di Scansione:
 - Basic Network Scan: Configurazione predefinita per una scansione di rete.
 - Advanced Scan: Configurabile in base alle tue esigenze specifiche.

Esecuzione della Scansione

- Avvia la scansione configurata su Nessus.
- Attendi il completamento della scansione e assicurati che tutte le porte specificate siano state analizzate.

Analisi del Report

- Una volta completata la scansione, scarica e analizza il report generato da Nessus.
- Per ogni vulnerabilità riportata:
 - Leggi attentamente la descrizione fornita nel report.
 - Approfondisci ulteriormente utilizzando i link e le risorse suggerite nel report.
 - Cerca ulteriori informazioni sul Web, se necessario.

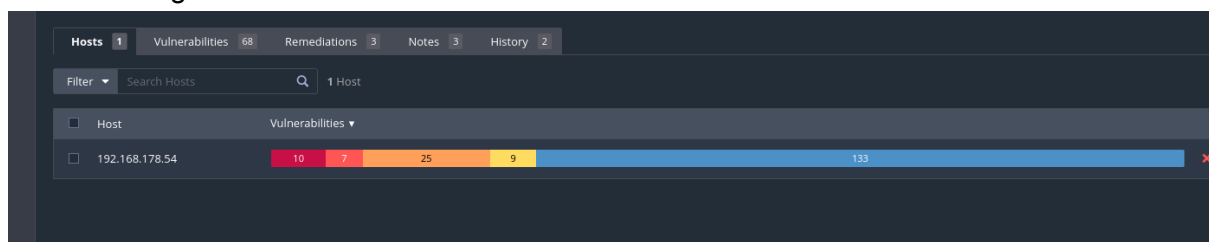
Risultato Atteso

Al termine dell'esercizio, lo studente dovrebbe essere in grado di:

- Configurare e avviare scansioni di vulnerabilità con Nessus.
- Analizzare i report di vulnerabilità e comprendere le informazioni fornite.

Come da richiesta dell'esercizio oggi andremo ad eseguire una scansione della vulnerabilità tramite lo strumento Nessus.

Abbiamo preso come target la MetaSploitable (192.168.178.54) e dopo aver proseguito a settare la scansione (New Scan - Basic Network Scan - compilato form - Launch) abbiamo ottenuto il seguente risultato.



Più nel dettaglio:

| <input type="checkbox"/> | Sev ▼ | CVSS ▼ | VPR ▼ | EPSS ▼ | Name ▲ | Family ▲ | Count ▼ | |
|--------------------------|----------|--------|-------|--------|---|-----------------------|---------|--|
| <input type="checkbox"/> | CRITICAL | 10.0 * | 7.4 | 0.6988 | UnrealIRCd Backdoor Detection | Backdoors | 1 | |
| <input type="checkbox"/> | CRITICAL | 10.0 * | | | VNC Server 'password' Password | Gain a shell remotely | 1 | |
| <input type="checkbox"/> | CRITICAL | 9.8 | | | SSL Version 2 and 3 Protocol Detection | Service detection | 2 | |
| <input type="checkbox"/> | CRITICAL | 9.8 | | | Bind Shell Backdoor Detection | Backdoors | 1 | |
| <input type="checkbox"/> | MIXED | ... | ... | ... | Apache Tomcat (Multiple Issues) | Web Servers | 4 | |
| <input type="checkbox"/> | CRITICAL | ... | ... | ... | SSL (Multiple Issues) | Gain a shell remotely | 3 | |
| <input type="checkbox"/> | HIGH | 7.5 | 5.9 | 0.0358 | Samba Badlock Vulnerability | General | 1 | |
| <input type="checkbox"/> | HIGH | 7.5 * | 5.9 | 0.015 | rlogin Service Detection | Service detection | 1 | |
| <input type="checkbox"/> | HIGH | 7.5 * | 5.9 | 0.015 | rsh Service Detection | Service detection | 1 | |
| <input type="checkbox"/> | HIGH | 7.5 | | | NFS Shares World Readable | RPC | 1 | |
| <input type="checkbox"/> | MIXED | ... | ... | ... | SSL (Multiple Issues) | General | 28 | |
| <input type="checkbox"/> | MIXED | ... | ... | ... | ISC Bind (Multiple Issues) | DNS | 5 | |
| <input type="checkbox"/> | MEDIUM | 6.5 | | | TLS Version 1.0 Protocol Detection | Service detection | 2 | |
| <input type="checkbox"/> | MEDIUM | 6.5 | | | Unencrypted Telnet Server | Misc. | 1 | |
| <input type="checkbox"/> | MEDIUM | 5.9 | 4.4 | 0.9524 | SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption) | Misc | 1 | |

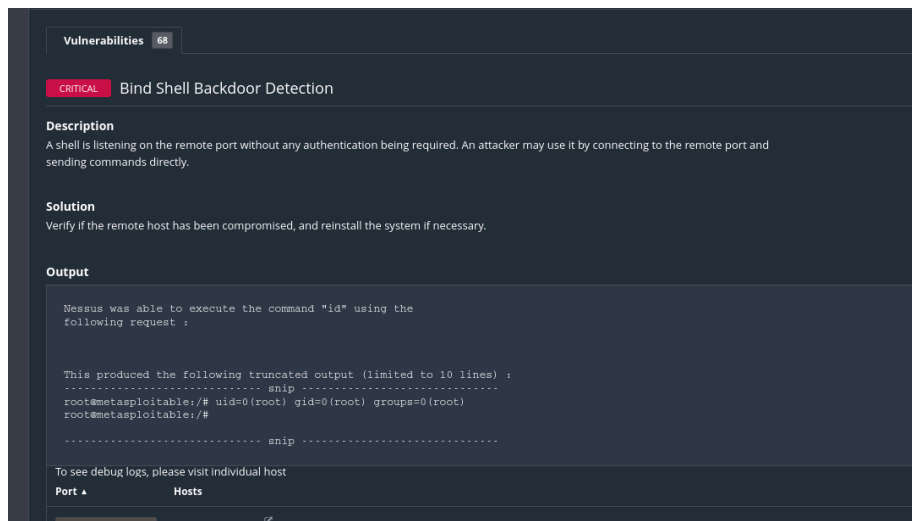
Come da impostazione di Nessus le vulnerabilità sono divise in:

- Rosso - CRITICAL
- Arancione scuro - HIGH
- Arancione chiaro - MEDIUM
- Giallo - LOW
- Celeste - INFO

Risulta normale per qualsiasi dispositivo avere delle vulnerabilità perchè per essere collegato alla rete bisogna per forza lasciare alcune porte aperte, il che equivale ad essere vulnerabile.

Noi oggi andiamo ad analizzare la Metasploitable che, a base didattica, è ricca di vulnerabilità.

Alcune tra quelle che abbiamo riscontrato possono essere:



In questo caso Nessus ha rivelato la presenza di una Backdoor (porta di ingresso secondaria) su una Bind Shell (è una porta che “resta in ascolto”). Da una ricerca su Chatgpt ho ottenuto i seguenti approfondimenti:

Una **bind shell** è un tipo di connessione in cui il sistema compromesso "aspetta" connessioni in entrata su una porta specifica. In una bind shell:

- L'attaccante compromette il sistema e avvia una shell (comando, terminale) su una porta aperta.
- La shell rimane in ascolto su quella porta.
- L'attaccante si connette direttamente a quella porta, ottenendo così accesso al sistema.

Questo tipo di backdoor è chiamato **bind shell** perché il sistema compromesso si "lega" a una porta specifica e aspetta connessioni.

Perché la Rilevazione (Detection) è Importante?

La **rilevazione della bind shell** è importante in quanto rappresenta un segnale di compromissione del sistema:

- Una **bind shell** può essere usata come backdoor per bypassare i controlli di sicurezza e ottenere accesso non autorizzato.
- Le bind shell sono solitamente configurate su porte non standard per evitare rilevamento, ma possono anche essere configurate su porte comuni (ad esempio la porta 80) per mascherarsi.

Come viene Eseguita la Rilevazione?

La rilevazione di una bind shell può essere effettuata in vari modi:

1. **Port Scanning:** Utilizzando strumenti come **Nmap**, puoi scansionare le porte aperte e cercare quelle in ascolto che non dovrebbero esserlo. Le bind shell sono spesso configurate su porte insolite o non utilizzate.

2. **Sniffing del Traffico:** Utilizzando strumenti come **Wireshark** o **Tcpdump** per monitorare il traffico di rete e individuare connessioni sospette su porte specifiche.
3. **File Integrity Monitoring:** Verificando la presenza di file sconosciuti o cambiamenti nelle configurazioni del sistema che possano indicare una bind shell in esecuzione.
4. **Log Monitoring:** Analizzare i log di sistema e di rete per rilevare connessioni in entrata su porte non autorizzate.
5. **Strumenti di Rilevazione di Malware e IDS/IPS:** Software di rilevazione delle intrusioni (come Snort o Suricata) sono in grado di rilevare il comportamento anomalo associato alle bind shell.

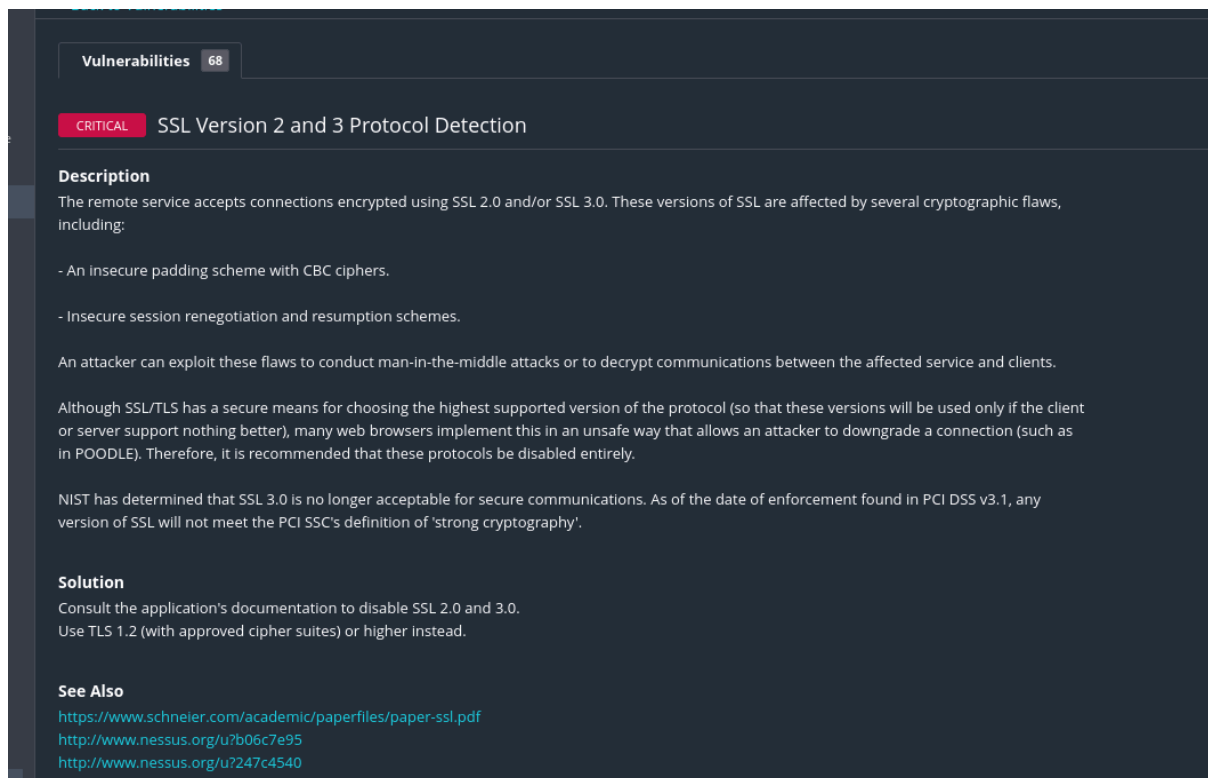
Differenza tra Bind Shell e Reverse Shell

- In una **bind shell**, il sistema compromesso apre una porta e aspetta che l'attaccante si connetta.
- In una **reverse shell**, il sistema compromesso inizia la connessione verso un sistema remoto controllato dall'attaccante. Questo è spesso utilizzato per bypassare firewall e restrizioni di rete.

Conclusione

Il **Bind Shell Backdoor Detection** è quindi un'attività fondamentale di sicurezza per individuare connessioni sospette che potrebbero essere state impostate da un attaccante per mantenere l'accesso al sistema.

Altra vulnerabilità rilevata:



The screenshot shows a Nessus vulnerability report. At the top, there's a header 'Vulnerabilities' with a count of '68'. Below it, a red box labeled 'CRITICAL' highlights the title 'SSL Version 2 and 3 Protocol Detection'. The 'Description' section explains that the remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0, which are affected by several cryptographic flaws, including insecure padding schemes and session renegotiation. It notes that an attacker can exploit these flaws for man-in-the-middle attacks or decryption. The 'Solution' section advises consulting the application's documentation to disable SSL 2.0 and 3.0, and using TLS 1.2 or higher instead. The 'See Also' section provides links to related resources, including a PDF from Schneier.com and Nessus.org.

Vulnerabilities 68

CRITICAL SSL Version 2 and 3 Protocol Detection

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.
- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

Solution

Consult the application's documentation to disable SSL 2.0 and 3.0. Use TLS 1.2 (with approved cipher suites) or higher instead.

See Also

- <https://www.schneier.com/academic/paperfiles/paper-ssl.pdf>
- <http://www.nessus.org/u?b06c7e95>
- <http://www.nessus.org/u?247c4540>
- <https://www.exploit-db.com/exploits/10000/>

In questo caso la versione del protocollo SSL non è aggiornata e pertanto andrebbe aggiornata all'ultima versione disponibile.

Piccolo memorandum sul protocollo SSL (preso da Chatgpt per comodità) :

Le **SSL** (Secure Sockets Layer) sono un protocollo di sicurezza progettato per proteggere le comunicazioni via Internet. **SSL** è usato principalmente per **crittografare** i dati scambiati tra un client (come un browser web) e un server, rendendo le informazioni illeggibili per chiunque cerchi di intercettarle.

A cosa Servono le SSL?

1. **Protezione della Comunicazione:** SSL cripta i dati scambiati tra client e server, proteggendo informazioni sensibili come password, numeri di carte di credito e dati personali.
2. **Autenticazione dell'Identità:** Un certificato SSL garantisce che il server a cui ci si sta connettendo è quello giusto, prevenendo attacchi di tipo **man-in-the-middle** (intercettazioni).
3. **Integrità dei Dati:** I certificati SSL aiutano a prevenire modifiche non autorizzate ai dati durante il trasferimento.

Come Funzionano le SSL?

1. **Handshake SSL:** Quando un client si connette a un server protetto da SSL, inizia un processo di "handshake" per stabilire una connessione sicura.
2. **Scambio di Chiavi:** Durante l'handshake, il server invia al client il suo certificato SSL (che contiene una chiave pubblica). Il client usa questa chiave per criptare una chiave di sessione che solo il server può decrittare con la propria chiave privata.
3. **Crittografia della Sessione:** Una volta stabilita la chiave di sessione, tutta la comunicazione successiva tra client e server è crittografata, garantendo la riservatezza e l'integrità dei dati.

SSL vs TLS

TLS (Transport Layer Security) è il successore di SSL, creato per correggere alcune vulnerabilità di SSL. Oggi, la maggior parte delle connessioni sicure usa TLS, anche se spesso si parla ancora di SSL per semplicità.

Identificazione di SSL nei Siti Web

Un sito web protetto da SSL/TLS può essere riconosciuto:

- Dalla presenza di **https://** nell'URL.
- Da un'icona di lucchetto nel browser, che indica una connessione sicura.

Tipologie di Certificati SSL

Esistono diversi tipi di certificati SSL, come:

- **Certificati a Dominio Validato (DV):** Offrono crittografia di base e sono i più semplici da ottenere.
- **Certificati a Organizzazione Validata (OV):** Forniscono maggiore autenticazione, verificando anche l'organizzazione proprietaria del dominio.
- **Certificati Extended Validation (EV):** Offrono il più alto livello di autenticazione, spesso con il nome della società mostrato nella barra degli indirizzi.

In sintesi, **SSL** serve a proteggere le comunicazioni via Internet, rendendo sicuri i dati e verificando l'identità del server a cui si sta connettendo.

Altra vulnerabilità rilevata:

Meta / Plugin #61708
[← Back to Vulnerabilities](#)

Vulnerabilities 68

CRITICAL VNC Server 'password' Password

Description
The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution
Secure the VNC service with a strong password.

Output

```
Nessus logged in using a password of "password".
```

To see debug logs, please visit individual host

| Port ▲ | Hosts |
|------------------|----------------------------------|
| 5900 / tcp / vnc | 192.168.178.54 🔗 |

In questo caso è stato rilevato da Nessus che il VNC Server di Metasploitable non ha una password. Ovviamente ci consiglia di mettere in sicurezza lo stesso con una password.

Anche in questo caso ho fatto una breve ricerca su Chatgpt per capire meglio cosa è un VNC Server ed il ruolo che svolge(essendo ancora uno studente ho molto da imparare):

Un **VNC Server** (Virtual Network Computing Server) è un software che permette di condividere lo schermo di un computer e controllarlo da remoto tramite una connessione di rete. Con VNC, un utente può visualizzare e interagire con il desktop di un computer remoto come se fosse davanti a quel dispositivo fisicamente.

Come Funziona VNC?

VNC funziona basandosi sull'architettura **client-server**:

- **VNC Server:** È installato sul computer che deve essere controllato (il computer remoto). Il server cattura gli input (come tastiera e mouse) dal client e invia l'immagine dello schermo al client.
- **VNC Client:** Questo software, chiamato anche **VNC Viewer**, è installato sul computer da cui l'utente desidera controllare la macchina remota. Il client invia i comandi di tastiera e mouse al server e visualizza il desktop remoto in tempo reale.

Caratteristiche di un VNC Server

- **Condivisione dello Schermo:** Consente di vedere il desktop di un computer remoto.
- **Controllo Remoto:** L'utente può interagire con il desktop remoto, eseguendo applicazioni, spostando file, ecc.
- **Accesso Multi-Piattaforma:** VNC è compatibile con diversi sistemi operativi, tra cui Windows, macOS e Linux.
- **Crittografia:** Alcuni VNC Server offrono la crittografia per proteggere i dati trasmessi tra client e server, garantendo maggiore sicurezza.

Esempi di VNC Server

Alcuni software VNC comuni sono:

- **RealVNC**
- **TightVNC**
- **TigerVNC**
- **UltraVNC**

Applicazioni di VNC

- **Supporto tecnico remoto:** Gli amministratori IT possono accedere ai computer degli utenti per fornire assistenza.
- **Accesso remoto personale:** Gli utenti possono accedere ai propri computer da remoto, per esempio da casa o da dispositivi mobili.
- **Collaborazione:** VNC consente a più utenti di vedere e interagire con lo stesso desktop, utile per dimostrazioni e presentazioni.

Differenza tra VNC e RDP

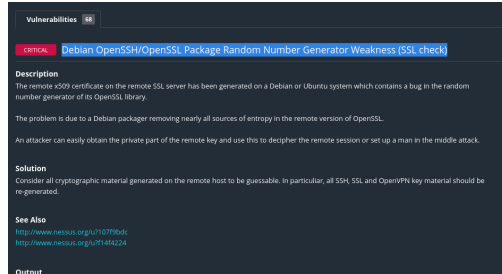
RDP (Remote Desktop Protocol) è una tecnologia simile, ma è specifica di Microsoft e offre prestazioni e funzionalità avanzate per il controllo remoto di sistemi Windows. VNC, invece, è un protocollo più universale e disponibile per molte piattaforme.

Sicurezza di VNC

Sebbene VNC sia molto utile, può esporre i sistemi a rischi se non configurato correttamente. È importante usare password sicure, configurare l'accesso solo per utenti autorizzati e, se possibile, usare una VPN per proteggere ulteriormente la connessione.

In sintesi, un **VNC Server** permette di controllare un computer a distanza, visualizzandone lo schermo e interagendo con esso tramite la rete, ed è uno strumento utile per supporto tecnico, accesso remoto e collaborazione.

Altro problema rilevato:



In questo caso, dopo una ricerca, ho capito che le chiavi di crittografia di OpenSSL/OpenSSH sono vecchie e facilmente hackerabili (infatti nell'header della vulnerabilità rilevata Nessus si dice anche "SSL Check" - che è una vulnerabilità di cui ho parlato in precedenza)

Anche in questo caso, con l'aiuto di Chatgpt mi sono informato meglio su questo problema:

La vulnerabilità **Debian OpenSSH/OpenSSL Package Random Number Generator Weakness** si riferisce a una debolezza nei pacchetti **OpenSSL** e **OpenSSH** distribuiti con alcune versioni di **Debian** e delle sue derivate (come **Ubuntu**) tra il 2006 e il 2008. Questa debolezza ha reso prevedibili i numeri casuali generati durante il processo di creazione delle chiavi di cifratura, compromettendo la sicurezza delle connessioni SSL e SSH su queste piattaforme.

Descrizione della Vulnerabilità

- **Origine del Problema:** Un errore nella configurazione di **OpenSSL** su Debian ha ridotto drasticamente la qualità della generazione di numeri casuali usati per creare le chiavi crittografiche. Ciò è avvenuto a causa della rimozione di alcune righe di codice che, erroneamente, si riteneva potessero causare problemi di sicurezza, ma che invece erano fondamentali per la generazione di numeri casuali sicuri.
- **Conseguenze:** A causa di questa modifica, il generatore di numeri casuali di OpenSSL generava chiavi deboli e prevedibili. Qualsiasi chiave SSH, certificato SSL o altro materiale crittografico creato su un sistema Debian (o derivato) vulnerabile era soggetto a una facile predizione e quindi a possibili attacchi.
- **Impatto:** Tutte le chiavi SSH, SSL/TLS e altre chiavi basate su OpenSSL generate su sistemi Debian vulnerabili erano compromesse. Gli attaccanti potevano facilmente ricostruire le chiavi o intercettare le comunicazioni crittografate.

Versioni Vulnerabili

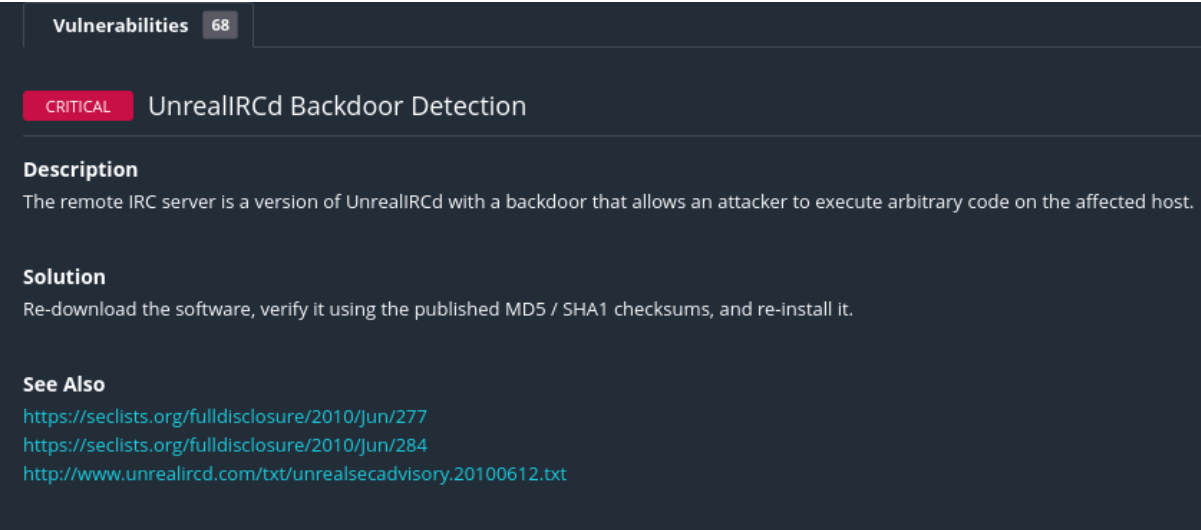
La vulnerabilità ha interessato i sistemi Debian e Ubuntu che utilizzavano versioni dei pacchetti OpenSSL e OpenSSH rilasciate tra il **2006** e il **maggio 2008**. È stata identificata e corretta nelle versioni successive dei pacchetti OpenSSL.

Come Verificare la Presenza della Vulnerabilità (SSL Check)

Per verificare se un sistema o una chiave è stata generata su un sistema vulnerabile, si possono utilizzare i seguenti metodi:

1. **Controllare la Data di Creazione delle Chiavi:** Se le chiavi sono state generate su un sistema Debian/Ubuntu vulnerabile prima del maggio 2008, dovrebbero essere considerate insicure e rigenerate.
2. **Strumenti di Verifica delle Chiavi (Debian-SSL check):** Alcuni strumenti sono stati sviluppati per controllare le chiavi e certificati sospetti, come:
 - **Dowkd.pl:** Un tool che controlla automaticamente se una chiave rientra tra quelle potenzialmente vulnerabili.
 - **Debian-SSL-check:** Uno script che esamina le chiavi esistenti per verificare se sono state generate usando OpenSSL vulnerabile.
3. **Rigenerare le Chiavi Compromesse:** Se le chiavi sono state create su un sistema vulnerabile, è essenziale generare nuove chiavi utilizzando una versione sicura di OpenSSL, per esempio:
bash

Altra vulnerabilità trovata:



The screenshot shows a vulnerability report interface. At the top, there's a header 'Vulnerabilities' with a count of '68'. Below this, a red box labeled 'CRITICAL' is next to the title 'UnrealIRCd Backdoor Detection'. The 'Description' section states: 'The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.' The 'Solution' section says: 'Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.' The 'See Also' section lists three URLs: <https://seclists.org/fulldisclosure/2010/Jun/277>, <https://seclists.org/fulldisclosure/2010/Jun/284>, and <http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt>.

In questo caso il server IRC, server popolare per la Chat ha una backdoor facilmente attaccabile.

Anche in questo caso ho approfondito chiedendo a Chatgpt:

UnrealIRCd Backdoor Detection si riferisce al processo di rilevamento di una backdoor scoperta in una versione compromessa del software **UnrealIRCd**, un popolare server di **Internet Relay Chat (IRC)**. Questa backdoor permetteva agli attaccanti di eseguire comandi arbitrari sui server colpiti con **pieni privilegi**.

Contesto della Vulnerabilità

Nel **2010**, è stata scoperta una backdoor in una versione del software **UnrealIRCd** scaricabile dal sito ufficiale. La backdoor era stata aggiunta da un attaccante che aveva compromesso il pacchetto disponibile per il download, inserendo codice maligno nel software senza che gli sviluppatori di UnrealIRCd ne fossero a conoscenza.

- **Versione compromessa:** La versione **3.2.8.1** di UnrealIRCd, scaricata dal sito ufficiale tra novembre 2009 e giugno 2010, conteneva il codice backdoor.
- **Funzionalità della Backdoor:** La backdoor permetteva a chiunque di eseguire comandi con i privilegi di root (o dell'utente con cui era in esecuzione il server IRC) semplicemente inviando una stringa di comando al server. In pratica, un attaccante poteva ottenere il pieno controllo del server senza bisogno di autenticazione.

Come Funzionava la Backdoor?

L'attaccante poteva attivare la backdoor inviando una stringa specifica. La backdoor era in ascolto su una porta IRC (generalmente la porta 6667) e accettava comandi remoti inviati in un certo formato.

Rilevazione della Backdoor (Backdoor Detection)

Per rilevare la presenza di questa backdoor, gli amministratori di sistema possono utilizzare diversi metodi:

1. **Controllare la Versione del Software:**
 - Verificare se la versione di UnrealIRCd installata è **3.2.8.1** scaricata prima di giugno 2010.
 - Aggiornare a una versione pulita e non compromessa se viene rilevata la versione 3.2.8.1 compromessa.
2. **Controllo dell'Integrità del File Binario:**
 - Confrontare l'hash del file eseguibile di UnrealIRCd con l'hash della versione ufficiale non compromessa. Il team di UnrealIRCd ha rilasciato checksum per aiutare a verificare l'integrità del pacchetto.
 - Esegui un comando come **sha256sum** o **md5sum** sull'eseguibile di UnrealIRCd e confronta l'hash risultante con l'hash fornito dagli sviluppatori.
3. **Monitoraggio delle Attività di Rete:**
 - Monitorare le connessioni sulla porta 6667 (o altre porte IRC configurate) per verificare eventuali comandi non autorizzati o connessioni sospette.
 - Utilizzare strumenti di monitoraggio come **Wireshark** o **Tcpdump** per cercare stringhe sospette (**AB; <comando>**).
4. **Strumenti di Scansione di Malware:**
 - Utilizzare strumenti di rilevamento di rootkit e malware, come **Chkrootkit** o **Rkhunter**, per cercare codice malevolo o modifiche nel sistema.

Mitigazione

1. **Aggiornare UnrealIRCd:** Se rilevi che il server UnrealIRCd è compromesso, scarica immediatamente una versione non compromessa dal sito ufficiale e aggiorna.

2. **Rimuovere le Versioni Compromesse:** Disinstalla qualsiasi versione sospetta e reinstalla il software da fonti verificate.
3. **Monitoraggio e Logging:** Implementa controlli di sicurezza e logging per rilevare attività anomale sui server IRC, come connessioni inaspettate o comandi non autorizzati.

Conclusione

La **UnrealIRCd Backdoor Detection** è fondamentale per prevenire accessi non autorizzati e mantenere la sicurezza dei server IRC. Questa backdoor evidenzia l'importanza di scaricare software solo da fonti sicure, verificare le firme digitali e monitorare costantemente i sistemi per attività sospette.

Conclusioni personali

Nessus è un tool potentissimo nelle mani di un Ethical Hacker e andrebbe sempre utilizzato in maniera costruttiva ed, appunto, etica.

A differenza di Nmap che ci restituisce solo dei dati OGGETTIVI, Nessus ci dà anche delle informazioni SOGGETTIVE.

Esso non si ferma solo al ping ed al TCP scanning, ma riesce anche a darci informazioni sulle app, i sistemi operativi, ed i registri di sistema ed, inoltre, “testa” la vulnerabilità degli stessi con degli exploit e ci propone anche delle soluzioni, come nell'esempio:

'password'. A remote, unauthenticated attacker could

Solution

Secure the VNC service with a strong password.

