# **CCDC Defensive Scripting Guide**

*A categorized reference of helpful scripts and one-liners to quickly audit, secure, or monitor systems during competition.*

## **1. Purpose**

This guide includes practical, fast-to-deploy scripts for both Windows and Linux systems. These are intended to be reviewed, customized, and executed to assist in early hardening, threat hunting, auditing, and recovery efforts during a CCDC competition. Scripts are categorized by use case and platform.

## **2. Windows Scripts**

### **2.1. Local User and Group Enumeration**

**Purpose:** Identify unknown or suspicious accounts quickly.

# List all local users

Get-LocalUser

# Show members of local Administrators group

Get-LocalGroupMember -Group "Administrators"

# List enabled users

Get-LocalUser | Where-Object {$\_.Enabled -eq $true}

### **2.2. Firewall Status and Rule Review**

**Purpose:** Quickly audit or enable Windows Firewall.

# Check firewall status for all profiles

Get-NetFirewallProfile

# Enable firewall for all profiles

Set-NetFirewallProfile -Profile Domain,Private,Public -Enabled True

# Show all enabled inbound rules

Get-NetFirewallRule | Where-Object {$\_.Direction -eq "Inbound" -and $\_.Enabled -eq "True"}

### **2.3. Event Log Quick Triage**

**Purpose:** Identify high-priority attacker behavior (e.g., account creation).

# Look for recent user creation events

Get-WinEvent -LogName Security | Where-Object {$\_.Id -eq 4720} | Format-Table TimeCreated, Message

# View last 10 failed logons

Get-WinEvent -LogName Security -MaxEvents 10 | Where-Object {$\_.Id -eq 4625}

### **2.4. Process & Network Inspection**

**Purpose:** Spot abnormal processes and active connections.

# Top CPU-consuming processes

Get-Process | Sort-Object CPU -Descending | Select-Object -First 10

# Show all active network connections

netstat -ano | findstr ESTABLISHED

### **2.5. Executable & Script Monitoring**

**Purpose:** Detect suspicious files in typical Red Team drop zones.

# Search for new executables in temp/public folders

Get-ChildItem C:\Windows\Temp -Recurse -Include \*.exe,\*.ps1 -ErrorAction SilentlyContinue

Get-ChildItem $env:TEMP -Recurse -Include \*.exe,\*.ps1 -ErrorAction SilentlyContinue

## **3. Linux Scripts**

### **3.1. Local User & Group Audit**

**Purpose:** Detect unauthorized users or elevated access.

# List all users

cut -d: -f1 /etc/passwd

# Show users with UID 0 (should usually only be root)

awk -F: '$3 == 0 { print $1 }' /etc/passwd

# Check sudo group membership

getent group sudo

### **3.2. Open Ports and Listening Services**

**Purpose:** Identify exposed services and reduce unnecessary exposure.

# Show all listening ports and associated programs

ss -tulnp

# Simplified netstat version (if available)

netstat -tulnp

### **3.3. Process and File Monitoring**

**Purpose:** Detect suspicious processes or dropped files.

# View top CPU consumers

ps aux --sort=-%cpu | head -n 10

# Search common drop zones

find /tmp /var/tmp /dev/shm -type f \( -name "\*.sh" -o -name "\*.py" -o -name "\*.exe" \)

### **3.4. User Login and History Analysis**

**Purpose:** Triage for account misuse or Red Team persistence.

# View last logins

last

# View currently logged in users

who

# Check command history

cat ~/.bash\_history

### **3.5. Scheduled Tasks and Cron Jobs**

**Purpose:** Identify persistence via scheduled commands.

# List cron jobs for all users

for user in $(cut -f1 -d: /etc/passwd); do crontab -u $user -l 2>/dev/null; done

# Check system-wide cron tasks

cat /etc/crontab

ls /etc/cron.d/

## 

## **4. Cross-Platform Recommendations**

* Always review a script before executing. Never run a one-liner without understanding its impact.
* Keep logs of what scripts were run, when, and on which machine.
* Modify default paths or assumptions based on your environment layout.
* Pair scripting efforts with appropriate documentation and inject reporting.

## **5. Optional Enhancements**

Teams may benefit from packaging these into:

* A deployable PowerShell module for Windows
* A bash script or Ansible playbook for Linux
* A shared repository with comments, version history, and usage notes

Let me know if you’d like this converted into:

* A shareable Google Doc
* A formatted Markdown or PDF handout
* A script bundle for distribution and deployment during competition prep