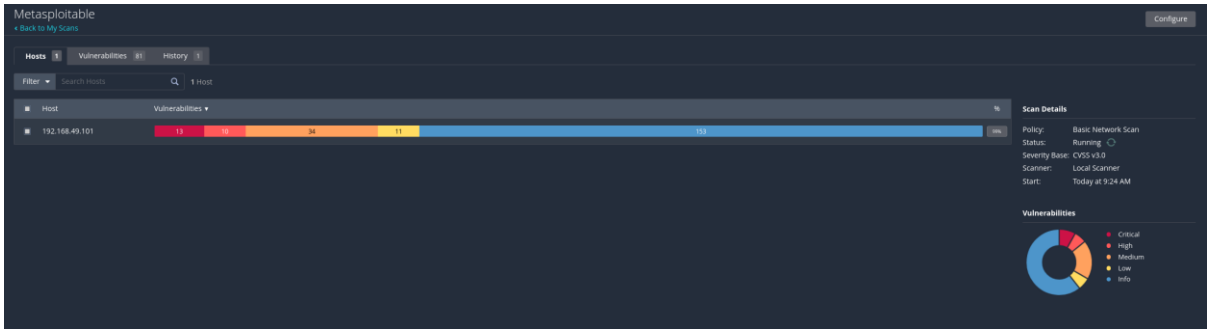
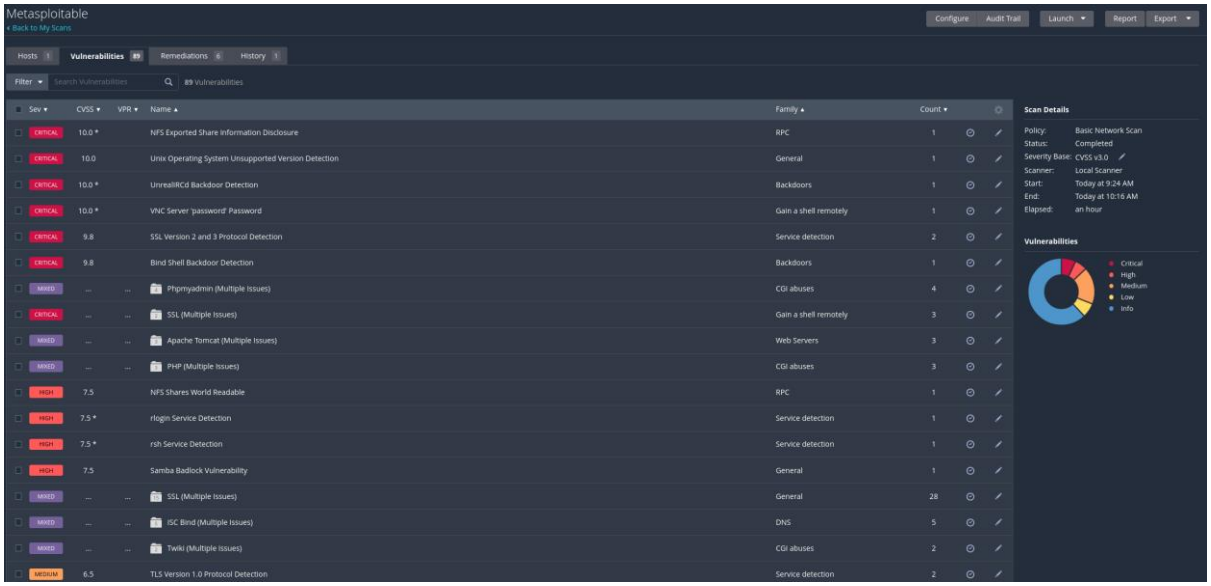
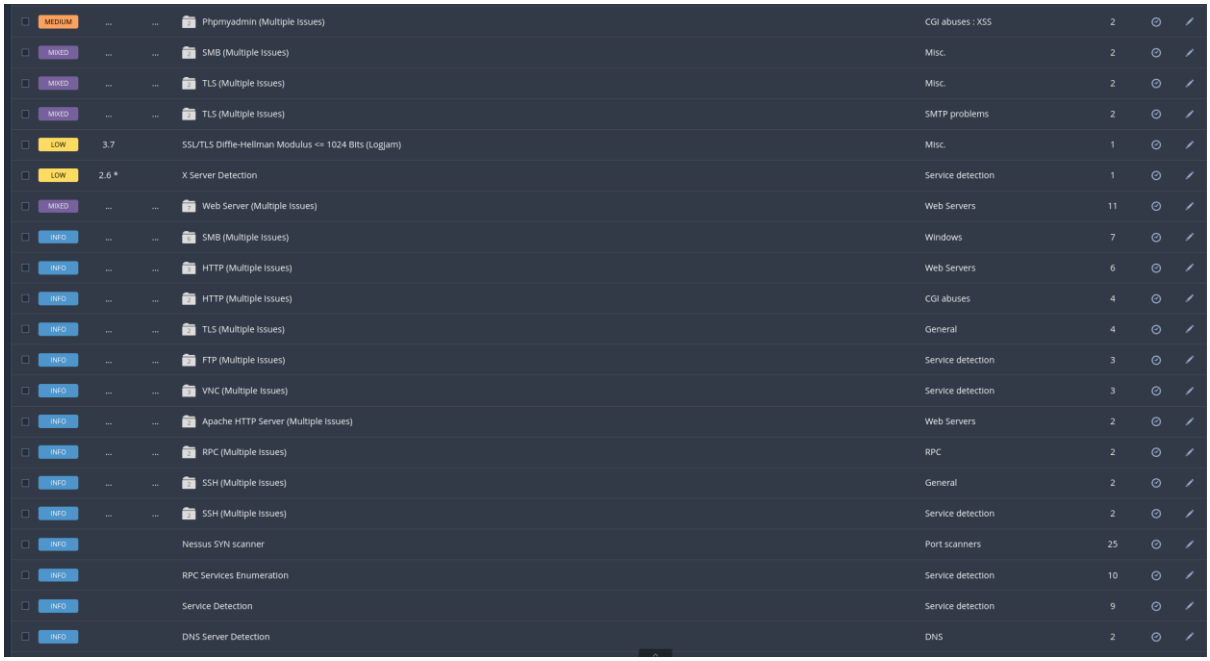
Abbiamo scansionato le vulnerabilità della macchina Metasploitable tramite l'uso di Nessus:



I risultati ci mostrano molte vulnerabilità con vari gradi di importanza.



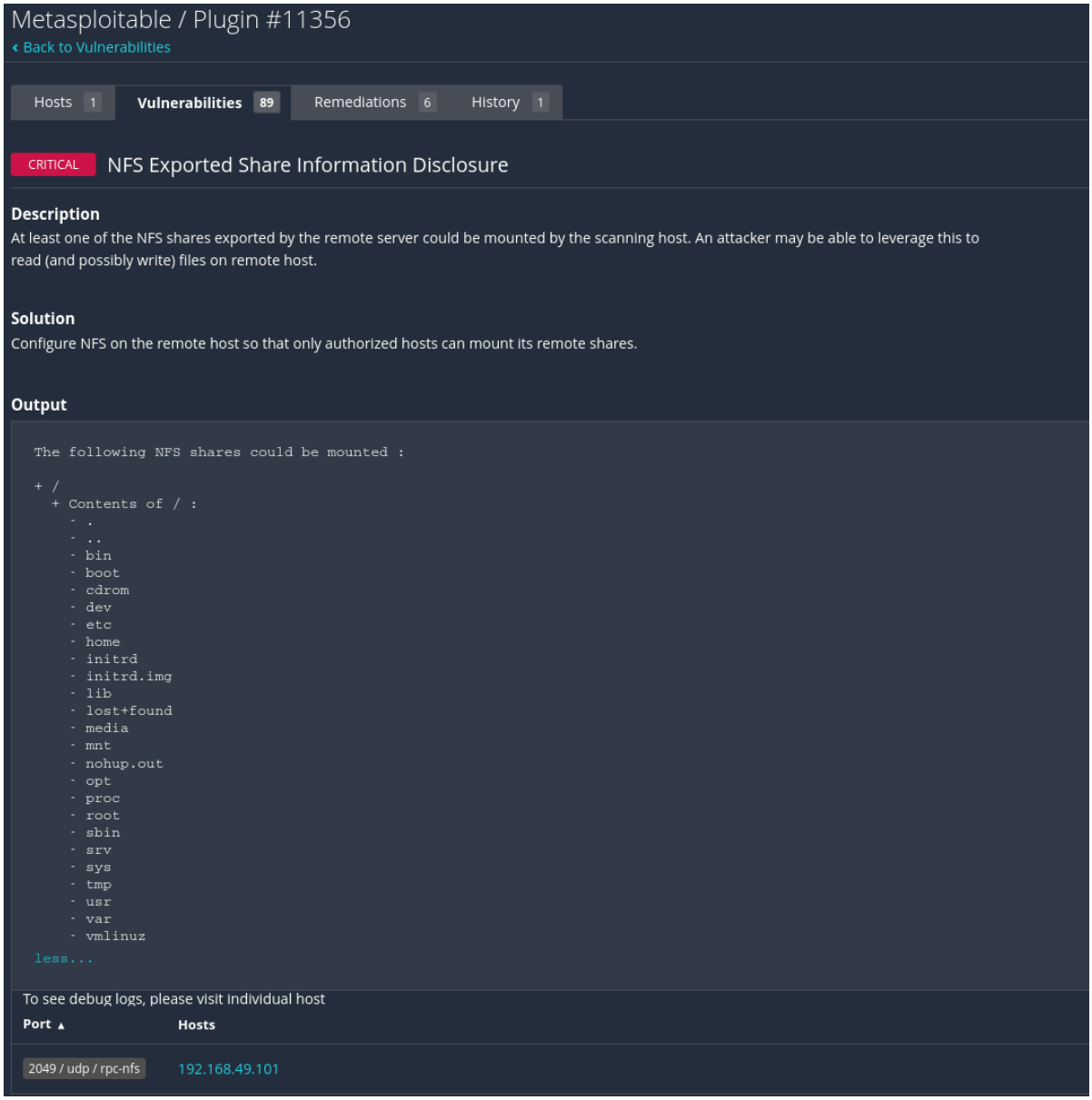


MEDIUM	...	...	Phpmyadmin (Multiple Issues)	CGI abuses : XSS	2		
MIXED	...	...	SMB (Multiple Issues)	Misc.	2		
MIXED	...	...	TLS (Multiple Issues)	Misc.	2		
MIXED	...	...	TLS (Multiple Issues)	SMTP problems	2		
LOW	3.7		SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)	Misc.	1		
LOW	2.6 *		X Server Detection	Service detection	1		
MIXED	...	...	Web Server (Multiple Issues)	Web Servers	11		
INFO	...	...	SMB (Multiple Issues)	Windows	7		
INFO	...	...	HTTP (Multiple Issues)	Web Servers	6		
INFO	...	...	HTTP (Multiple Issues)	CGI abuses	4		
INFO	...	...	TLS (Multiple Issues)	General	4		
INFO	...	...	FTP (Multiple Issues)	Service detection	3		
INFO	...	...	VNC (Multiple Issues)	Service detection	3		
INFO	...	...	Apache HTTP Server (Multiple Issues)	Web Servers	2		
INFO	...	...	RPC (Multiple Issues)	RPC	2		
INFO	...	...	SSH (Multiple Issues)	General	2		
INFO	...	...	SSH (Multiple Issues)	Service detection	2		
INFO			Nessus SYN scanner	Port scanners	25		
INFO			RPC Services Enumeration	Service detection	10		
INFO			Service Detection	Service detection	9		
INFO			DNS Server Detection	DNS	2		

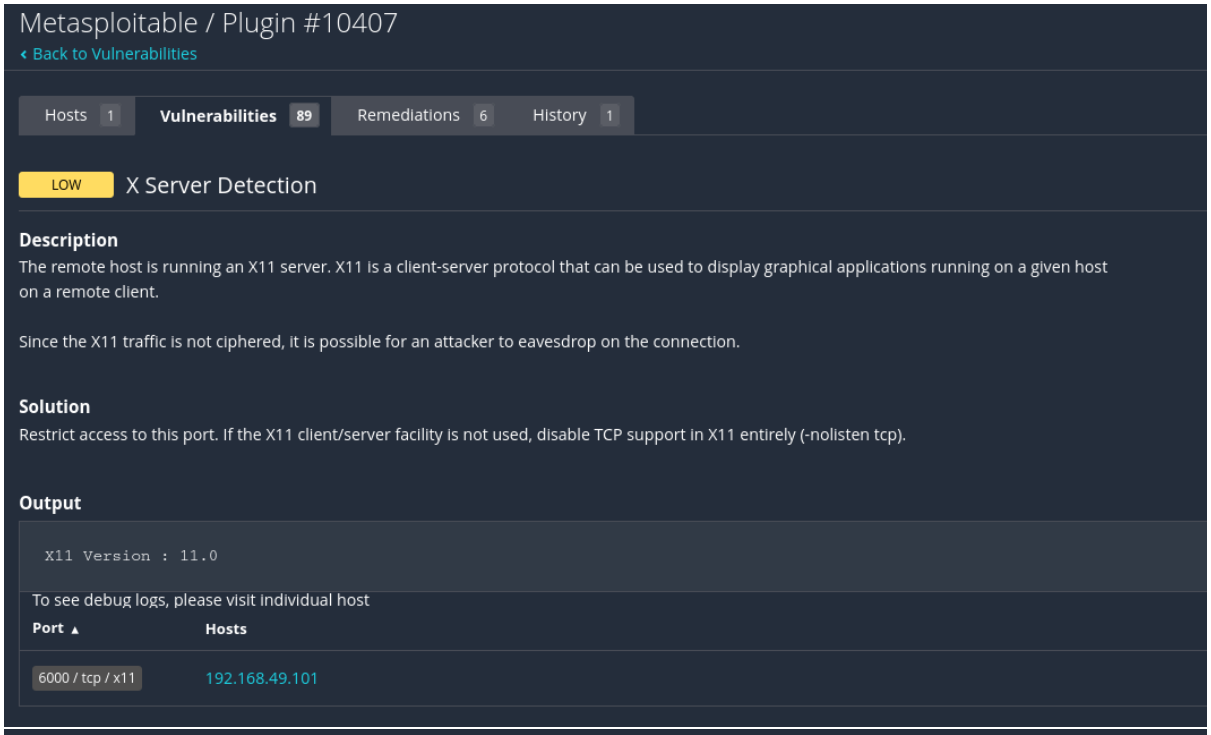
Se ad esempio proviamo ad aprire il log del primo risultato di livello critico Nessus ci mostrerá diverse opzioni, tra cui:
-tempo di scansione;
-host;
-nome vulnerabilità;
-descrizione dettagliata;
-link correlati;
-possibili soluzioni;
-rischi;

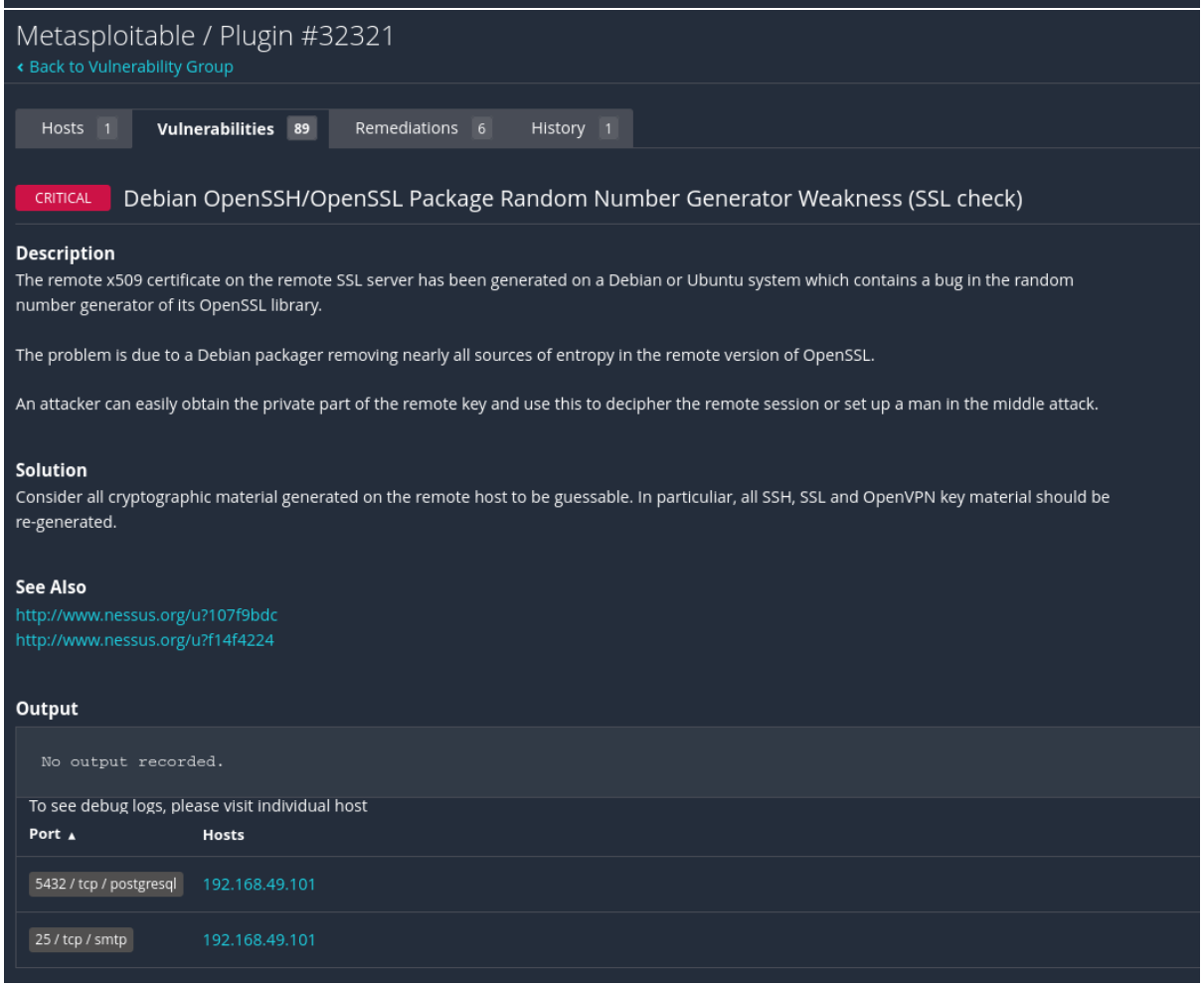
Ogni risultato avrà una soluzione e un log diversi in base alla sua funzione e gravità.

Per eseguire questo tipo di ricerca abbiamo usato il plug in ""basic netwok scan"



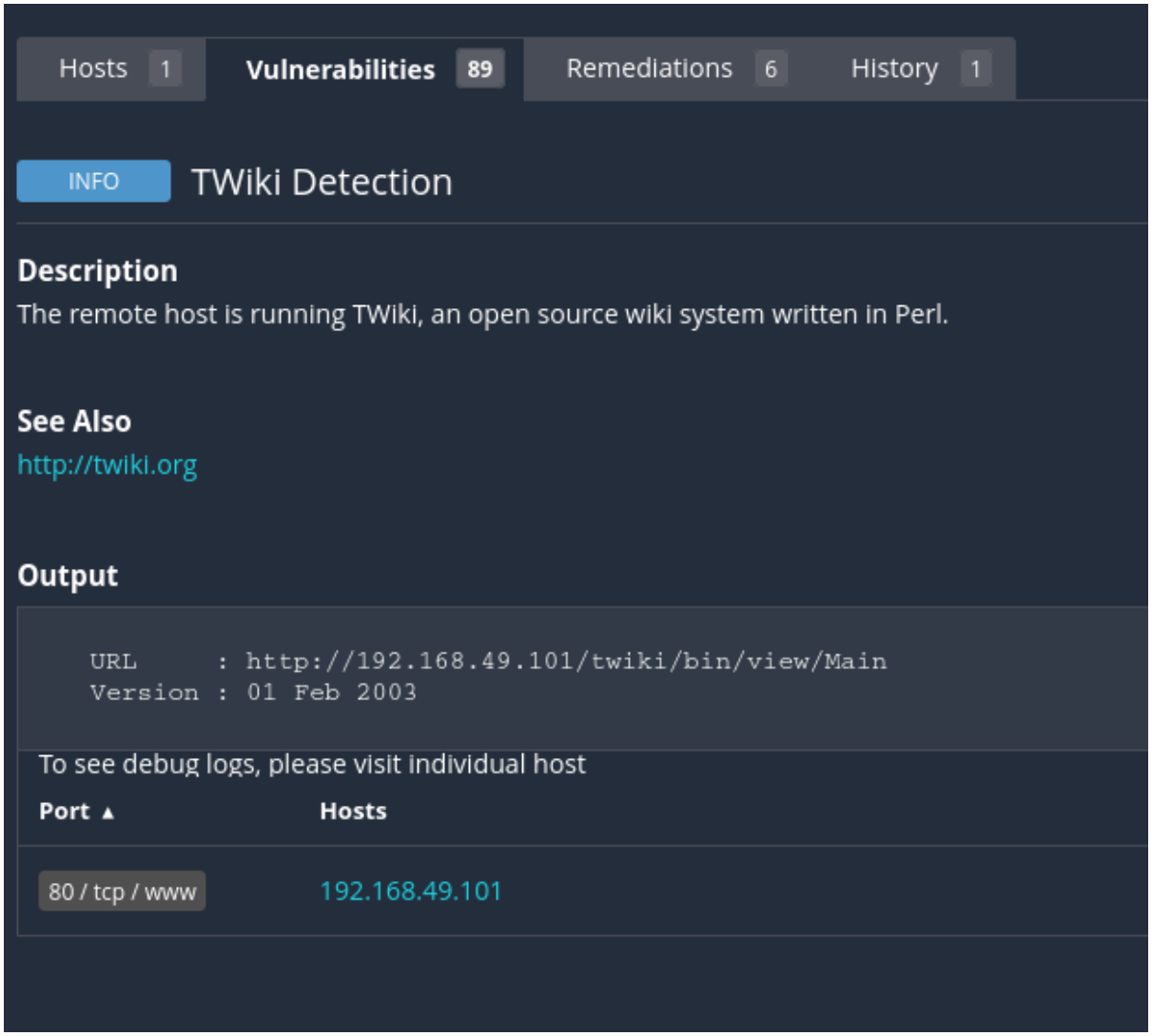
Altri esempi di vulenrabilitá sia critica che bassa:



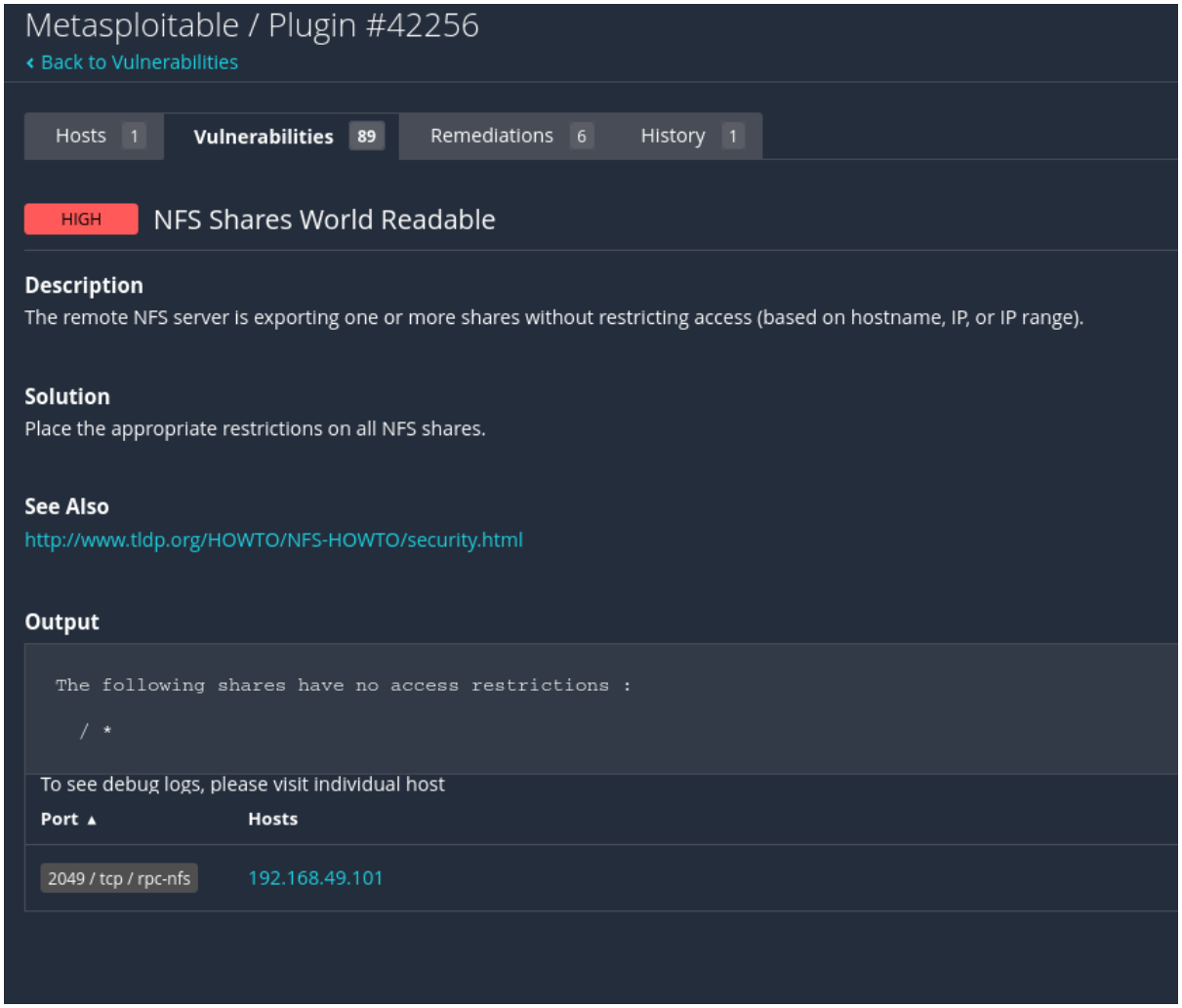


Vengono inoltre considerati anche delle "info" generali di alcuni funzionamenti interni della

macchina che potrebbero in futuro portare a delle possibili vulnerabilità:



Andiamo adesso a studiare e capire una vulnerabilità e a cercare di risolverla:



In questo caso abbiamo scelto un livello "HIGH" secondo il quale il server NFS remoto sta esportando una o piú condivisioni senza limitarne l'accesso (per host e ip). Una soluzione consigliata sarebbe quella di porre delle restrizioni su tutte le azioni NFS inserendo un controllo sul limite massimo di condivisioni contemporanee.

Ci viene inoltre fornito un link per diversi suggerimenti sulla sicurezza del NFS:

6. Security and NFS

This list of security tips and explanations will not make your site completely secure. NOTHING will make your site completely secure. Reading this section may help you get an idea of the security problems with NFS. This is not a comprehensive guide and it will always be undergoing changes. If you have any tips or hints to give us please send them to the HOWTO maintainer.

If you are on a network with no access to the outside world (not even a modem) and you trust all the internal machines and all your users then this section will be of no use to you. However, its our belief that there are relatively few networks in this situation so we would suggest reading this section thoroughly for anyone setting up NFS.

With NFS, there are two steps required for a client to gain access to a file contained in a remote directory on the server. The first step is mount access. Mount access is achieved by the client machine attempting to attach to the server. The security for this is provided by the /etc/exports file. This file lists the names or IP addresses for machines that are allowed to access a share point. If the client's ip address matches one of the entries in the access list then it will be allowed to mount. This is not terribly secure. If someone is capable of spoofing or taking over a trusted address then they can access your mount points. To give a real-world example of this type of "authentication". This is equivalent to someone introducing themselves to you and you believing they are who they claim to be because they are wearing a sticker that says "Hello, My Name is ...". Once the machine has mounted a volume, its operating system will have access to all files on the volume (with the possible exception of those owned by root, see below) and write access to those files as well, if the volume was exported with the rw option.

The second step is file access. This is a function of normal file system access controls on the client and not a specialized function of NFS. Once the drive is mounted the user and group permissions on the files determine access control.

An example: bob on the server maps to the UserID 9999. Bob makes a file on the server that is only accessible the user (the equivalent to typing *chmod 600 filename*). A client is allowed to mount the drive where the file is stored. On the client mary maps to UserID 9999. This means that the client user mary can access bob's file that is marked as only accessible by him. It gets worse: If someone has become superuser on the client machine they can **su -** username and become any user. NFS will be none the wiser.

Its not all terrible. There are a few measures you can take on the server to offset the danger of the clients. We will cover those shortly.

If you don't think the security measures apply to you, you're probably wrong. In Section 6.1 we'll cover securing the portmapper, server and client security in Section 6.2 and Section 6.3 respectively. Finally, in Section 6.4 we'll briefly talk about proper firewalling for your nfs server.

Finally, it is critical that all of your nfs daemons and client programs are current. If you think that a flaw is too recently announced for it to be a problem for you, then you've probably already been compromised.

A good way to keep up to date on security alerts is to subscribe to the bugtraq mailinglists. You can read up on how to subscribe and various other information about bugtraq here: http://www.securityfocus.com/forums/bugtraq/faq.html

Additionally searching for NFS at securityfocus.com's search engine will show you all security reports pertaining to NFS.

You should also regularly check CERT advisories. See the CERT web page at www.cert.org.

6.1. The portmapper

The portmapper keeps a list of what services are running on what ports. This list is used by a connecting machine to see what ports it wants to talk to access certain services.

The portmapper is not in as bad a shape as a few years ago but it is still a point of worry for many sys admins. The portmapper, like NFS and NIS, should not really have connections made to it outside of a trusted local area network. If you have to expose them to the outside world - be careful and keep up diligent monitoring of those systems.

Not all Linux distributions were created equal. Some seemingly up-to-date distributions do not include a securable portmapper. The easy way to check if your portmapper is good or not is to run strings(1) and see if it reads the relevant files, /etc/hosts.deny and /etc/hosts.allow. Assuming your portmapper is /sbin/portmap you can check it with this command:

```
strings /sbin/portmap | grep hosts
```

On a securable machine it comes up something like this:

```
/etc/hosts.allow
/etc/hosts.deny
@(#) hosts_ctl.c 1.4 94/12/28 17:42:27
@(#) hosts_access.c 1.21 97/02/12 02:13:22
```

First we edit /etc/hosts.deny. It should contain the line

```
portmap: ALL
```

which will deny access to everyone. While it is closed run:

```
rpcinfo -p
```

just to check that your portmapper really reads and obeys this file. Rpcinfo should give no output, or possibly an error message. The files /etc/hosts.allow and /etc/hosts.deny take effect immediately after you save them. No daemon needs to be restarted.

Closing the portmapper for everyone is a bit drastic, so we open it again by editing /etc/hosts.allow. But first we need to figure out what to put in it. It should basically list all machines that should have access to your portmapper. On a run of the mill Linux system there are very few machines that need any access for any reason. The portmapper administers **nfsd**, **mountd**, **ypbind/ypserv**, **rquotad**, **lockd** (which shows up as rstatd(), **statd** (which shows up as status) and 'r' services like **rsh/rlogin** and **rusers**. Of those only **nfsd**, **mountd**, **ypbind/ypserv** and perhaps **rquotad**, **lockd** and **statd** are of any consequence. All machines that need to access services on your machine should be allowed to do that. Let's say that your machine's address is *192.168.0.254* and that it lives on the subnet *192.168.0.0*, and that all machines on the subnet should have access to it (for an overview of those terms see the the Networking-Overview-HOWTO). Then we write:

```
portmap: 192.168.0.0/255.255.255.0
```

in /etc/hosts.allow. If you are not sure what your network or netmask are, you can use the **ifconfig** command to determine the netmask and the **netstat** command to determine the network. For, example, for the device eth0 on the above machine **ifconfig** should show: