Windows Initial Checklist

(Past checklist…
https://docs.google.com/document/d/1Cb-l9I-3WZpkUoZF13otxeazFGACwFaoaKNAcq0LBb8)

Get basic info about your system

```
> systeminfo
```

- Hostname (Host name:)
- OS Version (OS Name: and OS Version:)
- OS Configuration (OS Configuration:)
- IP Addresses (Network Card(s):)

```
> Get-netAdapter
```

- MAC Addresses (MacAddress Column)

```
> Get-service | where-object {$_.status -eq 'running'}
```

- SSH Server
- WinRM (powershell remote management)
- DNS
- DHCPServer
- IIS

```
> gsmbs
```

There are three default SMB shares, ADMIN$, C$, and IPC$.. In a DC, there are also NETLOGON and SYSVOL. Ignore these.

Secure old account and create new local admin account:

```
 > Set-LocalUser -Name <username> -Password
      (Read-Host -AsSecureString)
> New-LocalUser <username> -Password (Read-Host
-AsSecureString)
> Add-LocalGroupMember -Group "Administrators" -Member
      "NEW_ACCOUNT_NAME"
```

Get and lock existing local users **(NOT ON DC… OR ELSE!)** :

```
> glu | disable-localuser
Then re-enable your accounts
> enable-localuser <my account>
```

If you're a domain controller:

Congrats! It's your job to change allllllll the passwords. Change the domain admin password first.

```
> Set-ADAccountPassword -Identity Administrator -Reset
      -NewPassword (ConvertTo-SecureString -AsPlainText '<new
      systepassword>' -Force)
```

- Then, check who is a member of important builtin groups

```
> Get-ADGroupMember "<group name>"
```

- Domain Admins
- Schema Admins
- Enterprise Admins
- Administrators

If they don't need to be there,

```
> Remove-ADGroupMember -Identity "Domain Admins" -Members <...>
```

- Finally, make a new domain admin account for your team's usage.

```
> New-ADUser -Name <name> -SamAccountName <name>@<domain> -Enabled
$True -AccountPassword (Read-Host "pass" -AsSecureString)
> Add-ADGroupMember "Domain Admins" -Members <your admin account
```

- (**Optional**) Turn on default group policy

  This is best done from the group policy GUI console. Enable and enforce the links for the "Default Domain Policy" and "Default Domain Controller Policy" GPO objects in your domain. First, be sure to change the setting under "Default Domain Policy" > "Computer" > "Windows" > "Security" > "Password Policy" > "Minimum Password Age" to 0.

Install Useful Software

```
> iwr -OutFile 'install.ps1'
'https://chocolatey.org/install.ps1'
> Set-ExecutionPolicy Bypass -Scope Process -Force
> .\install.ps1
```

Start an installer for some software in the background

```
> Start-Job { choco install -y firefox autoruns tcpview
    winlogbeat }
> Set-ExecutionPolicy Restricted -Scope Process -Force # wait
    until the job is done
```

If you are not on a domain controller, you may want to install AD Powershell tools

```
> Get-WindowsCapability -Name Rsat.ActiveDirectory* -Online |
    Add-WindowsCapability -Online
```

Also, if you have a GUI install group policy tools

```
> Get-WindowsCapability -Name Rsat.GroupPolicy* -Online |
    Add-WindowsCapability -Online
```

Enable Security Services / anti virus

```
> Get-MPComputerStatus
```

If it is disabled, enable it:

```
> Set-MpPreference -DisableRealtimeMonitoring $False
```

Backup Important Data

- DNS

```
> Export-DNSSeverZone -Name <Zone Name> -FileName <Backup
FileName>
```

The backup file can be found in C:\Windows\System32\dns

- IIS: Zip the wwwroot directory and copy it somewhere else
- SMB: Use the > gsmbs command to see what shares are there. Any important content on non-default shares should be zipped and copied elsewhere

Firewall Time

- Enable the firewall

```
> Set-NetFirewallProfile -Profile Domain -Enabled True
```

- Set the firewall to block outbound traffic be default, and log

```
> Set-NetFirewallProfile -DefaultOutboundAction Block
      -LogBlocked True
```

- Create rules allowing outbound traffic for essential applications

```
> New-NetFirewallRule -name "<NAME>" -DisplayName "<NAME>"
-action Allow -Direction Outbound -program "<PATH>"
```

- For example, to use **chromium** you can use the following rule:

```
> $p = get-command chrome
> New-NetFirewallRule -name "chome_out" -DisplayName "Allow
Chrome" -action Allow -Direction Outbound -program $p.source
```

- For Chocolatey, you can allow traffic to the IP address of the API endpoint (chocolatey.org)

```
> ping chocolatey.org # get the ip address
> New-NetFirewallRule -Name "choco_out" -DisplayName "Allow
Chocolatey" -Direction Outbound -RemoteAddress <choco ip
address>
```

- For SSH

```
> New-NetFirewallRule -name "ssh_out" -DisplayName "Allow SSH"
-action Allow -Direction Outbound -program (get-command
ssh).Source
```

- For winlogbeat

```
> $p = C:\ProgramData\chocolatey\bin\winlogbeat.exe
> New-NetFirewallRule -name "winlogbeat_out" -DisplayName
"Allow Winlogbeat" -action Allow -Direction Outbound -program
$p
```

Some Basic Forensics

- Look at scheduled tasks
- Check services
- Check open ports

Repeat Forensics

If you have a chance later, run through these basic checks of your server's security at later points in the competition. Especially important on the DC.

Enable Security Services / anti virus

```
> Get-MPComputerStatus
```

If it is disabled, enable it:

```
> Set-MpPreference -DisableRealtimeMonitoring $False
```

Audit Accounts

Check who is a member of important builtin groups

```
> Get-ADGroupMember "<group name>"
```

- Domain Admins
- Schema Admins
- Enterprise Admins
- Administrators

Double Check Your Firewall

```
> Get-NetFirewallProfile | select
    Name,Enabled,DefaultOutboundAction,LogBlocked
> cat -tail 25 C:\Windows\System32\LogFiles\Firewall\pfirewall.log
```

Are there any weird blocked connections in the log?