

1. RUN WINDOWS LOCAL HARDENING SCRIPT:

```
Iwr -uri https://tinyurl.com/byunccdc/windows/hardening/Local-Hardening.ps1
```

```
If running the script doesn't work: [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

1. **Running Services:** This is your `systemctl list-units`.

- a. **Action:** Open `services.msc`. Go through the list of running services and disable anything that isn't essential. Services like "Telnet," "FTP," or any third-party update services are common targets.

Hardening the HTTP Server (IIS)

On Windows Server, your HTTP server is almost certainly Internet Information Services (IIS).

Immediate Triage

- **Delete Default/Unused Sites:** IIS comes with a "Default Web Site." If your competition site is a different one, delete the default. The same goes for any other sample applications or sites. Attackers have tools to find these defaults.
- **Check the Web Root (C:\inetpub\wwwroot):** Look for sensitive files left by the setup team. This is a classic CCDC tactic. Search for files like `.bak`, `.config`, `.old`, or anything with "password" in the name. Pay special attention to the `web.config` file, as it can contain database connection strings and other secrets.
- **Permissions:** Right-click the `wwwroot` folder for your site -> Properties -> Security. The user account that the web server runs as (often `IIS_IUSRS` or a specific Application Pool Identity) should have read-only access unless the application specifically needs to write files. Remove write/execute permissions where they are not needed.

Configuration Hardening

- **Disable Unnecessary Modules:** In the IIS Manager, you can see installed modules. Things like WebDAV are often enabled but not needed and have been sources of vulnerabilities.
- **Enable HTTPS:** If the service needs to be available via HTTPS, you need a certificate. If one isn't provided, create and bind a self-signed certificate immediately. Then, use a tool like **IIS Crypto** to disable weak SSL/TLS protocols (SSLv2, SSLv3, TLS 1.0, 1.1) and weak ciphers with a single click. Doing this manually in the registry is too slow for a competition.
- **Request Filtering:** This is a powerful feature in IIS. You can configure rules to deny requests that contain common attack strings (like SQL injection characters), double-encoded characters, or have certain extensions (like .bak).
- **Hide Banners:** Attackers use server banners to identify versions and find exploits. You can edit the configuration to remove the Server: Microsoft-IIS/10.0 header from responses.

Hardening the SMB Service

Immediate Triage

- **Verify SMBv1 is Disabled:** As we discussed, this is the protocol EternalBlue targets. You can verify it's off with this PowerShell command. It should return False.

PowerShell

```
Get-SmbServerConfiguration | Select EnableSMB1Protocol
```

- **Check Existing Shares:** Use the command net share or go to "Computer Management" -> "Shared Folders" -> "Shares" to see what's being shared.
- **Check Share Permissions:** Right away, check the permissions on these shares. A folder shared with the Everyone group with "Full Control" is the #1 classic CCDC finding. Lock it down to specific, authenticated users who need access.

Configuration Hardening

- **Principle of Least Privilege:** Remember there are two sets of permissions: **Share Permissions** and **NTFS Permissions** (on the folder itself). The *most restrictive* of the two is what applies. A good strategy is to set Share Permissions to "Authenticated Users" with "Change/Read" and then use the more granular NTFS permissions to control specific access.
- **Remove Administrative Shares:** By default, Windows shares every drive letter (e.g., C\$) and the system root (ADMIN\$). These are hidden but accessible to administrators. In a CCDC environment, these are a massive target. You can disable them via the registry, but be aware this can break some administrative tools.
- **Enable SMB Signing and Encryption:** These are critical for preventing man-in-the-middle attacks and eavesdropping. You can enforce them with PowerShell:

PowerShell

```
# Enforces integrity to prevent tampering
Set-SmbServerConfiguration -RequireSecuritySignature $true

# Encrypts all traffic for shares that are enabled for it
Set-SmbServerConfiguration -EncryptData $true
```

DISABLING SMBV1!!!!

1. The PowerShell Method (Recommended for CCDC)

This is the fastest and most scriptable way. It completely removes the feature.

Step 1: Check the Status Open PowerShell as an Administrator and run this command:

```
PowerShell
Get-WindowsOptionalFeature -Online -FeatureName SMB1Protocol
```

This will show you the "State." If it says Enabled, you need to disable it. If it says Disabled, you are already good to go, but running the disable command won't hurt.

Step 2: Disable the Feature Run the following command. It will disable the SMBv1 client, server, and related drivers.

PowerShell

```
Disable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol
```

You'll be prompted to restart the computer. You can press Y to do it immediately or N to do it later. **A reboot is required for the change to take full effect.**

Pro-Tip for CCDC: You can combine this with a force flag to avoid the prompt and pipe it into a restart command to automate the whole process in a script:

PowerShell

```
# This command will disable SMBv1 and reboot the machine in 60
# seconds.
Disable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol -
-NoRestart; shutdown /r /t 60
```

2. The Server Manager GUI Method

This is a good visual way to confirm the setting if you're not comfortable with PowerShell yet.

1. **Open Server Manager.**
2. Click **Manage** in the top right, then select **Remove Roles and Features**.
3. Click **Next** until you get to the **Features** page.
4. Scroll down and find the item named **SMB 1.0/CIFS File Sharing Support**.
5. **Uncheck the box.** If it's already unchecked, you're all set.
6. Click **Next** through the rest of the wizard, and then click **Remove**.
7. You will be prompted to restart the server to complete the removal.