

## Phase 1 Building Enterprise Environments (Blackridge Bank)

---

### 1. Set up Active Directory

I started by setting up a basic Active Directory domain — the kind of thing that quietly runs the backend of a lot of organizations. I followed [this IBM guide](#) as a reference just to make sure I wasn't missing anything big.

I named the domain `blackridge.local` to match the fictional bank this lab is built around.

How I structured it

Instead of throwing all users and computers into a giant, chaotic pile, I used **Organizational Units (OUs)** to separate things cleanly by site. That's how it's done in real orgs, especially those with multiple offices.

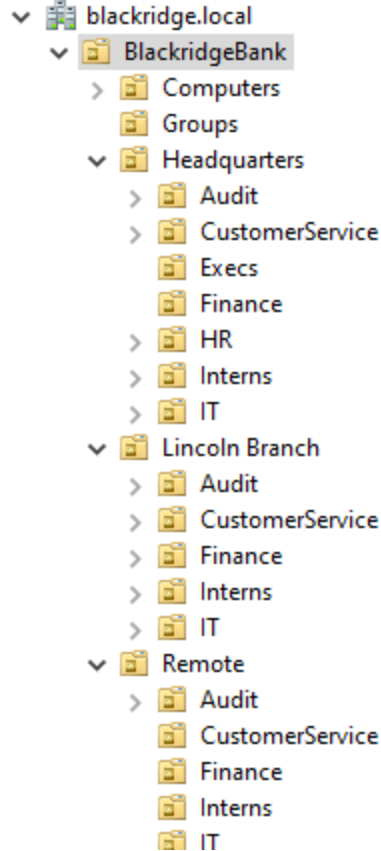
At the top, there's a master OU called `BlackridgeBank`, and under that I split it into three logical locations:

- `Headquarters`
- `Lincoln Branch`
- `Remote`

Each site has two OUs inside it:

- `Users` → for the people
- `Computers` → for the machines

This setup makes it way easier to apply policies later on. For example, I can give different settings to just Remote workers or push a drive mapping only to Lincoln Branch. It also makes group policy targeting and delegation feel a lot more organized.



---

## 2. Created Groups for RBAC

Once the OU structure was in place, the next thing I tackled was access control. And to be honest, I didn't feel like manually assigning folder permissions to 100 users one at a time — that's a headache even in a fake environment.

So instead, I went with a **group-based model** using something called **role-based access control**, or **RBAC** for short.

RBAC just means people get access based on **what they do**, not who they are. If you're in the IT department, you're part of the IT group. If you're an intern, you're in the intern group. Then, I give those groups access to what they need — once. No micromanaging.

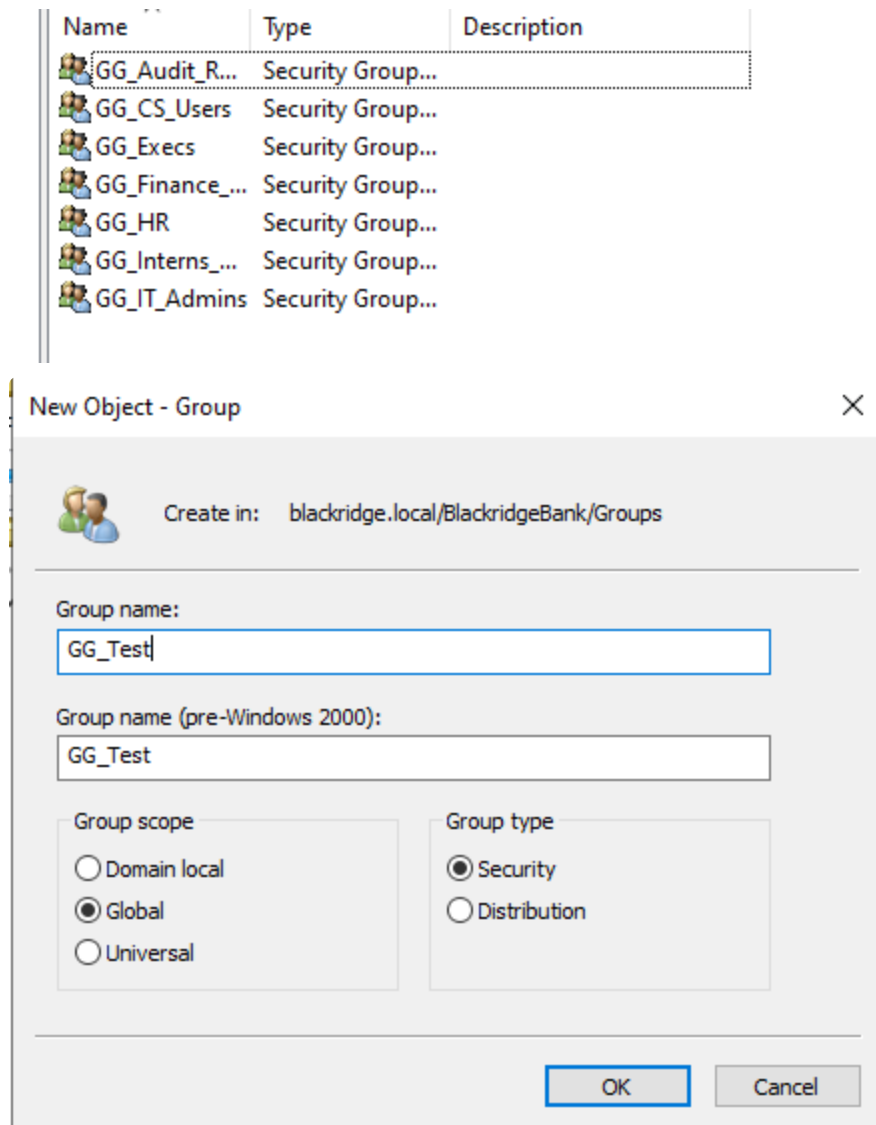
To keep it consistent and recognizable, I used the naming convention `GG_` for "Global Group." It makes them easy to spot and filter later.

Here are the ones I created:

- `GG_IT_Admins` — for sysadmins and IT staff
- `GG_Interns_Limited` — interns, with limited permissions (read-only in most places)
- `GG_Finance_Read` — finance team, with modify access to financial folders

- `GG_HR` — HR staff, access to employee-related files
- `GG_CS_Users` — customer service reps
- `GG_Audit_Read` — for internal/external auditors (read-only)
- `GG_Execs` — executive team, C-suite, board members

Once these groups were built, it was super easy to assign folder permissions, set GPOs, or delegate access. And later in the project, some of these groups (like `GG_Interns_Limited`) will be misused — on purpose — to simulate insider threats.



## 2.5. Deployed 100 Users and Groups via PowerShell

To automate the creation of users and groups in Active Directory, I wrote a script called `Deploy-BlackridgeAD.ps1`. This saved *hours* of manual clicking and ensured the setup was repeatable and clean.

What it does:

- Checks if security groups like `GG_IT_Admins` , `GG_Audit_Read` , etc. already exist — if not, creates them inside `OU=Groups`
- Creates 100 users with realistic names, titles, departments, and human-like passwords (e.g., `Kevin6398#` )
- Places users in department-based OUs inside the correct site ( `Headquarters` , `Lincoln Branch` , or `Remote` )
- Adds each user to their appropriate security group (e.g., `GG_CS_Users` , `GG_Interns_Limited` )
- Logs usernames and passwords to `C:\Blackridge_AD_100Users_Log.txt` for testing and validation
- 

Smart logic:

- Uses `if (-not (Get-ADUser ...))` so it won't duplicate users on rerun
- Passwords are passed using `ConvertTo-SecureString` , but they are still in plaintext (okay for a controlled lab environment)
- Adds final elevated access for `edrake` and `sbanks` to `GG_IT_Admins` and ``Domain Admins`

```
# Define top-level OU and substructures
$locations = @("Headquarters", "Lincoln Branch", "Remote")
$departments = @{
    "IT" = "GG_IT_Admins"
    "Interns" = "GG_Interns_Limited"
    "Finance" = "GG_Finance_Read"
    "HR" = "GG_HR"
    "CustomerService" = "GG_CS_Users"
    "Audit" = "GG_Audit_Read"
    "Execs" = "GG_Execs"
}

# Ensure OUs exist
foreach ($location in $locations) {
    $baseOU = "OU=$location,OU=BlackridgeBank,DC=blackridge,DC=local"
    if (-not (Get-ADOrganizationalUnit -LDAPFilter "(ou=$location)" -ErrorAction SilentlyContinue)) {
        New-ADOrganizationalUnit -Name $location -Path "OU=BlackridgeBank,DC=blackridge,DC=local"
    }
    foreach ($type in @("Users", "Computers")) {
        $childOU = "OU=$type,$baseOU"
        if (-not (Get-ADOrganizationalUnit -LDAPFilter "(ou=$type)" -SearchBase $baseOU -ErrorAction SilentlyContinue)) {
            New-ADOrganizationalUnit -Name $type -Path $baseOU
        }
    }
}

# Ensure groups exist
foreach ($group in $departments.Values) {
    if (-not (Get-ADGroup -Filter { Name -eq $group })) {
        New-ADGroup -Name $group -GroupScope Global -GroupCategory Security -Path "OU=Groups,DC=blackridge,DC=local"
    }
}

# Generate 100 users
$users = @()
for ($i = 1; $i -le 100; $i++) {
    $firstName = "User$i"
    $lastName = "Test"
    $username = "utest$i"
    $password = ConvertTo-SecureString "User$i!" -AsPlainText -Force
    $dept = ($departments.Keys | Get-Random)
    $group = $departments[$dept]
    $location = $locations | Get-Random
    $userOU = "OU=Users,OU=$location,OU=BlackridgeBank,DC=blackridge,DC=local"

    if (-not (Get-ADUser -Filter { SamAccountName -eq $username })) {
        New-ADUser -Name "$firstName $lastName" `
            -GivenName $firstName `
            -Surname $lastName `
            -SamAccountName $username `
            -UserPrincipalName "$username@blackridge.local" `
            -AccountPassword $password `
            -Enabled $true `
            -Path $userOU `
            -Department $dept `
            -Title "Analyst"
        Add-ADGroupMember -Identity $group -Members $username

        $users += "$username : User$i!"
    }
}

# Log credentials
$logPath = "C:\Blackridge_AD_100Users_Log.txt"
$users | Out-File -FilePath $logPath -Encoding UTF8

# Add special accounts with elevated privileges
```








The end result: a fully built, realistic Active Directory environment that's ready for red and blue team testing, complete with users, departments, and RBAC structure — all generated in minutes instead of hours.

### 3. Created Department Folders

Made folders at `C:\` to simulate network drives you'd see in a real org.

- C:\ITS\_Tools
- C:\InternFiles
- C:\FinanceShare
- C:\HRDocs
- C:\AuditLogs
- C:\CS\_Docs
- C:\ExecReports

These will be accessed over the network and mapped via drive letters later.

 ITS_Tools	5/10/2025 10:20 PM	File folder
 InternFiles	5/10/2025 10:20 PM	File folder
 FinanceShare	5/10/2025 10:20 PM	File folder
 HRDocs	5/10/2025 10:20 PM	File folder
 AuditLogs	5/10/2025 10:21 PM	File folder
 CS_Docs	5/10/2025 10:21 PM	File folder
 ExecReports	5/10/2025 10:21 PM	File folder

#### 4. Locked Down NTFS Permissions

By default, all domain users had read access — no thanks.

I tightened each folder's permissions using NTFS:

Steps:

- Right-click folder → Properties → Security → Advanced
- Disable inheritance → Convert permissions
- Removed Users (BLACKRIDGE\Users)
- Added only:
  - The correct GG\_ group
  - Administrators, SYSTEM, CREATOR OWNER

Example from ITS\_Tools :

Select Users, Computers, Service Accounts, or Groups

Select this object type:  
Users, Groups, or Built-in security principals

From this location:  
blackridge.local

Enter the object names to select (examples):  
GG IT Admins

Advanced... OK Cancel

Advanced Security Settings for ITS\_Tools

Name: C:\ITS\_Tools

Owner: Administrators (BLACKRIDGE\Administrators) Change

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

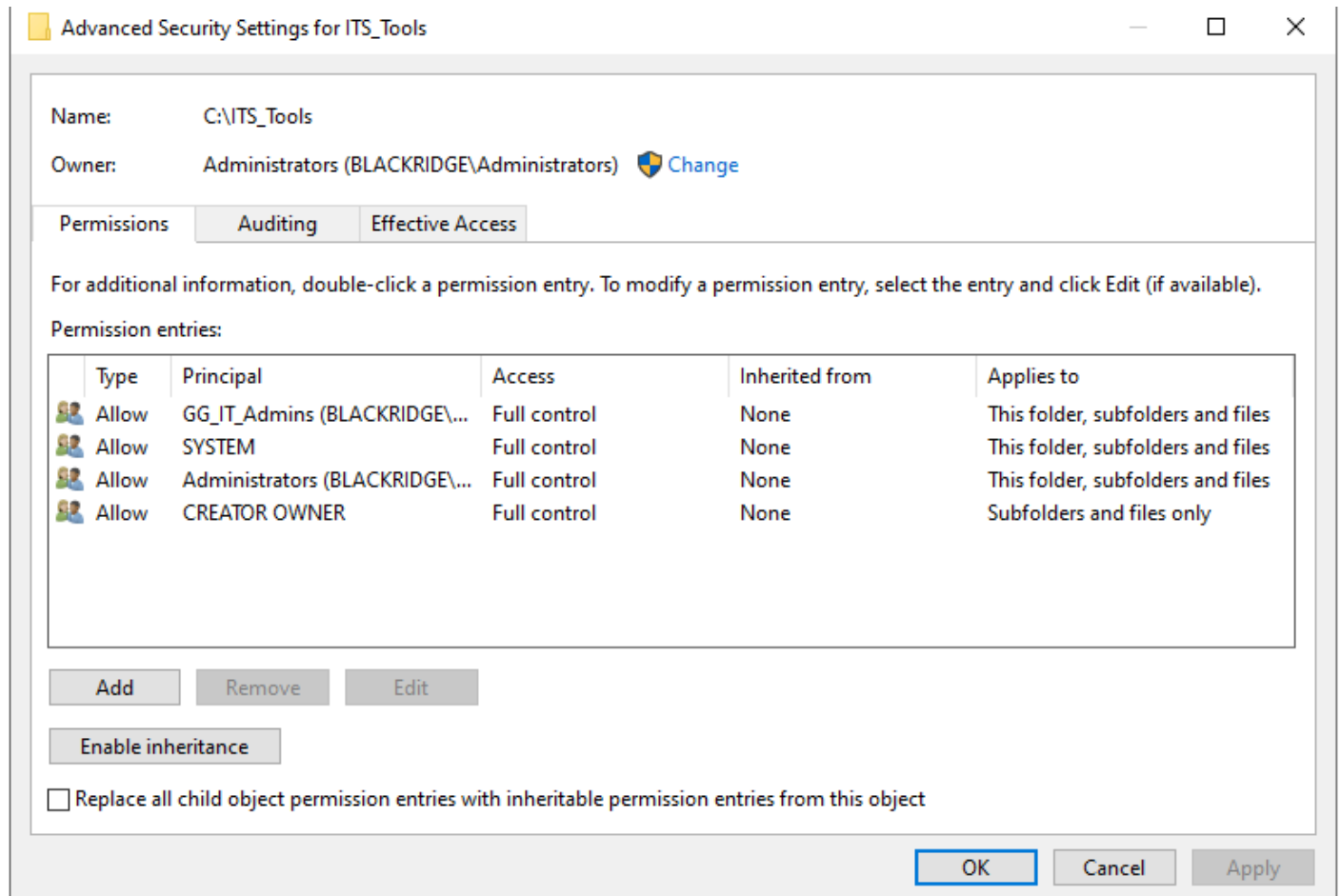
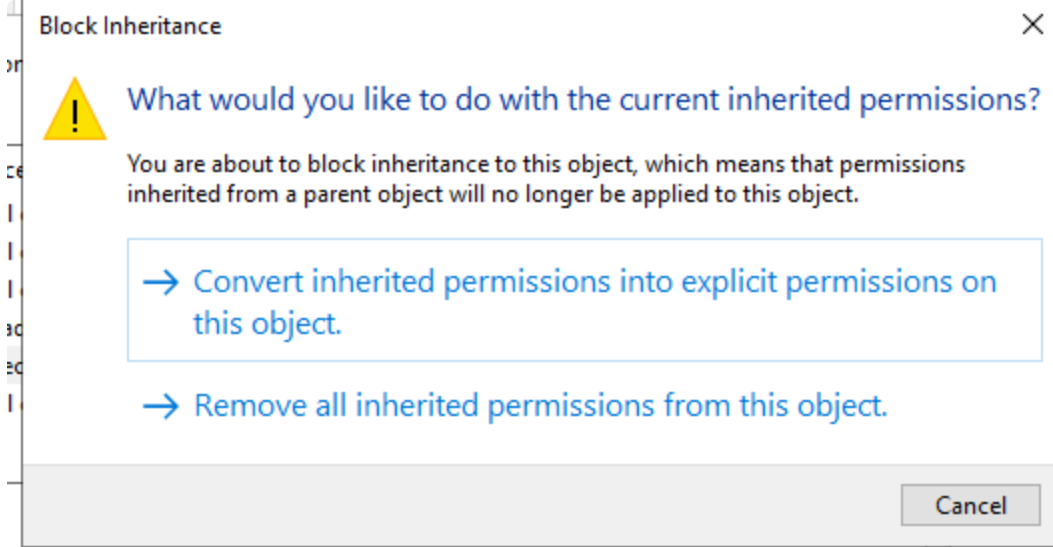
Type	Principal	Access	Inherited from	Applies to
Allow	GG_IT_Admins (BLACKRIDGE\...	Full control	None	This folder, subfolders and files
Allow	SYSTEM	Full control	C:\	This folder, subfolders and files
Allow	Administrators (BLACKRIDGE\...	Full control	C:\	This folder, subfolders and files
Allow	Users (BLACKRIDGE\Users)	Read & execute	C:\	This folder, subfolders and files
Allow	Users (BLACKRIDGE\Users)	Special	C:\	This folder and subfolders
Allow	CREATOR OWNER	Full control	C:\	Subfolders and files only

Add Remove View

Disable inheritance

☐ Replace all child object permission entries with inheritable permission entries from this object

OK Cancel Apply



## ✓ Folder Permissions Summary

Folder	Group	Permission
ITS_Tools	GG_IT_Admins	Full Control
InternFiles	GG_Interns_Limited	Read
FinanceShare	GG_Finance_Read	Modify



Folder	Group	Permission
HRDocs	GG_HR	Modify
AuditLogs	GG_Audit_Read	Read
CS_Docs	GG_CS_Users	Read
ExecReports	GG_Execs	Full Control

## 5. Seeded Fake Enterprise Data






Wrote a PowerShell script [Seed-MockBankData.ps1](#) that adds fake but realistic files to each folder:

- `Q1_Profit_Loss_2025.xlsx`
- `Wire_Transfer_Template.docx`
- `Employee_Compensation_Grid.xlsx`
- `Intern_Onboarding_Guide.pdf`
- `Executive_Credentials_Backup.txt` 🐙

Also:

- Adds fake content to `.txt` , `.csv` , `.ps1`
- Randomizes file timestamps to make it feel lived-in

is PC > Local Disk (C:) > ITS\_Tools

Name	Date modified	Type	Size
 AD_Backup_Config	3/20/2025 12:53 PM	XML Document	0 KB
 GG_IT_Admins_Members	5/2/2025 12:53 PM	CSV File	1 KB
 Inventory_Report_2025.xlsx	4/21/2025 12:53 PM	XLSX File	0 KB
 Password_Reset_Script	3/16/2025 12:53 PM	Windows PowerS...	1 KB
 VPN_Configuration_Manual	3/13/2025 12:53 PM	Microsoft Edge P...	0 KB

## 6. Shared All Folders Over the Network (SMB)

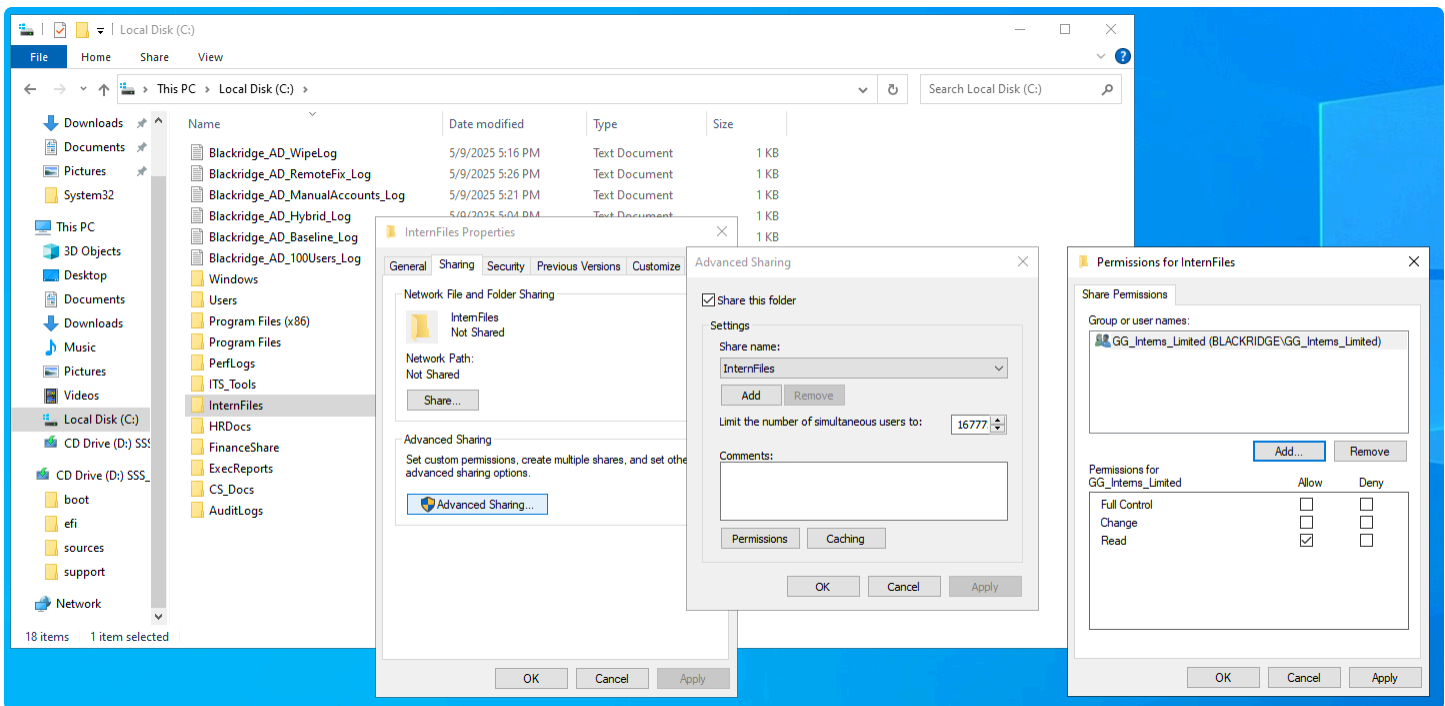
Each department folder is now shared using **Advanced Sharing** with correct group-level permissions.

Example:

- Shared `C:\InternFiles` as `\\d01\InternFiles`
- Share permissions:
  - Removed `Everyone`
  - Added `GG_Interns_Limited` with **Read** access

✓ Share Access Paths:

- `\\d01\ITS_Tools`
- `\\d01\InternFiles`
- `\\d01\FinanceShare`
- `\\d01\HRDocs`
- `\\d01\AuditLogs`
- `\\d01\CS_Docs`
- `\\d01\ExecReports`



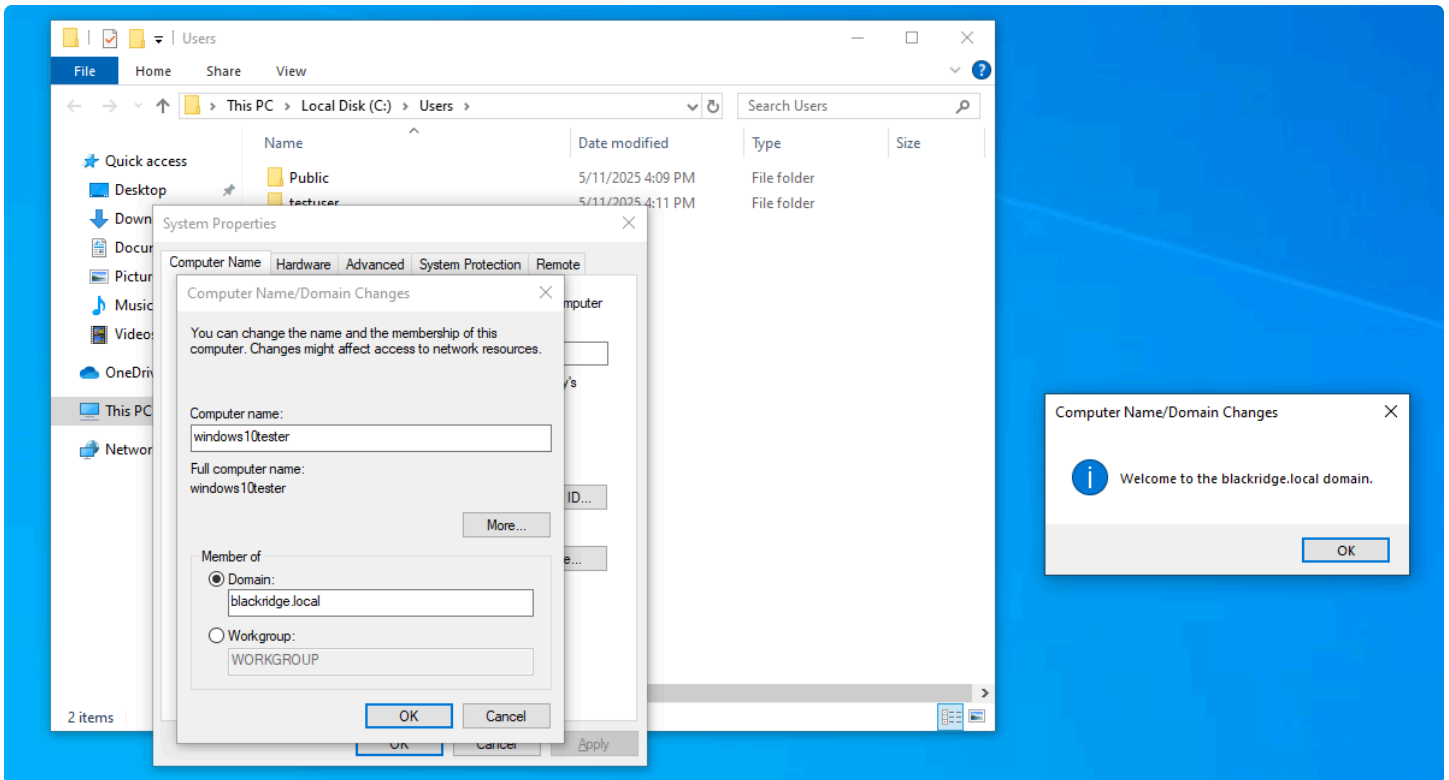
## 7. Built Windows 10 Test Machine

Made a Windows 10 VM in Proxmox to simulate a normal user workstation.

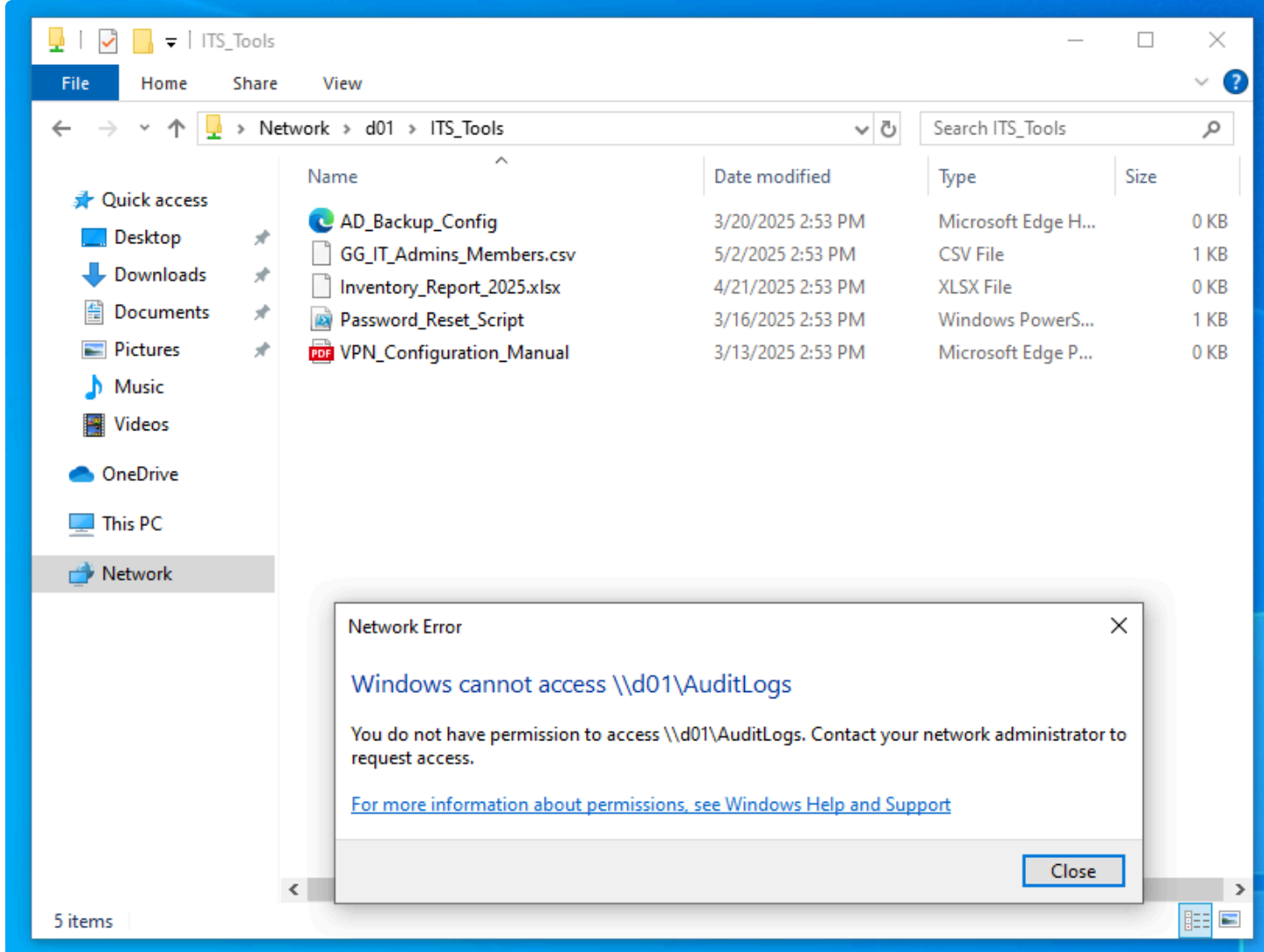
Steps:

- Installed from ISO

- Named the PC something like `W10-Client01`
- Manually pointed DNS to the DC's IP: `192.168.1.20` ✓
- Joined domain: `blackridge.local`
- Logged in as IT staff (e.g., `blackridge\tzane12`)



- checked if the folders i could access were accessible and the ones I didn't weren't



## ✓ Current Lab Status

- ✓ AD domain: `blackridge.local` up and running
- ✓ Users + groups structured and realistic
- ✓ Folders with NTFS + share permissions mapped by group
- ✓ Fake data in place for red/blue team testing
- ✓ Client PC joined to domain and fully functional
- ✓ Shared folders fully accessible from client

## 8. Cloud Setup (Azure Side of Blackridge Bank)

To mirror what you'd see in a real hybrid company, I built out a cloud environment using a free Azure account. This is **Blackridge Bank's cloud side** — but it's just as messy and unstructured as the on-prem side.

Most companies (especially fintech startups) start with Active Directory, then bolt on Azure later without cleaning things up. That's exactly what I wanted to show here:

**a loosely connected, over-permissioned Azure setup that an insider could totally abuse.**

---

## Azure IAM Users

These are the accounts I created to simulate typical employees, but with bad cloud identity management:

- `j.doe@blackridgebank.com` – Finance

Should only have limited read access to finance stuff, but was added to the full admin group.

- `sbanks@blackridgebank.com` – Executive

A Domain Admin on-prem who got recreated manually in Azure and given Global Admin. Terrible identity lifecycle.

- `cloudops@blackridgebank.com` – DevOps

Has long-lived access keys that end up hardcoded in a `.ps1` script. These get leaked and used by Eliot.

- `azureadmin@blackridge.local` – Simulated hybrid identity

A manually created Azure account to mimic a synced AD admin. Same name and password reused — no federation, no real sync.

# Create new user ...

Create a new internal user in your organization

- Basics
- Properties
- Assignments
- Review + create

Create a new user in your organization. This user will have a user name like alice@contoso.com. [Learn more](#)

## Identity

User principal name \*

cloudadmin

@

denneythogmail.onmicr... ▾

Domain not listed? [Learn more](#)

Mail nickname \*

cloudadmin

☒ Derive from user principal name

Display name \*

Cloud Administrator

Password \*



Blackridge123!

☐ Auto-generate password

Account enabled ⓘ

☒

Basics

User principal name	sbanks@denneythogmail.onmicrosoft.com	
Display name	Sandra Banks	
Mail nickname	sbanks	
Password	<div>ILoveKittens1234!</div>	
Account enabled	Yes	

Properties

User type	Member
-----------	--------

Assignments

Administrative units
Groups
Roles

# Create new user

Create a new internal user in your organization

- Basics
- Properties
- Assignments
- Review + create

## Basics

User principal name	cloudops@denneythogmail.onmicrosoft.com
Display name	CloudOps Engineer
Mail nickname	cloudops
Password	CoolBeans1234561!
Account enabled	Yes

## Properties

User type	Member
-----------	--------

## Assignments

- Administrative units
- Groups
- Roles





# Create new user ...

Create a new internal user in your organization

Basics   Properties   Assignments   Review + create

## Basics

User principal name	azureadmin@denneythogmail.onmicrosoft.com 
Display name	AD Admin
Mail nickname	azureadmin
Password	<input type="password" value="....."/> 
Account enabled	Yes

## Properties

User type	Member
-----------	--------

## Assignments

Administrative units


Groups

Roles

---

 IAM Group: `CloudAdmins`

- I created a group called `CloudAdmins` and gave it **full Global Admin rights**.
- Then I added *everyone* — finance, execs, DevOps, even the fake AD user.
- There's no conditional access, no MFA, and no role scoping.

 This mirrors real orgs where privilege just spreads because no one locks things down.

CloudAdmins | Members

Overview  
Diagnose and solve problems  
Manage  
Properties  
Members  
Owners  
Roles and administrators  
Administrative units  
Group memberships  
Applications  
Licenses  
Azure role assignments  
Activity  
Troubleshooting + Support

Direct members All members

Search Add filter

5 group members found

<input type="checkbox"/>	Name	Type	Email	User type	Object Id	Device Id
<input type="checkbox"/>	AD Admin	User		Member	386aa480-0675-4d6c-970d-4c0c520a824b	
<input type="checkbox"/>	Cloud Administrator	User		Member	1d2c1c2a-2241-4de8-a0c5-66ce49ddf99	
<input type="checkbox"/>	CloudOps Engineer	User		Member	1be34dab-a538-42ee-94c6-9cf7e2830e7f	
<input type="checkbox"/>	Jane Doe	User		Member	dc0b348c-b0c0-4da1-b2e3-eb39f93a62f1	
<input type="checkbox"/>	Sandra Banks	User		Member	c9d0bda0-94f9-4050-bf9d-9c9621b2d39d	

Global Administrator | Assignments

All roles

Add assignments Remove assignments Download assignments Refresh Manage in PIM Got feedback?

Diagnose and solve problems  
Manage  
Assignments  
Description  
Activity  
Troubleshooting + Support

You can also assign built-in roles to groups now. [Learn More](#)

Search

Search by name

Type

All

Name	UserName
<input type="checkbox"/> Cloud Administrator	cloudadmin@denneythogmail.onmicrosoft.com
<input type="checkbox"/> Denney Thongphet	denneytho@gmail.com

## What is a Blob / Container?

In Azure:

- A **Blob** is basically a file — could be a document, script, photo, anything.
- A **Container** is like a folder that holds blobs.
- A **Storage Account** is the top-level thing that holds all your containers.

So for Blackridge:

- I created a storage account for finance data
- Inside that, I created a container called `financial-data`
- Then uploaded sensitive docs — and **left it open to the public**

Storage Setup: `blackridgefinance`

- Created a storage account

Basics

Advanced

Networking

Data protection

Encryption

Tags

Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

#### Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \*

Azure subscription 1



Resource group \*

(New) blackridgecloud

[Create new](#)

#### Instance details

Storage account name \* ⓘ

blackridgefinance

Region \* ⓘ

(US) East US

[Deploy to an Azure Extended Zone](#)

Primary service ⓘ

Azure Blob Storage or Azure Data Lake Storage Gen 2

Performance \* ⓘ

- ☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)
- ☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy \* ⓘ

Locally-redundant storage (LRS)

- I left **anonymous access enabled** — because someone forgot to disable it during the rushed cloud migration.

Allow Blob anonymous access ⓘ

☐ Disabled ☒ Enabled

- Created the `financial-data` container

# New container



Name \*

financial-data

Anonymous access level ⓘ

Container (anonymous read access for containers and blobs) ▾

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.

▾ Advanced

- Uploaded fake files:
  - Exec compensation
  - Audit findings
  - Cloud keys

[Blob containers](#) > [financial-data](#)

Authentication method: Access Key ([Switch to Microsoft Entra user account](#))

▾ Add filter

🔍 Search blobs by prefix (case-sensitive)

Only show active blobs ▾

Showing all 7 items

<input type="checkbox"/> Name	Last modified	Access tier	Blob type	Size	Lease state
<input type="checkbox"/> AuditFindings_2024.txt	5/11/2025, 8:26:20 PM	Hot (Inferred)	Block blob	178 B	Available ...
<input type="checkbox"/> Blackridge_MFA_Disabled_Users.csv	5/11/2025, 8:26:20 PM	Hot (Inferred)	Block blob	82 B	Available ...
<input type="checkbox"/> CloudAccess_Credentials.txt	5/11/2025, 8:26:20 PM	Hot (Inferred)	Block blob	63 B	Available ...
<input type="checkbox"/> Exec_Compensation_Grid.csv	5/11/2025, 8:26:20 PM	Hot (Inferred)	Block blob	119 B	Available ...
<input type="checkbox"/> FinanceDept_CredRequest.ps1	5/11/2025, 8:26:20 PM	Hot (Inferred)	Block blob	129 B	Available ...
<input type="checkbox"/> Q4_Forecast_2025.csv	5/11/2025, 8:26:21 PM	Hot (Inferred)	Block blob	78 B	Available ...
<input type="checkbox"/> meeting_notes.txt	5/11/2025, 8:26:58 PM	Hot (Inferred)	Block blob	73 B	Available ...

Key Vault Setup

- Created blackridge-vault

## Create a key vault ...

**Basics** Access configuration Networking Tags Review + create

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<div>Azure subscription 1</div>
Resource group *	<div>blackridgecloud</div> <div>Create new</div>

### Instance details

Key vault name * ⓘ	<div>blackridge-vault</div>
Region *	<div>East US</div>
Pricing tier * ⓘ	<div>Standard</div>

### Recovery options

Soft delete protection will automatically be enabled on this key vault. This feature allows you to recover or permanently delete a key vault and secrets for the duration of the retention period. This protection applies to the key vault and the secrets stored within the key vault.

To enforce a mandatory retention period and prevent the permanent deletion of key vaults or secrets prior to the retention period elapsing, you can turn on purge protection. When purge protection is enabled, secrets cannot be purged by users or by Microsoft.

Soft-delete ⓘ	Enabled
Days to retain deleted vaults * ⓘ	<div>90</div>
Purge protection ⓘ	<div><input checked="" type="radio"/> Disable purge protection (allow key vault and objects to be purged during retention period)</div> <div><input type="radio"/> Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)</div>







- For “convenience,” Blackridge added **everyone** to Secrets Officer
- Add role assignment ...

Role Members Conditions [Review + assign](#)

**Selected role** Key Vault Secrets Officer

**Assign access to** ☒ User, group, or service principal  
☐ Managed identity


**Members** [+ Select members](#)

Name	Object ID	Type	
Sandra Banks	c9d0bda0-94f9-4050-bf9d-9c9621b2d3...	User	
Jane Doe	dc0b348c-b0c0-4da1-b2e3-eb39f93a62f1	User	
CloudOps Engineer	1be34dab-a538-42ee-94c6-9cf7e2830e7f	User	
Cloud Administrator	1d2c1c2a-2241-4de8-a0c5-66ce49dfd99	User	
AD Admin	386aa480-0675-4d6c-970d-4c0c520a82...	User	
Denney Thongphet(Guest)	ac1ffce9-43cd-487c-85a6-7bf0cec2b279	User	

**Description**

Optional

- Added 3 fake secrets:
  - slack\_token
  - prod\_db\_pass
  - azure\_key\_pair

 **Create a secret** ...

Upload options

Manual

Name \* ⓘ

slacktoken

Secret value \* ⓘ

.....

Content type (optional)

Internal chat access

Set activation date ⓘ

☒

Set expiration date ⓘ

☐

Enabled

Yes

No

Tags

0 tags

Name	Type	Status
azurekeypair	Cross-cloud credential	✓ Enabled
prodbpass	Production DB root	✓ Enabled
slacktoken	Internal chat access	✓ Enabled

## 🔴 Dropped .ps1 Script on DC01

- File path: `C:\ITS_Tools\CloudSync_Backup.ps1`
- Hardcoded credentials + Azure login command
- This is what Eliot finds during Phase 2 and uses to pivot into Azure

```
CloudSync_Backup.ps1 - Notepad
File Edit Format View Help
# Blackridge Cloud Sync Utility (Unsecured)
# Last updated: 2025-04-21

# ----- Configuration -----
$Username = "cloudadmin@denneythogmail.onmicrosoft.com"
$Password = "Blackridge123!"
$VaultName = "blackridge-vault"
$SecretName = "prodbpass"
$Container = "financial-data"
$StorageAcct = "blackridgefinance"

# ----- Simulate Azure Login -----
Write-Host "[INFO] Logging into Azure with provided credentials..." -ForegroundColor Cyan
az login --username $Username --password $Password | Out-Null

# ----- Download Blob from Storage -----
Write-Host "[INFO] Syncing finance container to local backup directory..." -ForegroundColor Cyan
$LocalPath = "C:\CloudSync\Finance"
New-Item -ItemType Directory -Path $LocalPath -Force | Out-Null
az storage blob download-batch `
    --destination $LocalPath `
    --source $Container `
    --account-name $StorageAcct

# ----- Retrieve DB Password from Vault -----
Write-Host "[INFO] Pulling production DB password from Key Vault..." -ForegroundColor Cyan
az keyvault secret show `
    --vault-name $VaultName `
    --name $SecretName `
    --query value `
    --output tsv

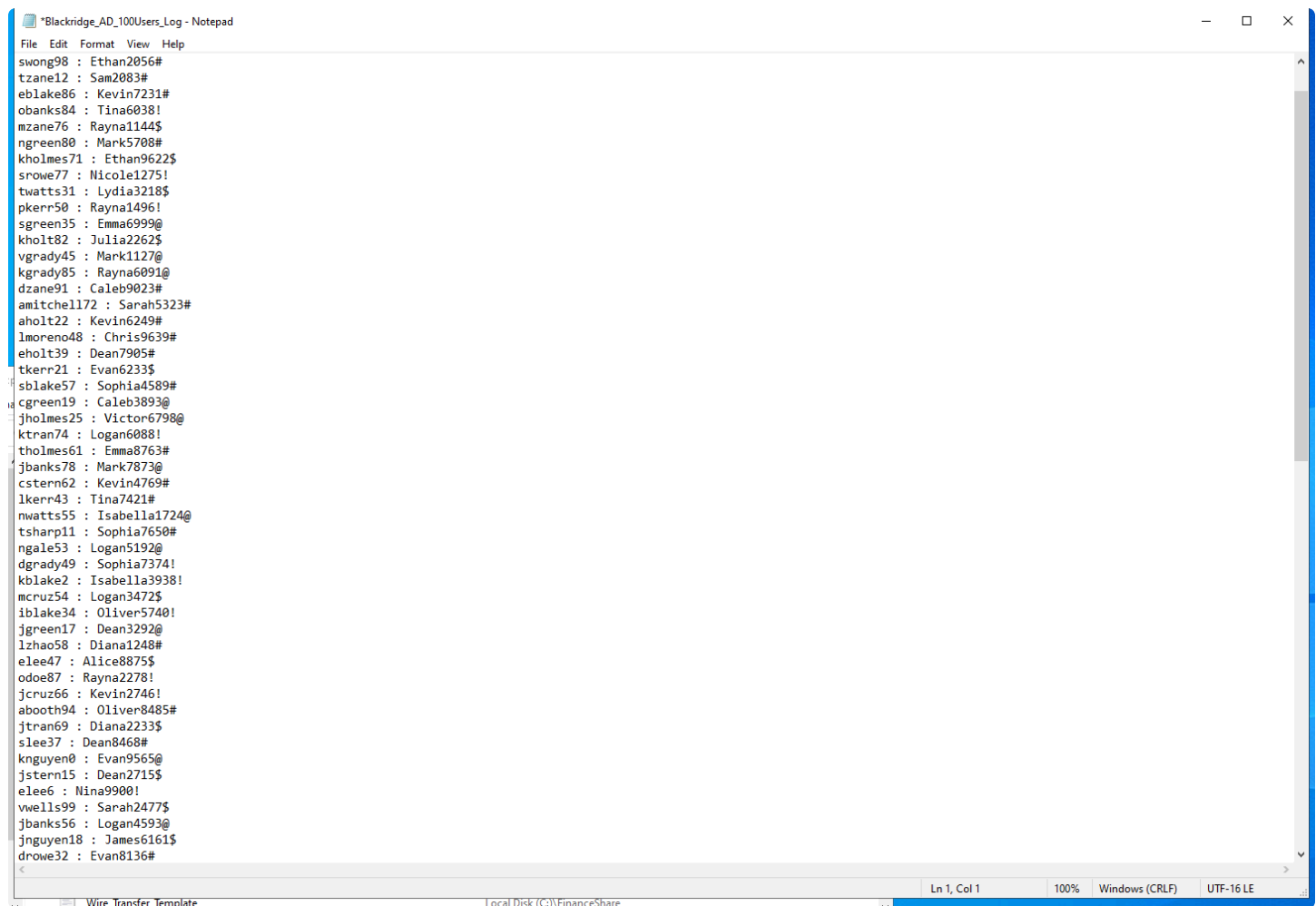
# ----- Final Log -----
Write-Host "[INFO] Backup completed. Files stored locally at $LocalPath" -ForegroundColor Green
```

To prep for Phase 2, I intentionally added weak configurations and poor security hygiene — the kind of stuff you still see in real companies. These open up realistic attack paths for a rogue insider to abuse.

## Weak, Human-Guessable Passwords

- User passwords follow easy-to-guess patterns (e.g., Kevin6398# , Rayna6091@ )
- Most are just names + birth years, reused across accounts
- No real randomness or complexity enforcement
- Passwords are stored in plaintext in:

C:\Blackridge\_AD\_100Users\_Log.txt

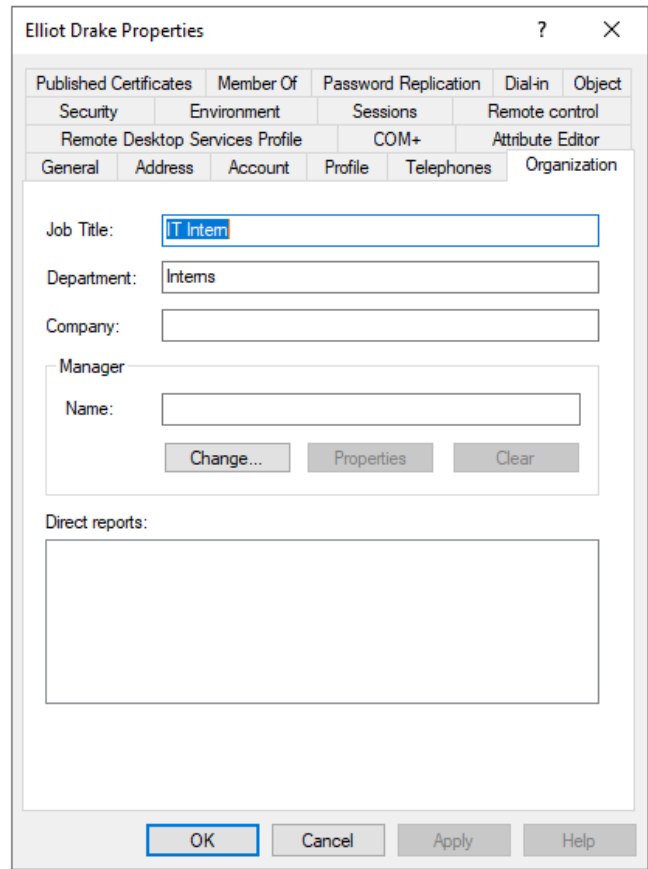
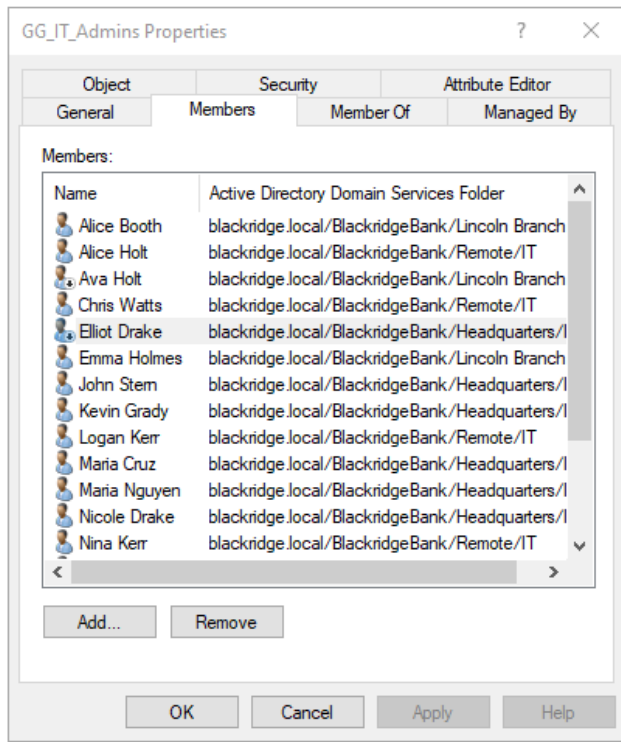


```
*Blackridge_AD_100Users_Log - Notepad
File Edit Format View Help
swong98 : Ethan2056#
tzane12 : Sam2083#
eblake86 : Kevin7231#
obanks84 : Tina6038!
mzane76 : Rayna1144$
ngreen80 : Mark5708#
kholmes71 : Ethan9622$
srowe77 : Nicole1275!
twatts31 : Lydia3218$
pkerr50 : Rayna1496!
sgreen35 : Emma6999@
kholt82 : Julia2262$
vgrady45 : Mark1127@
kgrady85 : Rayna6091@
dzane91 : Caleb9023#
amitchell172 : Sarah5323#
aholt22 : Kevin6249#
lmoreno48 : Chris9639#
eholt39 : Dean7905#
tkerr21 : Evan6233$
sblake57 : Sophia4589#
cgreen19 : Caleb3893@
jholmes25 : Victor6798@
ktran74 : Logan6088!
tholmes61 : Emma8763#
jbanks78 : Mark7873@
cstern62 : Kevin4769#
lkerr43 : Tina7421#
nwatts55 : Isabella1724@
tsharp11 : Sophia7650#
ngale53 : Logan5192@
dgrady49 : Sophia7374!
kblake2 : Isabella3938!
mcruz54 : Logan3472$
iblake34 : Oliver5740!
jgreen17 : Dean3292@
lzhao58 : Diana1248#
elee47 : Alice8875$
odoe87 : Rayna2278!
jcruz66 : Kevin2746!
abooth94 : Oliver8485#
jtran69 : Diana2233$
slee37 : Dean8468#
knguyen0 : Evan9565@
jstern15 : Dean2715$
elee6 : Nina9900!
vwell99 : Sarah2477$
jbanks56 : Logan4593@
jnguyen18 : James6161$
drowe32 : Evan8136#
```

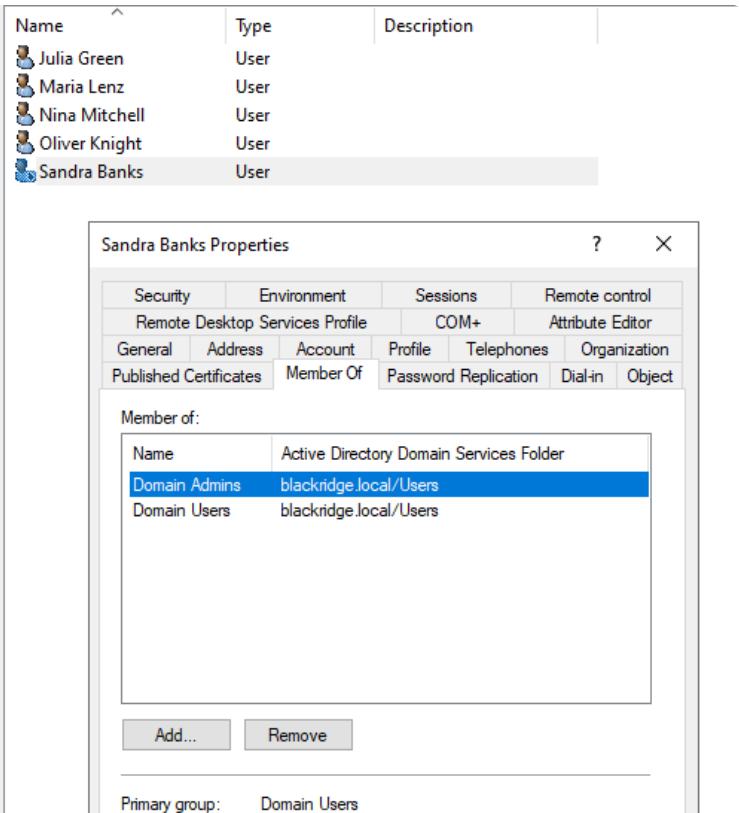
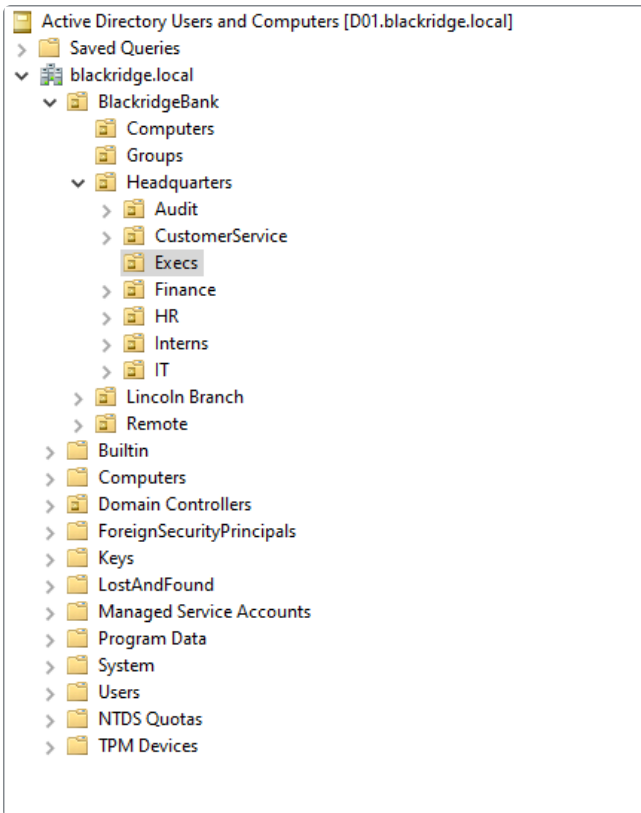
## Overprivileged Accounts



- **Eliot Drake ( edrake0 )**, an intern in IT, was added to **GG\_IT\_Admins**



- **Sandra Bank ( sbanks )**, an executive, was added to **Domain Admins**  
(Too much power for non-technical staff)



## Excessive Access via GPOs (Planned)

Will deploy GPOs that:

- Map sensitive folders (like `ITS_Tools`) to **all users**, regardless of role
- Enable **PowerShell and RDP access** across the board
- Skip proper access control — no separation of duties or least privilege enforcement

This simulates sloppy internal controls often seen in rushed setups or during org growth.

---

## Poor Logging & Monitoring (In Progress)

- No Sysmon or advanced logging installed
- PowerShell transcription logging is disabled
- No centralized logging, no SIEM
- Audit policy is default — no alerts, no correlation
- Event logs can roll over or be cleared without detection

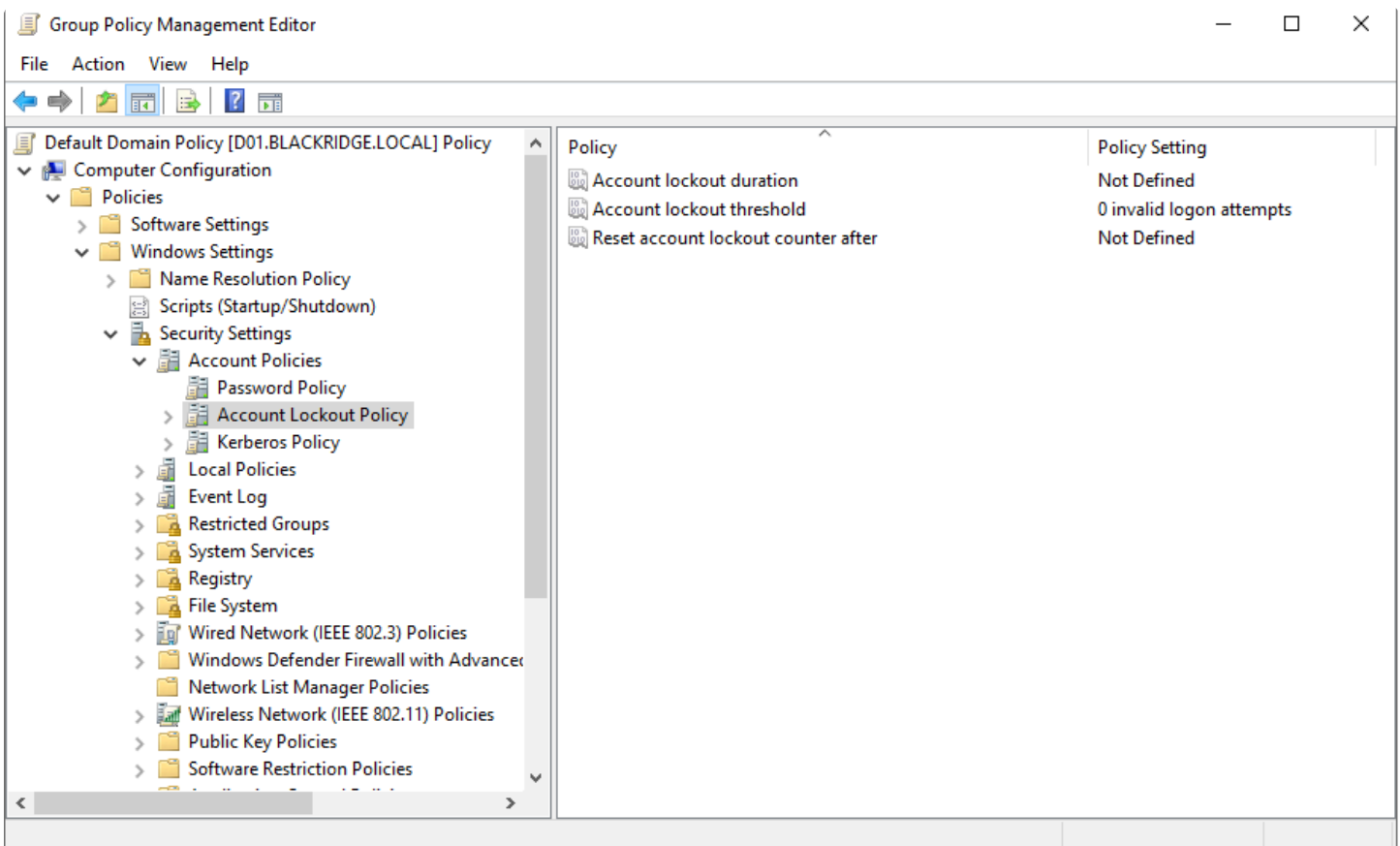
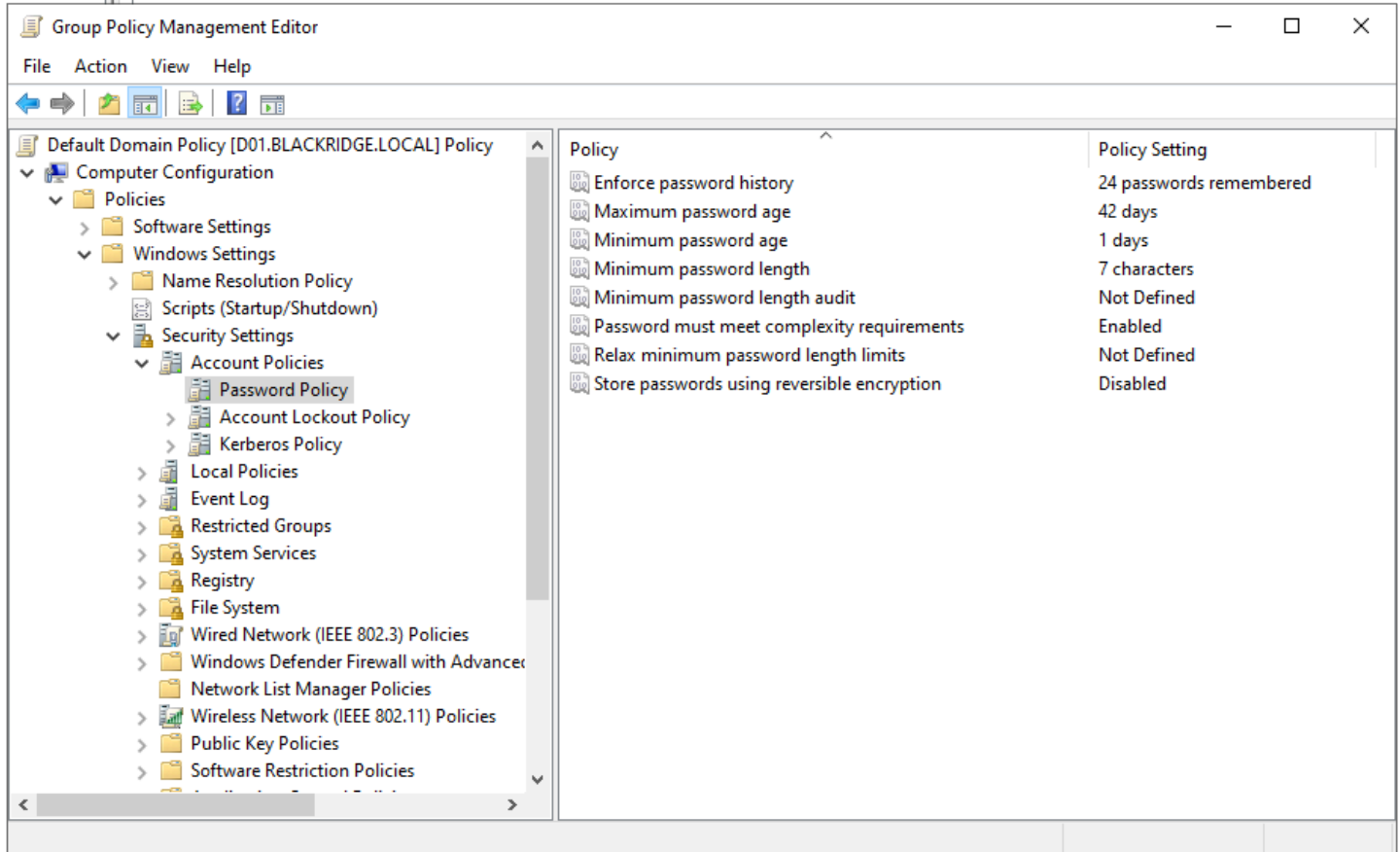
This sets the stage for stealthy abuse and delayed incident response.

---

## Weak Password Policy

Default domain policy is still mostly untouched. It allows:

- Short passwords (min = 7 characters)
- No account lockout — unlimited login attempts
- Password history is set (24 remembered), but still vulnerable due to the simple patterns allowed



## Cloud-Side Weaknesses

- No MFA Anywhere**

Not enforced on any account, including Global Admins and Key Vault users.

Conditional Access | Policies

Microsoft Entra ID

Overview Policies Insights and reporting Diagnose and solve problems Manage Monitoring Troubleshooting + Support

New policy New policy from template Upload policy file What if Preview features Got feedback?

Create your own policies and target specific conditions like cloud apps, sign-in risk, and device platforms with Microsoft Entra ID Premium.

### What is Conditional Access?

Conditional Access gives you the ability to enforce access requirements when specific conditions occur. Let's take a few examples [Learn more](#)

Conditions	Controls
When any user is outside the company network	They're required to sign in with multifactor authentication
When users in the 'Managers' group sign-in	They are required be on an Intune compliant or domain-joined device

### Get Started

1. Create your first policy by clicking "+ Create new policy"
2. Specify policy Conditions and Controls
3. When you are done, don't forget to Enable policy and Create

[Interested in common scenarios?](#)

- **Overpowered Group: CloudAdmins**
  - Everyone was added to this group: finance, execs, interns, IT
  - Given **Global Administrator** permissions
  - No scoping or access separation
- **Public Blob Storage**
  - Anonymous access was left on for `financial-data`
  - Anyone with the URL can view/download sensitive finance docs
- **Hardcoded Azure Credentials in PowerShell Script**
  - Leaked in `C:\ITS_Tools\CloudSync_Backup.ps1`
  - Contains cloudadmin credentials and Azure login logic
- **Key Vault Exposed**
  - `blackridge-vault` was created without RBAC scoping
  - All CloudAdmins are Secrets Officers
  - No expiration or secret rotation
- **Reused Identities Between On-Prem and Cloud**
  - Example: `azureadmin@blackridge.local` exists in both environments with the same password
  - No federation or lifecycle management — just copy-paste identity reuse
- **No Detection in Azure**
  - No Defender for Cloud
  - No Key Vault logging
  - No access review policies
  - No alerting on privilege abuse or sign-in anomaly

These weaknesses will fuel red team escalation paths in Phase 2 and 3 — allowing an intern or compromised account to slowly climb the ladder, evade detection, and access sensitive systems.

