NFS

Cheat Sheet
Resources
Go to Questions

Network File System (NFS) is a network file system developed by Sun Microsystems and has the same purpose as SMB. Its purpose is to access file systems over a network as if they were local. However, it uses an entirely different protocol. NFS is used between Linux and Unix systems. This means that NFS clients cannot communicate directly with SMB servers. NFS is an Internet standard that governs the procedures in a distributed file system. While NFS protocol version 3.0 (NFSv3), which has been in use for many years, authenticates the client computer, this changes with NFSv4. Here, as with the Windows SMB protocol, the user must authenticate.

Version	Features
NFSv2	It is older but is supported by many systems and was initially operated entirely over UDP.
NFSv3	It has more features, including variable file size and better error reporting, but is not fully compatible with NFSv2 clients.
NFSv4	It includes Kerberos, works through firewalls and on the Internet, no longer requires portmappers, supports ACLs, applies state-based operations, and provides performance improvements and high security. It is also the first version to have a stateful protocol.

NFS version 4.1 (RFC 8881) aims to provide protocol support to leverage cluster server deployments, including the ability to provide scalable parallel access to files distributed across multiple servers (pNFS extension). In addition, NFSv4.1 includes a session trunking mechanism, also known as NFS multipathing. A significant advantage of NFSv4 over its predecessors is that only one UDP or TCP port 2049 is used to run the service, which simplifies the use of the protocol across firewalls.

NFS is based on the Open Network Computing Remote Procedure Call (ONC-RPC/SUN-RPC) protocol exposed on TCP and UDP ports 111, which uses External Data Representation (XDR) for the system-independent exchange of data. The NFS protocol has no mechanism for authentication or authorization. Instead, authentication is completely shifted to the RPC protocol's options. The authorization is taken from the available information of the file system where the server is responsible for translating the user information supplied by the client to that of the file system and converting the corresponding authorization information as correctly as possible into the syntax required by UNIX.

The most common authentication is via UNIX UID/GID and group memberships, which is why this syntax is most likely to be applied to the NFS protocol. One problem is that the client and server do not necessarily have to have the same mappings of UID/GID to users and groups, and the server does not need to do anything further. No further checks can be made on the part of the server. This is why NFS should only be used with this authentication method in trusted networks.

Default Configuration

NFS is not difficult to configure because there are not as many options as FTP or SMB have. The /etc/exports file contains a table of physical filesystems on an NFS server accessible by the clients. The NFS Exports Table shows which options it accepts and thus indicates which options are available to us.

Exports File

Exports File

```
Govardhan Gujji22@htb[/htb]$ cat /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,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
```

The default exports file also contains some examples of configuring NFS shares. First, the folder is specified and made available to others, and then the rights they will have on this NFS share are connected to a host or a subnet. Finally, additional options can be added to the hosts or subnets.

Option	Description
rw	Read and write permissions.
ro	Read only permissions.
sync	Synchronous data transfer. (A bit slower)
async	Asynchronous data transfer. (A bit faster)
secure	Ports above 1024 will not be used.
insecure	Ports above 1024 will be used.
no_subtree_check	This option disables the checking of subdirectory trees.
root_squash	Assigns all permissions to files of root UID/GID 0 to the UID/GID of anonymous.

Let us create such an entry for test purposes and play around with the settings.

ExportFS

ExportFS

```
root@nfs:~# echo '/mnt/nfs  10.129.14.0/24(sync,no_subtree_check)' >> /etc/exports
root@nfs:~# systemctl restart nfs-kernel-server
root@nfs:~# exportfs

/mnt/nfs        10.129.14.0/24
```

We have shared the folder /mnt/nfs to the subnet 10.129.14.0/24 with the setting shown above. This means that all hosts on the network will be able to mount this NFS share and inspect the contents of this folder.

Dangerous Settings

However, even with NFS, some settings can be dangerous for the company and its infrastructure. Here are some of them listed:

Option	Description
rw	Read and write permissions.
insecure	Ports above 1024 will be used.
nohide	If another file system was mounted below an exported directory, this directory is exported by its own exports entry.
no_root_squash	All files created by root are kept with the UID/GID 0.

It is highly recommended to create a local VM and experiment with the settings. We will discover methods that will show us how the NFS server is configured. For this, we can create several folders and assign different options to each one. Then we can inspect them and see what settings can have what effect on the NFS share and its permissions and the enumeration process.

We can take a look at the insecure option. This is dangerous because users can use ports above 1024. The first 1024 ports can only be used by root. This prevents the fact that no users can use sockets above port 1024 for the NFS service and interact with it.

Footprinting the Service

When footprinting NFS, the TCP ports 111 and 2049 are essential. We can also get information about the NFS service and the host via RPC, as shown below in the example.

Nmap

Nmap

```
Govardhan Gujji22@htb[/htb]$ sudo nmap 10.129.14.128 -p111,2049 -sV -sC

Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-19 17:12 CEST
Nmap scan report for 10.129.14.128
Host is up (0.00018s latency).

PORT    STATE SERVICE VERSION
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      41982/udp6  mountd
|   100005  1,2,3      45837/tcp   mountd
|   100005  1,2,3      47217/tcp6  mountd
|   100005  1,2,3      58830/udp   mountd
|   100021  1,3,4      39542/udp   nlockmgr
|   100021  1,3,4      44629/tcp   nlockmgr
|   100021  1,3,4      45273/tcp6  nlockmgr
|   100021  1,3,4      47524/udp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp open  nfs_acl 3 (RPC #100227)
MAC Address: 00:00:00:00:00:00 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.58 seconds
```

The rpcinfo NSE script retrieves a list of all currently running RPC services, their names and descriptions, and the ports they use. This lets us check whether the target share is connected to the network on all required ports. Also, for NFS, Nmap has some NSE scripts that can be used for the scans. These can then show us, for example, the contents of the share and its stats.

Nmap

```
Govardhan Gujji22@htb[/htb]$ sudo nmap --script nfs* 10.129.14.128 -sV -p111,2049

Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-19 17:37 CEST
Nmap scan report for 10.129.14.128
Host is up (0.00021s latency).

PORT     STATE SERVICE VERSION
111/tcp  open  rpcbind 2-4 (RPC #100000)
| nfs-ls: Volume /mnt/nfs
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID    GID    SIZE  TIME                 FILENAME
| rwxrwxrwx   65534  65534  4096  2021-09-19T15:28:17  .
| ??????????  ?      ?      ?     ?                    ..
| rw-r--r--   0      0      1872  2021-09-19T15:27:42  id_rsa
| rw-r--r--   0      0      348   2021-09-19T15:28:17  id_rsa.pub
| rw-r--r--   0      0      0     2021-09-19T15:22:30  nfs.share
|_
| nfs-showmount:
|_  /mnt/nfs 10.129.14.0/24
| nfs-statfs:
|   Filesystem  1K-blocks   Used       Available   Use%  Maxfilesize  Maxlink
|_  /mnt/nfs    30313412.0  8074868.0  20675664.0  29%   16.0T        32000
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      41982/udp6  mountd
|   100005  1,2,3      45837/tcp   mountd
|   100005  1,2,3      47217/tcp6  mountd
|   100005  1,2,3      58830/udp   mountd
|   100021  1,3,4      39542/udp   nlockmgr
|   100021  1,3,4      44629/tcp   nlockmgr
|   100021  1,3,4      45273/tcp6  nlockmgr
|   100021  1,3,4      47524/udp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp open  nfs_acl 3 (RPC #100227)
MAC Address: 00:00:00:00:00:00 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

Once we have discovered such an NFS service, we can mount it on our local machine. For this, we can create a new empty folder to which the NFS share will be mounted. Once mounted, we can navigate it and view the contents just like our local system.

Show Available NFS Shares

Show Available NFS Shares

```
Govardhan Gujji22@htb[/htb]$ showmount -e 10.129.14.128

Export list for 10.129.14.128:
/mnt/nfs 10.129.14.0/24
```

Mounting NFS Share

Mounting NFS Share

```
Govardhan Gujji22@htb[/htb]$ mkdir target-NFS
Govardhan Gujji22@htb[/htb]$ mount -t nfs 10.129.14.128:/ ./target-NFS/ -o nolock
Govardhan Gujji22@htb[/htb]$ cd target-NFS
Govardhan Gujji22@htb[/htb]$ tree .

.
└── mnt
    └── nfs
        ├── id_rsa
        ├── id_rsa.pub
        └── nfs.share

2 directories, 3 files
```

There we will have the opportunity to access the rights and the usernames and groups to whom the shown and viewable files belong. Because once we have the usernames, group names, UIDs, and GUIDs, we can create them on our system and adapt them to the NFS share to view and modify the files.

List Contents with Usernames & Group Names

List Contents with Usernames & Group Names

```
Govardhan Gujji22@htb[/htb]$ ls -l mnt/nfs/

total 16
-rw-r--r-- 1 cry0l1t3 cry0l1t3 1872 Sep 25 00:55 cry0l1t3.priv
-rw-r--r-- 1 cry0l1t3 cry0l1t3  348 Sep 25 00:55 cry0l1t3.pub
-rw-r--r-- 1 root     root     1872 Sep 19 17:27 id_rsa
-rw-r--r-- 1 root     root      348 Sep 19 17:28 id_rsa.pub
-rw-r--r-- 1 root     root        0 Sep 19 17:22 nfs.share
```

List Contents with UIDs & GUIDs

List Contents with UIDs & GUIDs

```
Govardhan Gujji22@htb[/htb]$ ls -n mnt/nfs/

total 16
-rw-r--r-- 1 1000 1000 1872 Sep 25 00:55 cry0l1t3.priv
-rw-r--r-- 1 1000 1000  348 Sep 25 00:55 cry0l1t3.pub
-rw-r--r-- 1    0    0 1872 Sep 19 17:27 id_rsa
-rw-r--r-- 1    0    0  348 Sep 19 17:28 id_rsa.pub
-rw-r--r-- 1    0    0    0 Sep 19 17:22 nfs.share
```

After we have done all the necessary steps and obtained the information we need, we can unmount the NFS share.

Unmounting

Unmounting

```
Govardhan Gujji22@htb[/htb]$ cd ..
Govardhan Gujji22@htb[/htb]$ umount ./target-NFS
```

Start Instance

∞ / 1 spawns left

Waiting to start...

Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet
Get VPN Key

Target: Click here to spawn the target system!

+ 0 Enumerate the NFS service and submit the contents of the flag.txt in the "nfs" share as the answer.

HTB{hjglmvtkjhlkfuhgi734zthrie7rjmdze}

Submit | No Hint

+ 0 Enumerate the NFS service and submit the contents of the flag.txt in the "nfsshare" share as the answer.

HTB{8o7435zhtuih7fztdrzuhdhkfjcn7ghi4357ndcthzuc7rtfghu34}

Submit | No Hint

Previous | Next | Mark Complete & Next

Table of Contents

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left