WINDOWS / ACTIVE DIRECTORY PLAYBOOK

CCDC Windows Defense Guide

FIRST 15 MINUTES

Execute in order. No exceptions.

Minute 0-5: Credential Hygiene

```
# Change local Administrator password
$pwd = ConvertTo-SecureString "NewSecureP@ss2024!" -AsPlainText -Force
Set-LocalUser -Name "Administrator" -Password $pwd

# Change all local admin passwords
Get-LocalUser | Where-Object {$_.Enabled} | ForEach-Object {
    Set-LocalUser -Name $_.Name -Password $pwd
}

# Change Domain Admin password (on DC)
Set-ADAccountPassword -Identity "Administrator" -Reset -NewPassword $pwd
```

Minute 5-10: Disable Dangerous Accounts

```
# Find and disable Guest
Disable-LocalUser -Name "Guest"

# Find suspicious accounts
Get-LocalUser | Select Name, Enabled, LastLogon
Get-ADUser -Filter {Enabled -eq $true} | Select Name, LastLogonDate

# Disable unknown accounts
Disable-LocalUser -Name "SuspiciousUser"
Disable-ADAccount -Identity "SuspiciousUser"
```

Minute 10-15: Verify Services

```
# Check critical services
Get-Service | Where-Object {$_.Status -eq "Running"} |
    Where-Object {$_.Name -match "W3SVC|DNS|SMTP|MSSQL|AD"}

# Quick service health check
Test-NetConnection -ComputerName localhost -Port 80
Test-NetConnection -ComputerName localhost -Port 443
Test-NetConnection -ComputerName localhost -Port 53
```

AD HARDENING CHECKLIST

Critical (Do Immediately)

- Change all admin passwords
- Disable Guest account
- Disable LLMNR (Responder attack)
- Disable NBT-NS
- Enable SMB signing
- Set account lockout policy

Important (Within First Hour)

- Review Domain Admins group membership
- Review local Administrators group on all systems
- Enable advanced audit logging
- Disable null sessions
- Review scheduled tasks
- Check Group Policy for backdoors

DISABLE LLMNR & NBT-NS (Stop Responder)

```
# Disable LLMNR via Registry
New-Item -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT" -Name "DNSClient" -Force
Set-ItemProperty -Path "HKLM:\SOFTWARE\Policies\Microsoft\Windows NT\DNSClient" -Name "EnableMulticast" -Value 0 -Type DWord

# Disable NBT-NS (do on each interface)
$adapters = Get-WmiObject Win32_NetworkAdapterConfiguration | Where-Object {$_.IPEnabled -eq $true}
foreach ($adapter in $adapters) {
    $adapter.SetTcpipNetbios(2)  # 2 = Disable
}
```

SMB HARDENING

```
# Enable SMB Signing (CRITICAL)
Set-SmbServerConfiguration -RequireSecuritySignature $true -Force
Set-SmbClientConfiguration -RequireSecuritySignature $true -Force

# Disable SMBv1
Disable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol -NoRestart

# Check current SMB settings
Get-SmbServerConfiguration | Select EnableSMB1Protocol, RequireSecuritySignature
```

ACCOUNT LOCKOUT POLICY

```
# Set via Group Policy (preferred) or:
net accounts /lockoutthreshold:5
net accounts /lockoutduration:30
```

```
net accounts /lockoutwindow:30

# Verify
net accounts
```

WINDOWS FIREWALL

```
# Enable firewall on all profiles
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True

# Set default deny inbound
Set-NetFirewallProfile -DefaultInboundAction Block -DefaultOutboundAction Allow

# Allow specific services (adjust as needed)
New-NetFirewallRule -DisplayName "Allow HTTP" -Direction Inbound -Port 80 -Protocol TCP -Action Allow
New-NetFirewallRule -DisplayName "Allow HTTPS" -Direction Inbound -Port 443 -Protocol TCP -Action Allow
New-NetFirewallRule -DisplayName "Allow DNS" -Direction Inbound -Port 53 -Protocol UDP -Action Allow
New-NetFirewallRule -DisplayName "Allow RDP from Admin" -Direction Inbound -Port 3389 -Protocol TCP -RemoteAddress 10.0.0.0/24 -Acti

# Block specific attacker IP
New-NetFirewallRule -DisplayName "Block Attacker" -Direction Inbound -RemoteAddress 10.10.10.50 -Action Block

# View current rules
Get-NetFirewallRule | Where-Object {$_.Enabled -eq "True"} | Select DisplayName, Direction, Action
```

AUDIT LOGGING

```
# Enable comprehensive logging
auditpol /set /category:"Logon/Logoff" /success:enable /failure:enable
auditpol /set /category:"Account Logon" /success:enable /failure:enable
auditpol /set /category:"Account Management" /success:enable /failure:enable
auditpol /set /category:"Object Access" /success:enable /failure:enable
auditpol /set /category:"Policy Change" /success:enable /failure:enable
auditpol /set /category:"Privilege Use" /success:enable /failure:enable

# Verify
auditpol /get /category:*
```

KEY EVENT IDS TO MONITOR

Event ID	Meaning	Action
4625	Failed login	Check for brute force
4624	Successful login	Verify legitimate
4720	User created	Investigate immediately
4732	User added to group	Check if authorized
4672	Special privileges assigned	Verify admin logon

7045	Service installed	Check for malware
4688	Process created	Look for suspicious
1102	Audit log cleared	RED FLAG - investigate

Quick Log Queries

```
# Failed logins (last 100)
Get-WinEvent -FilterHashtable @{LogName="Security";ID=4625} -MaxEvents 100 |
    Select TimeCreated, @{N='User';E={$_.Properties[5].Value}}, @{N='Source';E={$_.Properties[19].Value}}

# New users created
Get-WinEvent -FilterHashtable @{LogName="Security";ID=4720} -MaxEvents 50

# Services installed
Get-WinEvent -FilterHashtable @{LogName="System";ID=7045} -MaxEvents 20

# Admin logons
Get-WinEvent -FilterHashtable @{LogName="Security";ID=4672} -MaxEvents 50
```

PERSISTENCE HUNTING

Check Scheduled Tasks

```
Get-ScheduledTask | Where-Object {$_.State -eq "Ready"} |
    Select TaskName, TaskPath, @{N='Action';E={$_.Actions.Execute}}

# Look for suspicious tasks
schtasks /query /fo LIST /v | findstr /i "Task\|Run\|Author"
```

Check Services

```
Get-WmiObject win32_service |
    Where-Object {$_.StartMode -eq "Auto"} |
    Select Name, DisplayName, PathName, StartName |
    Format-Table -AutoSize

# Look for unquoted service paths (privesc)
wmic service get name,displayname,pathname,startmode | findstr /i "auto" | findstr /i /v "c:\windows\\" | findstr /i /v """
```

Check Run Keys

```
Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
Get-ItemProperty -Path "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce"
```

Check Startup Folders

```
Get-ChildItem "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
Get-ChildItem "$env:APPDATA\Microsoft\Windows\Start Menu\Programs\Startup"
```

GPO QUICK REFERENCE

Check Applied GPOs

```
gpresult /r
gpresult /h gpo_report.html  # Generate HTML report
```

Force GPO Update

```
gpupdate /force
```

Check for GPO Backdoors

- Review all linked GPOs in Active Directory Users and Computers
- Check for unexpected scripts in:
- Computer Configuration > Scripts
- User Configuration > Scripts
- Review security settings in Default Domain Policy

CREDENTIAL ROTATION SOP

1. **Domain Admin Accounts**
- Change password to 20+ chars
- Document new password securely
- Test login immediately
2. **Service Accounts**
- Identify all service accounts: `Get-ADServiceAccount -Filter *`
- Change passwords
- Restart dependent services
- Verify services function
3. **Local Admin Accounts**
- Change on every machine
- Use different passwords per machine if possible
- Document mapping

QUICK RECOVERY COMMANDS

If Service Won't Start

```
# Check service status and dependencies
Get-Service <ServiceName> | Select *
Get-Service <ServiceName> -DependentServices

# Check event log for errors
Get-WinEvent -LogName System -MaxEvents 50 | Where-Object {$_.LevelDisplayName -eq "Error"}

# Try restart
Restart-Service <ServiceName> -Force
```

If Locked Out

```
# Unlock account
Unlock-ADAccount -Identity "username"

# Reset lockout
Search-ADAccount -LockedOut | Unlock-ADAccount
```

If GPO Broken

```
# Reset to default
dcgpofix /ignoreschema
```

WINDOWS DEFENDER MANTRA

> *""Change passwords FIRST."*
>
> *"Block LLMNR SECOND."*
>
> *"Everything else AFTER services are verified.""*

CCDC.x1000.ai - Championship Training