Silverfort
Test Flight

Silverfort Test Flight Exercise Lab Guide

2025

Table of Contents

Introduction **3**

What is Needed **3**

What's in the Silverfort Test Flight Lab **4**

Access and Credentials **5**

The Lab **5**

Lab Exercises **6**

Exercise 1 - MFA for RDP **7**

Create the RDP Policy *7*

Testing the RDP Policy *9*

Policy Validation Using Silverfort Logs *10*

Exercise 2 - MFA for PowerShell **11**

Create the PowerShell Policy *11*

Testing the PowerShell Policy *12*

Policy Validation Using Silverfort Logs *13*

Exercise 3 - Management Console MFA **14**

Create the Management Console MFA Policy .. *14*

Testing the Management Console Policy *15*

Policy Validation Using Silverfort Logs *15*

Exercise 4 - MFA for Privilege Escalation Scenarios **16**

Create the Privilege Escalation MFA Policy *17*

Testing the Privilege Escalation Policy *18*

Policy Validation Using Silverfort Logs *18*

Exercise 5 - Service Account Discovery & Virtual Fencing **18**

Discovery of Service Accounts *20*

Part 1: Service Account Discovery 20

Part 2: Silverfort Virtual Fencing 21

Policy Validation Step #1 23

Silverfort Virtual Fencing - Enforcement Mode *23*

Test Virtual Fencing - Enforcement Mode 23

Exercise 6 - Risk-Based Policy **24**

Building Silverfort Policy Using Logs (optional deep dive) *25*

Testing Your New Policy *26*

Policy Validation Using Silverfort Logs *27*

Dynamic Step-Up MFA *27*

Create a Risk-Based Policy *27*

Testing the Risk-Based Policy *29*

Simulate a Security Event: *30*

Exercise 7 - Privileged Access Security **31**

Part 1: Privileged Access Classification *31*

Defining Tier 0 Assets 32

Part 2: Privileged Access Security User Virtual Fencing *33*

Part 3: Privileged Access Security Just-In-Time (JIT) *38*

Lab summary - Key Takeaways **44**

Thank you for completing the Silverfort Test Flight Lab Guide! Your journey into next-generation identity security starts here. **44**

Introduction

Welcome to Silverfort's Test Flight! This is your hands-on playground to get familiar with Silverfort and experience its core functionality firsthand. Think of this as your "sandbox" before implementing these powerful security controls in your real environment.

This lab guidebook provides step-by-step exercises for configuring, testing, and validating that security policies work exactly as intended. Each exercise builds on the previous one, taking you from basic MFA protection to advanced identity segmentation.

What You'll Learn: How to stop the attack techniques used in 95% of successful breaches - from credential theft to lateral movement to privilege escalation.

What is Needed

Very little, but you will need:

- ✓ Your own laptop (Windows, Mac, Chromebook or Linux)
- ✓ A smartphone (highly recommended for the full MFA experience)

For Lab Access: Test Flight uses a virtual desktop from which all exercises run.
You'll access this through:

- ✓ A web browser (Chrome recommended)
- ✓ Remote Desktop Client supporting Microsoft Remote Desktop Services
- ✓ The native Remote Desktop client in Windows also works perfectly

For MFA Testing: While optional, a smartphone gives you the complete Silverfort experience. You'll use MFA push notifications to see how seamless security can be.

 **Supported Devices:** iPhone, iOS, and Android devices

 **App Store:** Download "Silverfort" from Apple App Store or Google Play Store

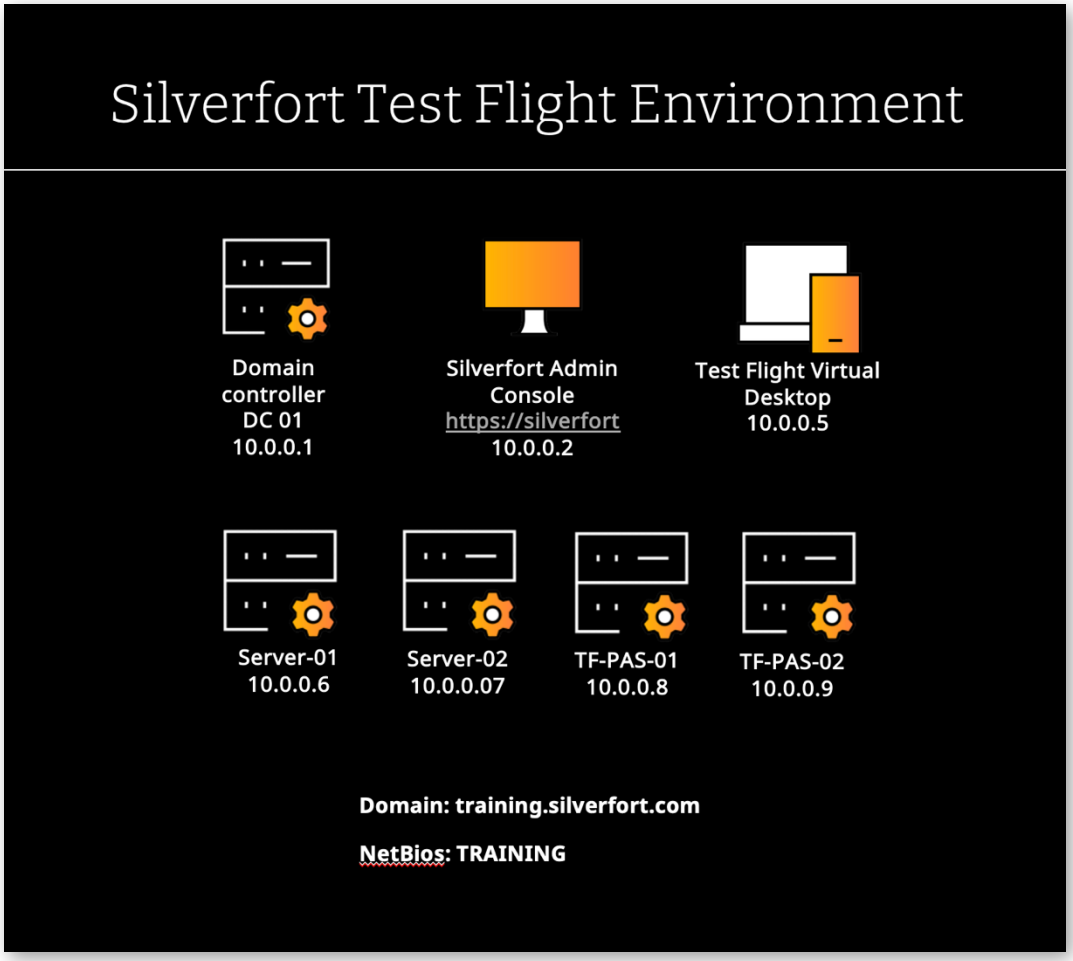
 **Easy Cleanup:** Safely remove the app after the workshop

What's in the Silverfort Test Flight Lab

The lab includes a preconfigured Active Directory environment with Silverfort already integrated and ready to use. Below is the network diagram showing what you'll be working with.

Lab Environment Includes:

- An Active Directory domain with realistic user accounts
- Multiple member servers (Server-01, Server-02, TF-PAS-01, TF-PAS-02)
- A fully configured Silverfort platform
- Simulated attack scenarios
- Real-world service accounts and privileged users



Access and Credentials

Your Unique Lab Setup Each lab session is potentially unique, so you'll receive these at the start:

1. **Web/Remote Desktop Address** - The URL to access your virtual lab
2. **Your Silverfort Lab Username & Password** - Your unique credentials
3. **Domain Format** - Use your standard account like training\maverick
4. **Additional Test Accounts** - Provided upon need for specific exercises

Important Notes:

- Your "standard" username works for Windows desktops, services and **the Silverfort admin console**
- Don't use 'adm' or 'svc' prefixed accounts unless specifically told to in the exercise
- Access the admin console through Chrome browser from the lab desktop
- Username format example: training\maverick

Getting Started Checklist:

- Verify you can access the lab's remote desktop immediately
- Don't worry if you see "Access Denied" initially - this is often intentional
- Install Silverfort MFA app on your phone
- Have your credentials ready

The Lab

If you have an instructor, they'll walk you through:

- **Main Dashboard** - Your command center for monitoring & policies
- **System Status** - How Silverfort integrates with Active Directory
- **Logs & Filtering** - Essential for validating your work (you'll use this a lot!)
- **Policy Engine** - Where the magic happens
- **MFA User Enrollment** - Getting your phone connected

Pro Tip: This is the perfect time to install the Silverfort MFA app and get familiar with the interface before diving into exercises.

Lab Exercises

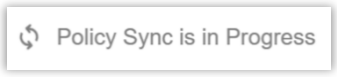
The Journey Ahead: These exercises follow a logical progression - each one builds your understanding while demonstrating real-world attack prevention:

Exercise	Focus	Time	Why It Matters
1-4	MFA Protection	5-10 min each	Stop credential theft & misuse
5	Service Account Security	15 min	Prevent lateral movement
6	Risk-Based Controls	10 min	Adaptive security based on behavior
7	Identity Segmentation	20 min	Ultimate privilege protection

Real-World Context: These exercises simulate the exact attack paths used in major breaches. By the end, you'll have built defenses against the most common attack techniques.

Lab Etiquette for Shared Environment:

- Prefix all policies with your username (e.g., "Maverick - RDP MFA")
- Wait for "Policy Sync is in Progress" to complete before testing



- Allow 30-45 seconds after clicking 'Save' before testing your policy
- Be mindful that others are working in the same environment

Exercise 1 - MFA for RDP

Estimated Time: 10 minutes

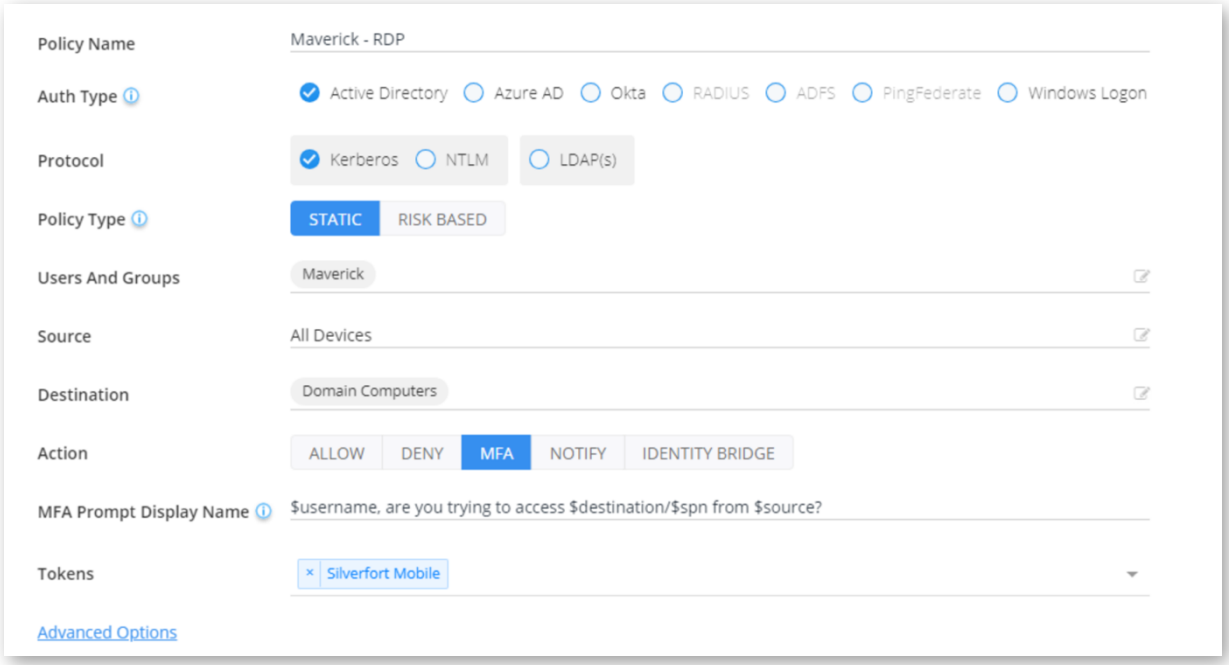
Why This Matters

Remote Desktop Protocol (RDP) is used in **70% of ransomware attacks** as the initial entry point. Attackers love RDP because once they're in, they can access critical resources and steal credentials. This exercise shows how MFA can slam the door on this attack vector.

Real Attack Scenario: An attacker compromises a user's password through phishing. Without MFA, they can RDP directly into your servers and begin their attack. With Silverfort MFA, that stolen password becomes useless.

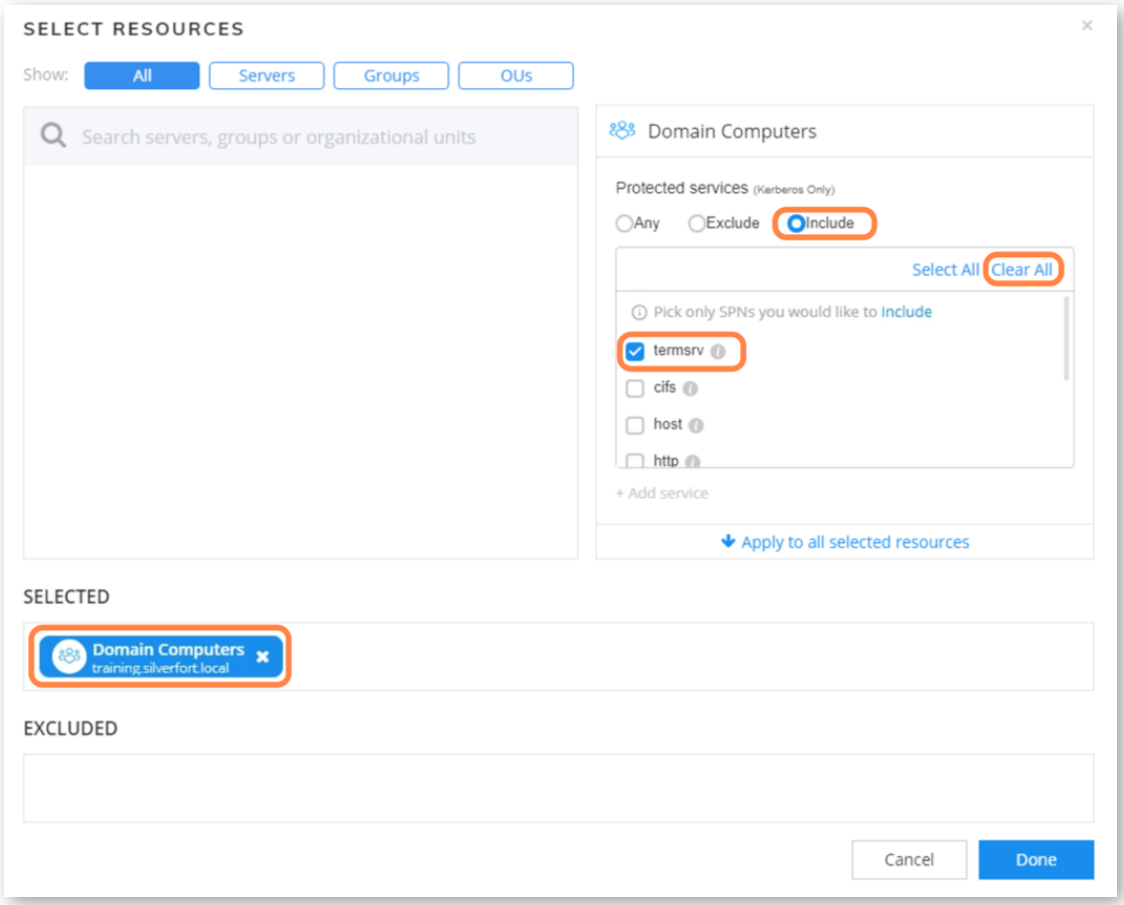
In this lab: You'll create a targeted MFA policy that protects all Remote Desktop connections for your lab user. You'll configure the policy to specifically detect RDP authentication attempts using the "termsrv" service, test the protection by connecting to a remote server, and validate that MFA is properly enforced by reviewing the authentication logs.

Create the RDP Policy



Follow these steps to create your protection policy:

1. Open the **Policies** dashboard from the left menu
2. Click "**+ Create Policy**" in the top right
3. **Policy Name:** Enter YourUsername - RDP MFA (e.g., "Maverick - RDP MFA")
4. **Auth Type:** Select Active Directory
5. **Protocol:** Select Kerberos
6. **Policy Type:** Select STATIC
7. **Users And Groups:** Find and enter your Silverfort username
8. **Source:** Select All Devices
9. **Destination:** Configure for RDP protection:
    - Choose "**Include**"
    - Click "**Clear All**"
    - Select "**termsrv**" (this is the RDP service)
    - Click "**Done**"



10. **Action:** Select MFA
11. **MFA Prompt Display Name:** Enter your custom message or leave default
12. **Tokens:** Leave as Silverfort Mobile
13. Click "**Save**" and "**OK**" to activate the rule

Testing the RDP Policy

Time to see your protection in action!

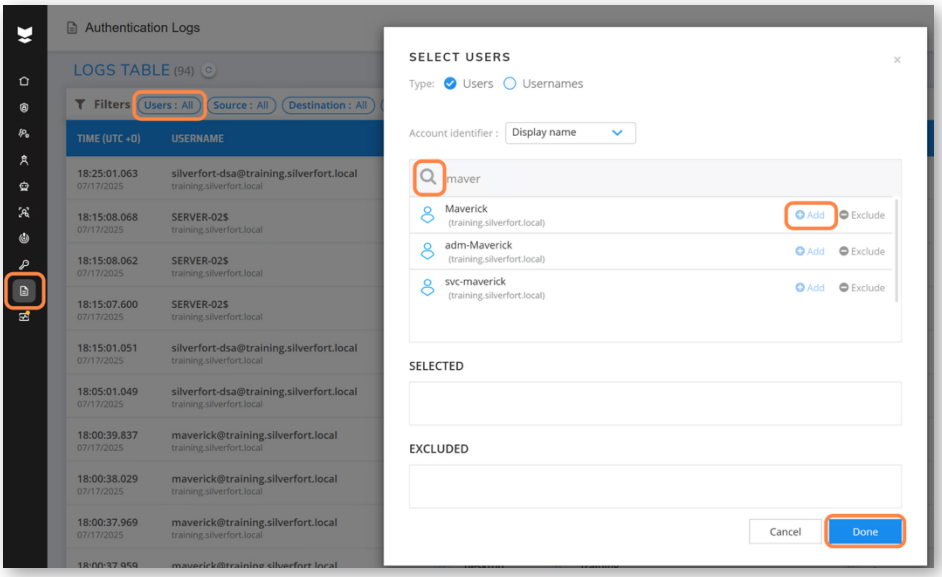
1. Open the **Labs Folder** shortcut on the desktop
2. Run the "**Exercise 1 - RDP**" application
3. Click "**Connect**" and enter your lab user password
4. **Check your phone!** Approve or deny the MFA request
5. When finished, sign off (don't just disconnect)

What Should Happen: You should see an MFA prompt on your phone asking you to approve the RDP connection. This is Silverfort in action!

Policy Validation Using Silverfort Logs

Let's verify your policy worked correctly:

1. Click "**Authentication Logs**" in the left menu
2. Filter for your activity:
    - Click "**Users**" in the filter bar
    - Type your lab username (e.g., "maverick")
    - Click "**Add**" then "**Done**"



3. Look for the log entry with Silverfort Action = "**MFA Approved**"
4. Notice the destination shows **termsrv** - that's RDP protection working
5. Expand the entry to see detailed info including:
    - Policy that was matched
    - MFA token used
    - Your response time

Success Criteria: You should see a log entry showing MFA was required and approved for your RDP connection.

Exercise 2 - MFA for PowerShell

Estimated Time: 10 minutes

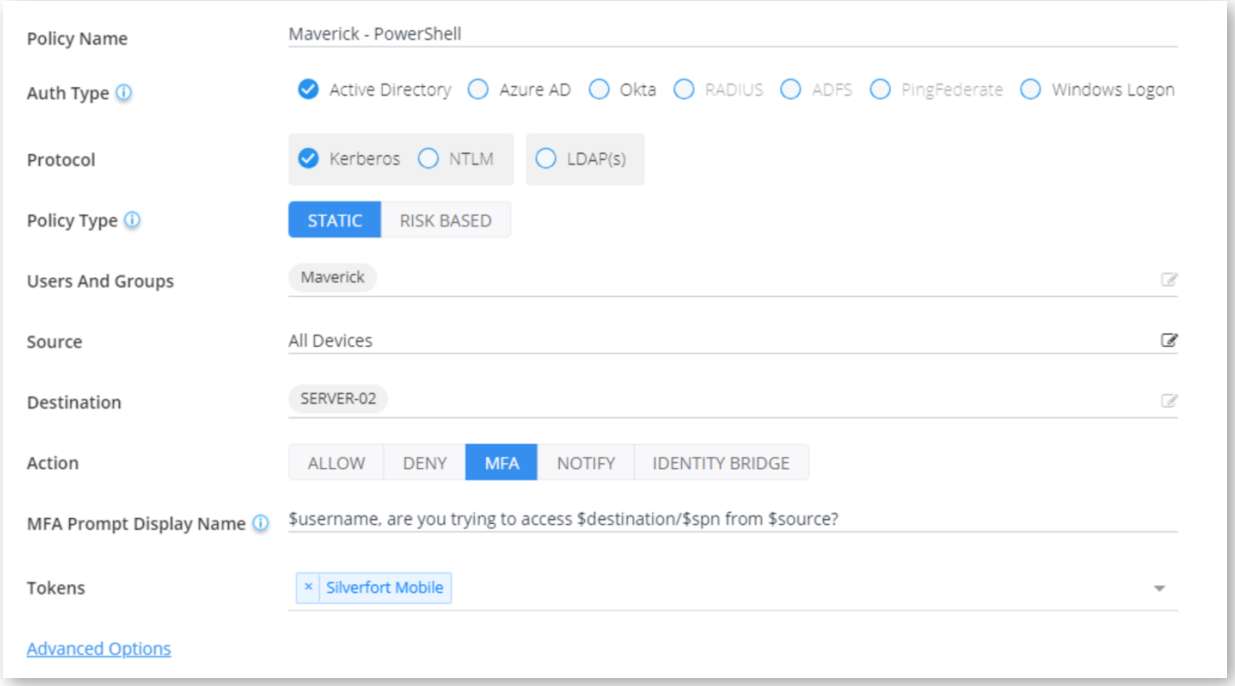
Why This Matters

PowerShell and remote command shells are the **#1 tool for lateral movement** in enterprise attacks. Once attackers have initial access, they use PowerShell to enumerate your network, steal more credentials, and move to critical systems. This is incredibly hard to detect with traditional tools.

Real Attack Scenario: After gaining access to one machine, attackers use PowerShell to remotely execute commands on other servers, gradually working their way to domain controllers and sensitive data. Silverfort can stop this technique immediately.

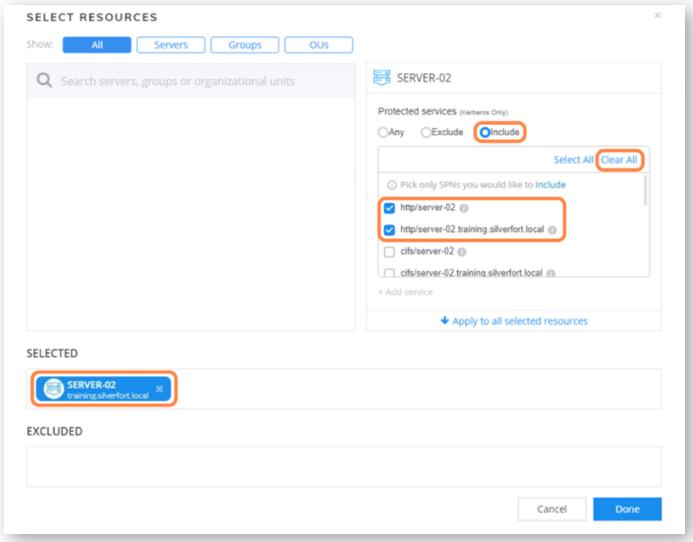
In this lab: You'll build a policy that requires MFA for PowerShell remoting sessions by targeting HTTP protocol communications (which PowerShell uses for WinRM). You'll configure protection for Server-02, test remote PowerShell execution that triggers your MFA policy, and observe how Silverfort captures the session details including Kerberos tickets and remote execution context.

Create the PowerShell Policy



1. **Policy Name:** Enter “YourUsername - PowerShell MFA” (e.g., "Maverick - PowerShell MFA")
2. **Auth Type:** Select Active Directory

3. **Protocol:** Select Kerberos
4. **Policy Type:** Select STATIC
5. **Users And Groups:** Find and select your Silverfort username
6. **Source:** Select All Devices
7. **Destination:** Configure for PowerShell protection:
    - Select "**Server-02**"
    - Choose "**Include**"
    - Click "**Clear All**"
    - Select "**http**" (PowerShell remoting uses the HTTP SPN)
    - Click "**Done**"



8. **Action:** Select MFA
9. **MFA Prompt Display Name:** Customize or leave default
10. **Tokens:** Leave as Silverfort Mobile
11. Click "**Save**" and "**OK**" to activate

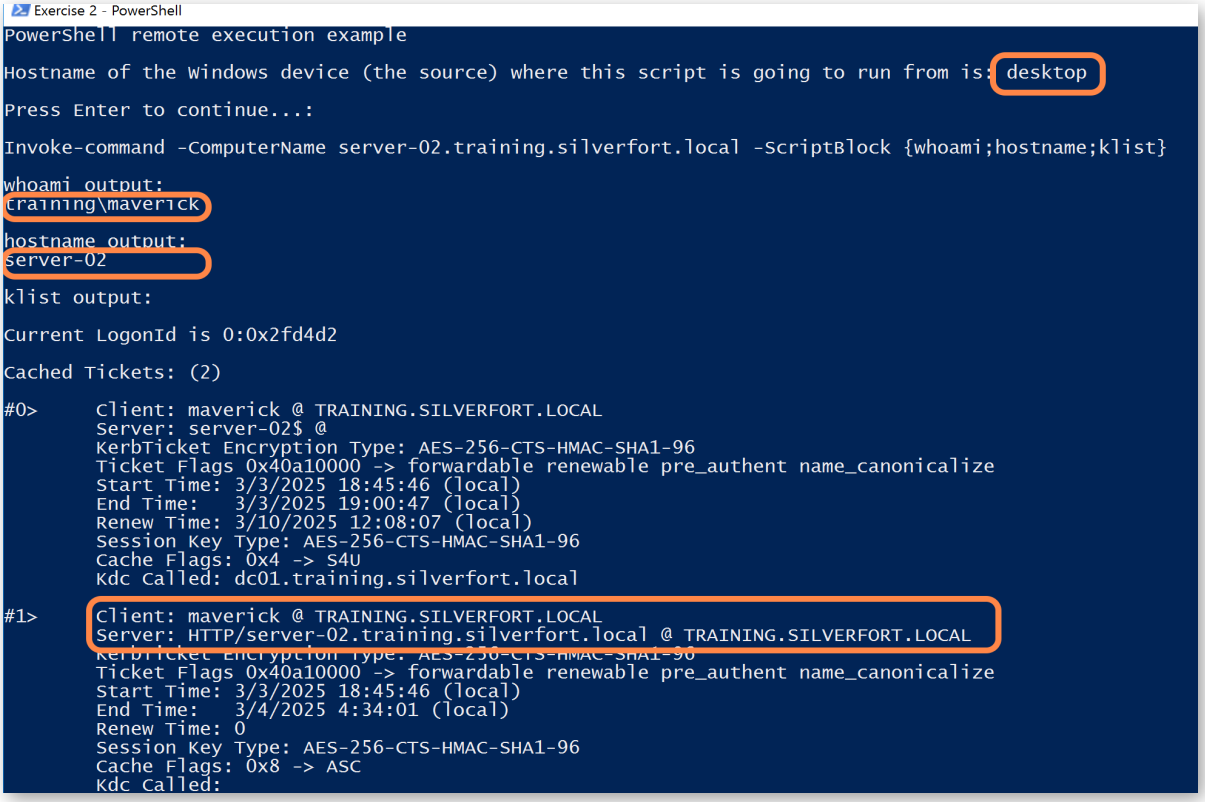
Testing the PowerShell Policy

1. Open **Labs Folder** from desktop
2. Double-click "**Exercise 2 - PowerShell**" application
3. **Check your phone** for the MFA challenge
4. Approve the request

Expected Output: You should see something like this showing the remote PowerShell session details.

The output shows:

- Hostname of the server you're connecting from
- Your user identity
- Kerberos tickets created for the session



Policy Validation Using Silverfort Logs

Follow the same log validation steps from Exercise 1, but this time:

- Look at the Destination column, see HTTP (the service) in the logs
- Try filtering by your policy name
- Notice how Silverfort caught the PowerShell remoting attempt

Troubleshooting: If you don't receive MFA prompts, wait 30-45 seconds after saving the policy before testing.

Exercise 3 - Management Console MFA

Estimated Time: 10 minutes

Why This Matters

Management interfaces are **goldmines for attackers**. Once attackers compromise credentials, accessing admin consoles lets them change configurations, create backdoors, and disable security controls. Even your security tools need protection!

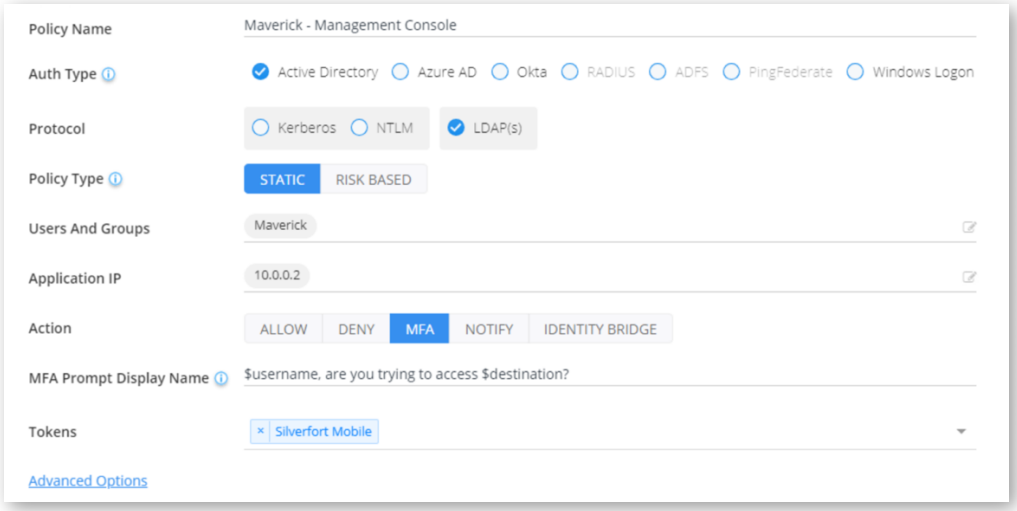
Real Attack Scenario: Attackers compromise admin credentials through credential stuffing or social engineering, then access management consoles to disable monitoring, create persistent access, or steal sensitive configuration data.

Protecting management platforms, application interfaces, and web applications reduces attack surface and prevents threat actors from operating even after gaining initial access. Through controls like access segmentation and MFA, Silverfort stops enumeration, redirection, data exfiltration, and system disruption attacks. Since Silverfort integrates into authentication workflows to protect resource access identities, securing the Silverfort Management console itself is equally crucial.

In this lab: You'll build a policy requiring your lab user to use MFA for Silverfort console access and customize MFA messages for push notifications and the Silverfort desktop client.

Create the Management Console MFA Policy

This exercise protects access to the Silverfort console itself.



1. **Policy Name:** YourUsername - Silverfort Console MFA (e.g., "Maverick – Silverfort Console MFA")
2. **Auth Type:** Select Active Directory

3. **Protocol:** Select LDAP(s) (management consoles often use LDAP auth)
4. **Policy Type:** Select STATIC
5. **Users And Groups:** Find and enter your Silverfort username
6. **Application IP:**
    - Click "Find" and enter the Silverfort Admin Console IP address

      This IP was provided earlier in your lab credentials
    - Click "**Add**"
7. **Action:** Select MFA
8. **MFA Prompt Display Name:** Try a custom message like:
9. "Are you accessing the Silverfort Management Console? If not, DENY and report immediately!"
10. **Tokens:** Leave as Silverfort Mobile
11. Click "**Save**" and "**OK**"

Testing the Management Console Policy

1. Open **Labs Folder** from desktop
2. Double-click "**Exercise 3 - Silverfort Admin Console**"
3. This opens an Incognito Chrome session
4. Authenticate with your lab credentials (format: domain\username or username@domain.name)
5. **Check your phone** - you should see your custom MFA message!
6. Approve the request to access the console

What's Happening: Silverfort is now protecting its own management interface. This creates layered security - even if someone gets admin credentials, they still need the second factor.

Policy Validation Using Silverfort Logs

Use the Authentication Logs to verify:

- Look for LDAP protocol entries
- Find your custom policy name in the matched policies
- Notice the Application IP in the destination details

Exercise 4 - MFA for Privilege Escalation Scenarios

Estimated Time: 15 minutes

Why This Matters

Privilege escalation is involved in 95% of successful breaches. Even with good hygiene (separate admin accounts), attackers can still abuse the "Run as Administrator" function to elevate privileges. This exercise shows how to protect that critical moment when privileges are elevated.

Real Attack Scenario: An attacker compromises a regular user account. They discover the user has a separate admin account (good practice!), steal those credentials, and use "Run as Administrator" to elevate privileges and install malware or access sensitive systems.

Best Practice Context: Users should have:

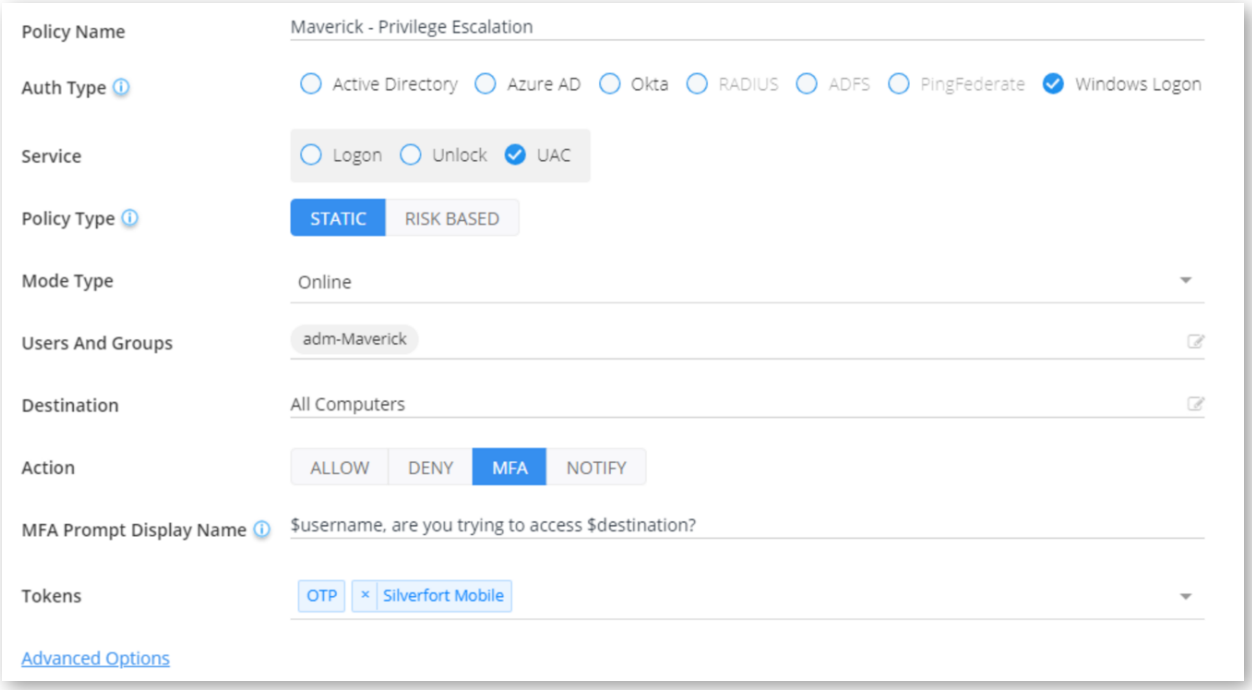
- Regular account for daily activities (limited privileges)
- Separate admin account used ONLY when administrative access is needed
- This exercise protects that elevation process

Silverfort can be added to Windows Desktops and Servers as a credential provider and then apply the same access controls, including using MFA to protect the use of the privileged accounts for this use case.

In this lab: You'll configure Silverfort as a Windows credential provider to intercept privilege escalation attempts. Using your regular account and separate "adm-" privileged account, you'll create a policy targeting Windows Logon authentication (excluding regular logon/unlock to focus only on elevation). You'll test the protection by attempting to "Run as Administrator" with your privileged credentials, experience the MFA challenge during elevation, and validate that Silverfort successfully distinguishes between regular authentication and privilege escalation events.

You will have been provided with a “privileged” lab user account for this exercise. For example, if your lab user is “Maverick", then your privileged lab user is “adm-maverick” The password will be the same, and your phone will still receive the MFA push notification.

Create the Privilege Escalation MFA Policy



1. **Policy Name:** YourUsername - Privilege Escalation (e.g., "Maverick – Privilege Escalation MFA")
2. **Auth Type:** Select Windows Logon (this catches local elevation)
3. **Service: Important!** Unselect "Logon" and "Unlock" - we only want elevation
4. **Policy Type:** Select STATIC
5. **Mode Type:** Select Online
6. **Users And Groups:** Find and enter your **privileged account** (e.g., "adm-maverick")
7. **Destination:** Leave as All Computers
8. **Action:** Select MFA
9. Click "**Save**" and "**OK**"

Testing the Privilege Escalation Policy

Part 1: Normal User (*Should Fail*)

1. Open **Labs Folder** from desktop
2. Double-click "**Exercise 4 - Privilege Escalation UAC**"
3. Enter your **standard user credentials** at the prompt
4. This opens Active Directory Users and Computers
5. Try to edit a user object and add a description
6. **Expected Result:** You can't make changes - insufficient privileges
7. Close the application

Part 2: Admin User (*Should Require MFA*)

8. Double-click "**Exercise 4 - Privilege Escalation UAC**" again
9. At the prompt, click "**More Choices**" → "**Use another account**"
10. Enter your **privileged account credentials** (e.g., "adm-maverick")
11. **Check your phone!** MFA should be required for the privilege escalation
12. Approve the MFA request
13. Now try to add a description to a user object 14.
14. **Expected Result:** You can make changes! But **don't save** - click Cancel

What Just Happened: Silverfort caught the moment when privileges were elevated and required MFA. The attacker's stolen admin credentials are now useless without the second factor.

Policy Validation Using Silverfort Logs

Check the logs for:

- Windows Logon authentication type
- Your privileged account name
- MFA Approved action
- Notice this was NOT a regular logon - it was privilege elevation

Exercise 5 - Service Account Discovery & Virtual Fencing

Estimated Time: 18 minutes

Why This Matters

Service accounts are the hidden highways of lateral movement. They're often highly privileged, rarely monitored, and perfect for attackers to abuse. Worse, traditional MFA doesn't work for service accounts since they're machine-to-machine communications.

Real Attack Scenario: An attacker compromises a service account and discovers it has access to multiple critical systems. They use these credentials to move laterally across your network, accessing databases, file servers, and eventually domain controllers. Service account compromise is involved in 80% of successful lateral movement attacks.

Silverfort's Unique Solution: Virtual Fencing creates invisible boundaries around service accounts, limiting where they can authenticate FROM and TO. If a service account suddenly tries to access unauthorized systems, Silverfort blocks it instantly. Silverfort discovers service accounts in real-time, maintaining visibility of where these accounts are used (source and destination) and identifying dormant accounts. This gives organizations confidence to perform password rotation without fear of breaking systems, since many don't know exactly where all service accounts are deployed. It also enables removal of unused accounts from Active Directory, reducing the attack surface.

In this lab: You'll experience Silverfort's real-time service account discovery by generating authentication activity and watching it appear instantly in the platform. You'll then create Virtual Fencing policies with both passive monitoring (notify-only) and active enforcement (blocking) modes. Using your designated service account, you'll test legitimate connections to multiple servers, configure allow/deny rules that intentionally create policy violations, and observe how Virtual Fencing prevents unauthorized lateral movement while maintaining legitimate service functionality.

This exercise demonstrates both monitoring for policy violations and enforcement to block unauthorized access, showing how organizations can gradually implement service account protection.

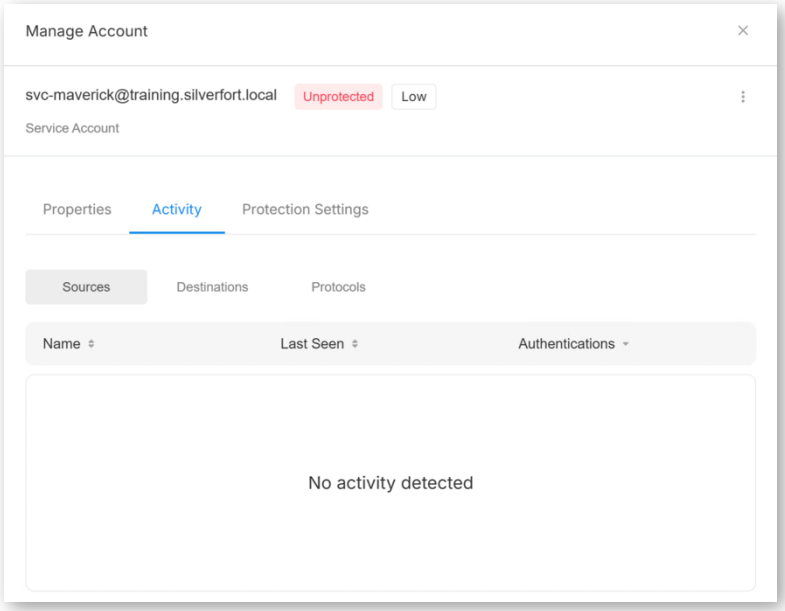
Discovery of Service Accounts

PART 1: SERVICE ACCOUNT DISCOVERY

Let's see Silverfort's real-time discovery in action:

1. Navigate to **Non-Human Identities** → **Active Directory**
2. Use "**Search Account Name**" to find your service account (e.g., "svc-maverick")

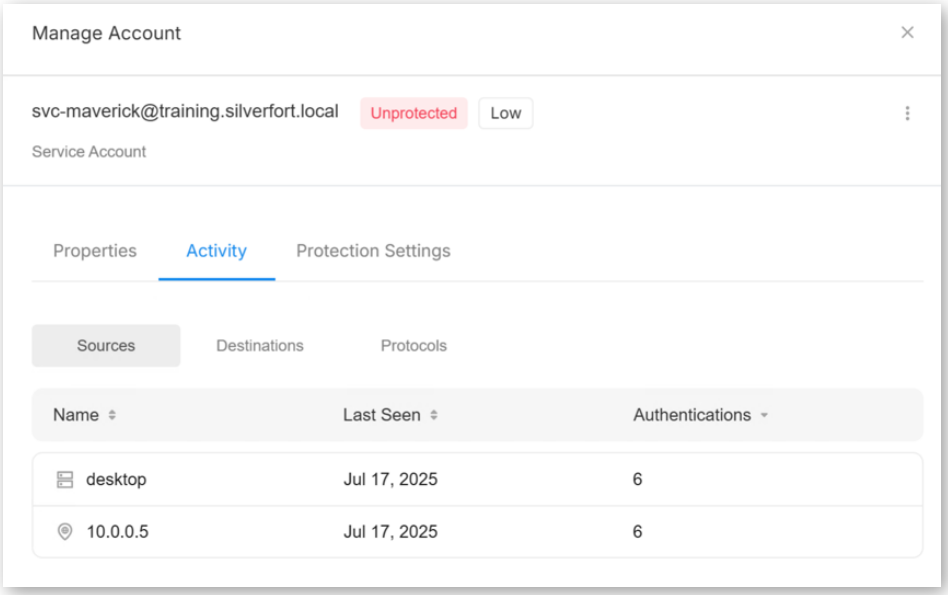
   Note: *Your account might be in the "Dormant" category initially*
3. Click on the service account name to open "**Manage Account**"
4. Click on the "**Activity**" tab
5. Look at the **Source**, **Destination**, and **Protocol** tabs - they should be empty



Now let's generate some activity:

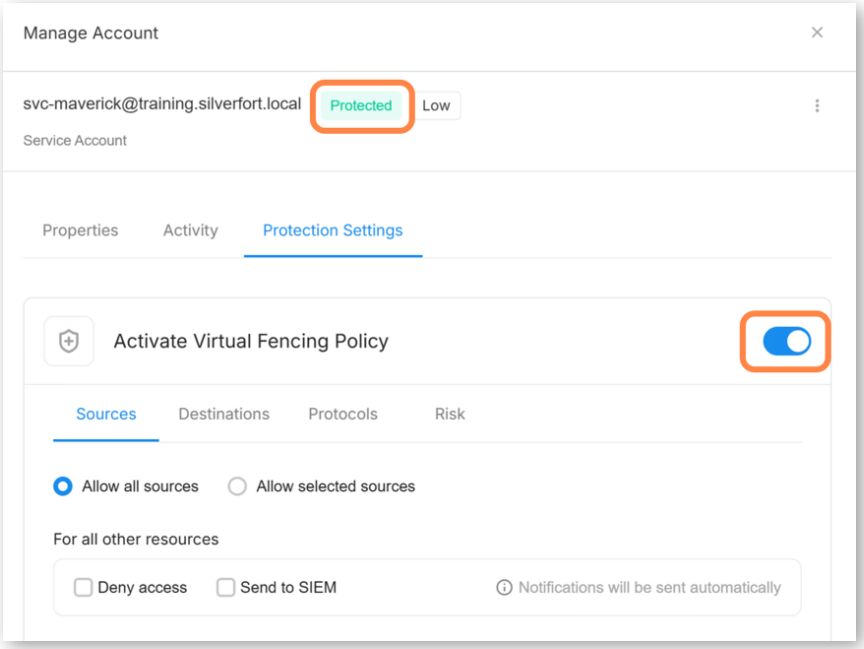
6. Open Labs Folder and run "Exercise 5 - Service Account"
7. Enter your service account credentials when prompted (e.g., "svc-maverick")
8. The script will attempt random connections to Server-01 and Server-02
9. Keep the script open! You'll use it again later
10. When complete, return to Silverfort dashboard
11. Watch Real-Time Discovery:
12. Close and reopen the "Manage Account" window (to refresh data)
13. Click "Activity" and check the Source, Destination, Protocol tabs

14. Amazing! Silverfort has automatically mapped where your service account is used. If you wish, you can look in the authentication logs for this activity as well, filtering for your service account.



PART 2: SILVERFORT VIRTUAL FENCING

1. In the **Manage Account** window, click "**Protection Settings**" tab
2. Move the slider to **ON** for "Active Virtual Fencing Policy"
3. Notice the account is now marked as **Protected**

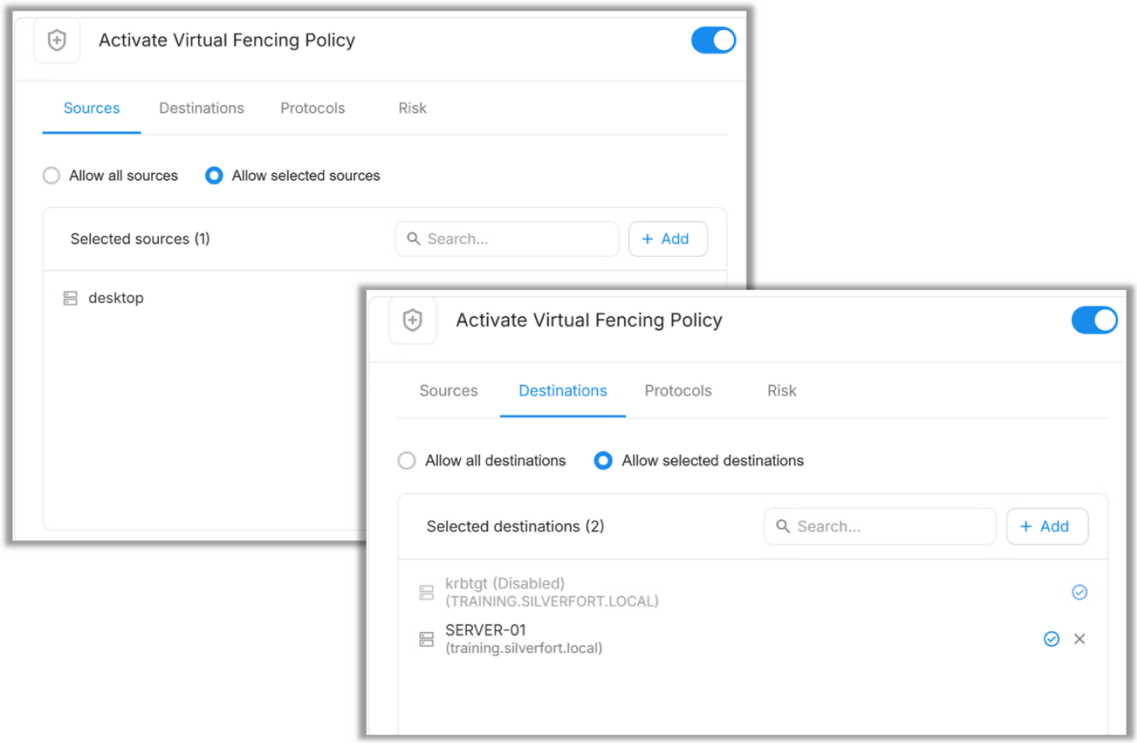


Configure Allowed Sources:

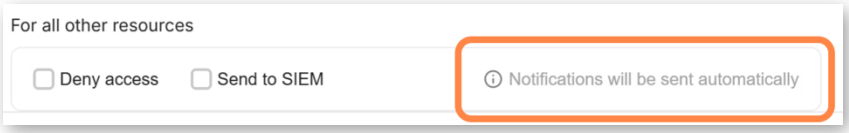
4. In the Sources view, select "Allow selected sources"
5. Click "+ Add" button
6. From dropdown, select "Detected by Silverfort"
7. Select "desktop" (where the service account legitimately runs from)
8. Click "Done"

Configure Allowed Destinations:

9. Click the "Destinations" tab
10. Select "Allow selected destinations"
11. Click "+ Add" button
12. From dropdown, select "Detected by Silverfort"
13. Select either Server-01 OR Server-02 (but not both!) - This intentionally creates a policy violation for testing
14. Click on **Save**.



Note: This has created an allow policy for this service account, a notification will be generated for any violations.



Note: Notify is a 'passive' action. In production some customers only want notifications when there is a policy violation, so in addition to "Notify" may also use the "Alert to SIEM" to support part of a SOC workflow and Silverfort can be configured to send this alert using syslog.

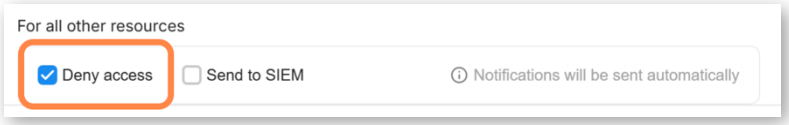
POLICY VALIDATION STEP #1

1. Return to your "**Exercise 5 - Service Account**" script
2. Choose "yes" to rerun the tests
3. **Expected Result:** All connections still succeed (we're only notifying at this stage)
4. Check **Authentication Logs**, filter for your service account
5. **Look for entries with Silverfort Action = "Notify"** - these are policy violations!

Silverfort Virtual Fencing - Enforcement Mode

Now let's activate real protection:

1. Go to **Service Account dashboard**
2. Use **Filters** → **Name** to find your service account
3. Click to expand the policy view
4. Change the "**Do**" action to **Block Access**
5. Click "**Save**"



Test Virtual Fencing - Enforcement Mode

5. Return to your script and rerun the tests
6. Expected Result: Some connections will now fail!
    - Connections to the allowed server: ✅ Success
    - Connections to the blocked server: ❌ Blocked

1. Check Authentication Logs for entries with Silverfort Action = "**Deny**"

What You've Accomplished:

- ✅ Discovered service account usage patterns in real-time
- ✅ Created virtual security boundaries around the service account
- ✅ Prevented unauthorized lateral movement
- ✅ Maintained legitimate service functionality

 ***Real-World Impact:*** *An attacker who compromises this service account can no longer use it to access unauthorized systems - their lateral movement is blocked.*

Exercise 6 - Risk-Based Policy

Estimated Time: 18 minutes

Why This Matters

Traditional security is binary - either you're authenticated or you're not. But real attacks happen gradually, with risk indicators building up over time. Risk-based authentication adapts security controls based on behavior patterns, applying stronger protection when activity looks suspicious.

Real Attack Scenario: An attacker compromises user credentials and accesses systems from unusual locations, at odd times, or in patterns that don't match the real user. Rather than blocking everything (friction) or allowing everything (unsafe), risk-based policies apply MFA only when the risk score elevates.

In this lab: You'll create dynamic policies that respond to risk levels in real-time. First, you'll generate baseline authentication activity and optionally build a policy directly from the authentication logs (demonstrating Silverfort's policy-from-logs capability). Then you'll configure a risk-based policy that triggers MFA only when user risk reaches "High or above." You'll test normal-risk scenarios (no MFA required), simulate suspicious activity to artificially elevate your risk score, and experience step-up authentication where the same action now requires MFA due to increased risk. This showcases adaptive security that's seamless for legitimate users but protective when threats are detected.

*Note: The first part of this exercise includes an optional element. If you are not doing this, please jump to the "**Dynamic Step-Up MFA**" section of this exercise and continue from there.*

Building Silverfort Policy Using Logs (optional deep dive)

This first section demonstrates Silverfort's powerful capability to create policies directly from authentication events you observe in the logs.

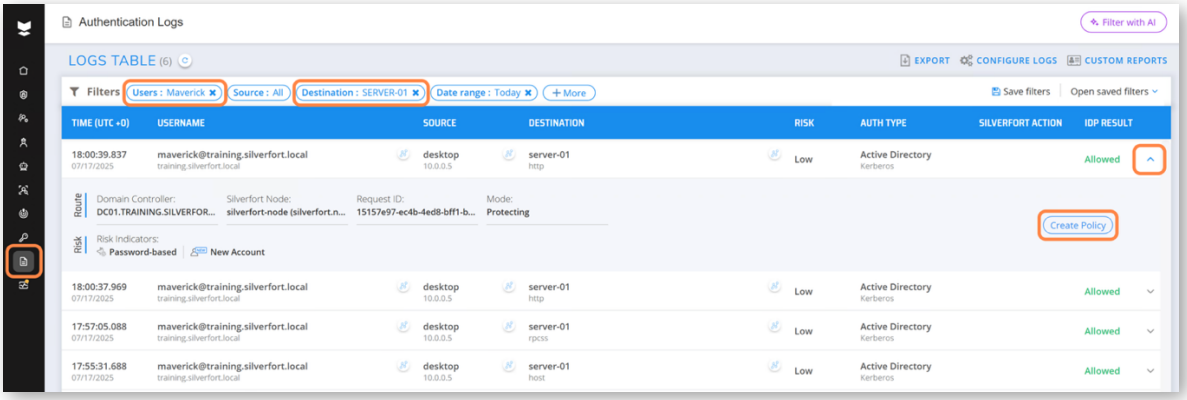
Generate Baseline Activity:

1. Open **Labs Folder** and run "**Exercise 6a - Risk**"
2. This opens an incognito Chrome browser to the "Test Flight" application on Server-01
3. Authenticate with your credentials (e.g., "training\maverick")
4. You should see the Test Flight application page
5. **Close the incognito window** before continuing

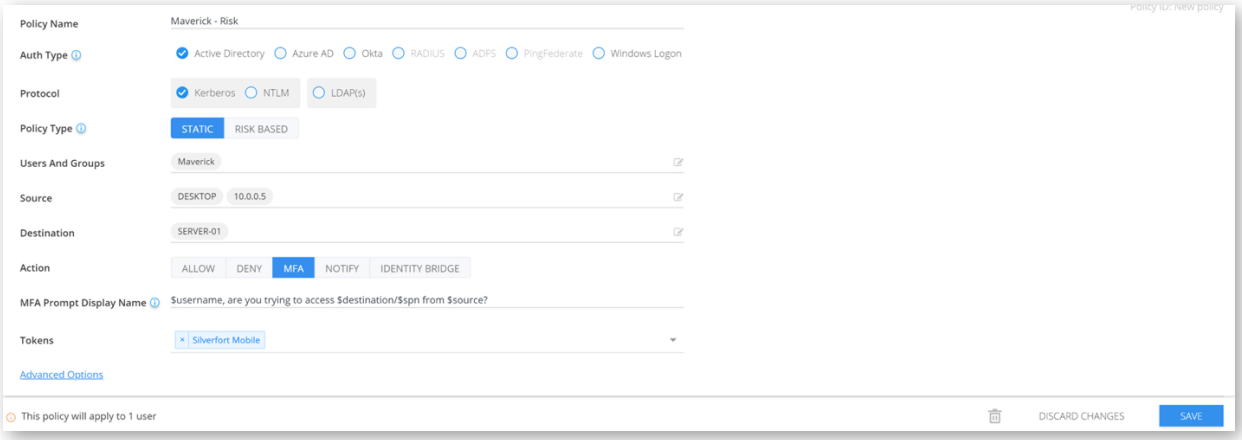
Create Policy from Logs:

6. Go to **Authentication Logs** in Silverfort dashboard

7. Create filters for:
    - **Users:** Your test flight user
    - **Destination:** Server-01
8. Find a relevant log entry and expand it
9. Look for the "**Create Policy**" option and click it



10. The "New Policy" window opens with conditions auto populated!
11. **Name the Policy:** YourUsername - Risk (e.g., "Maverick - Risk MFA")
12. Customize the MFA prompt message if desired
13. Click "**Save**" and "**OK**"



Testing Your New Policy

14. In your Lab folder, run “Exercise 6a - Risk” again.
15. This will open an incognito Chrome browser for the “Test Flight” application.
16. Authenticate using your Silverfort credentials, for example, “training\maverick”.
17. You should now be prompted for MFA.

18. The Test Flight page should be displayed.
19. Make sure to close the Incognito window before continuing.

Policy Validation Using Silverfort Logs

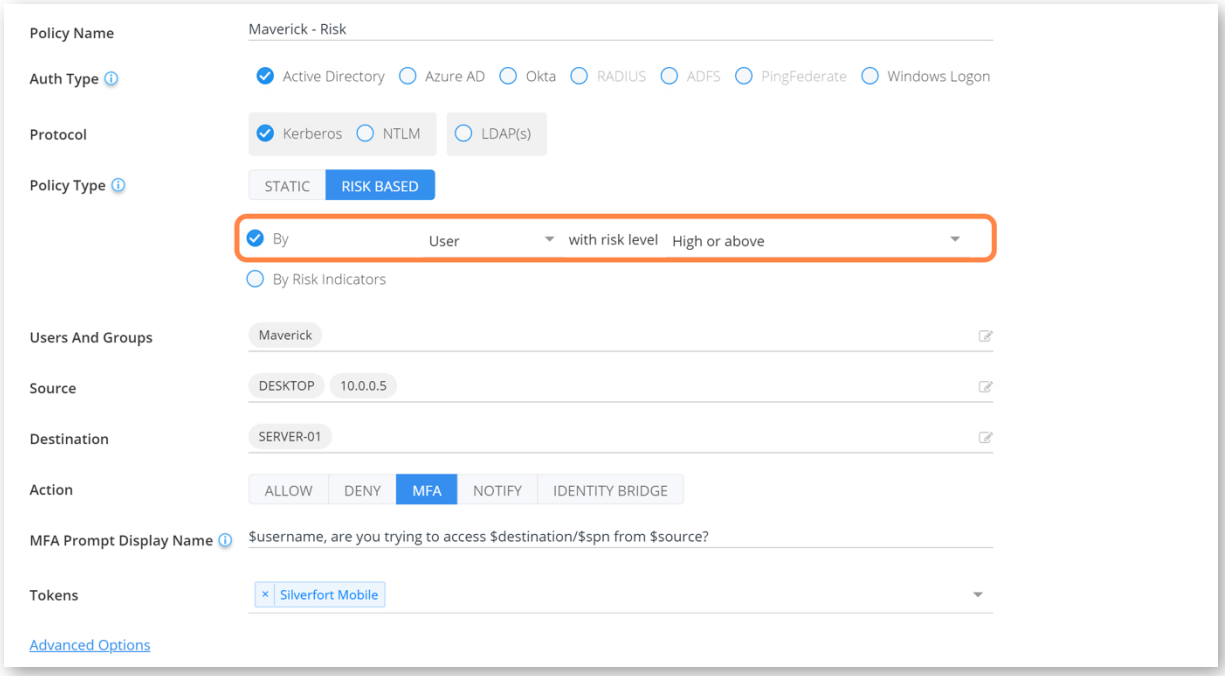
From the Silverfort Dashboard, go to the Authentication Logs view and create a filter using your Test Flight user and the destination “Server-01”. Note the Silverfort Action. Expand the log view to see more information, like the MFA token used and the user response time.

Dynamic Step-Up MFA

Silverfort evaluates risk in real time, assigning a risk level to each identity or entity based on authentication activity, threat signals, and Active Directory configuration. It can send alerts when suspicious behavior is detected, such as brute force attempts against service accounts or suspected *kerberoasting* of domain admins. More critically, Silverfort calculates the risk level of every authentication attempt as it happens.

These risk levels can be used as conditions in policy decisions for both users and service accounts. For example, if an employee logs in and the risk is high, the policy may require MFA. If the request comes from a third party with the same risk level, the policy could be configured to **DENY** access instead.

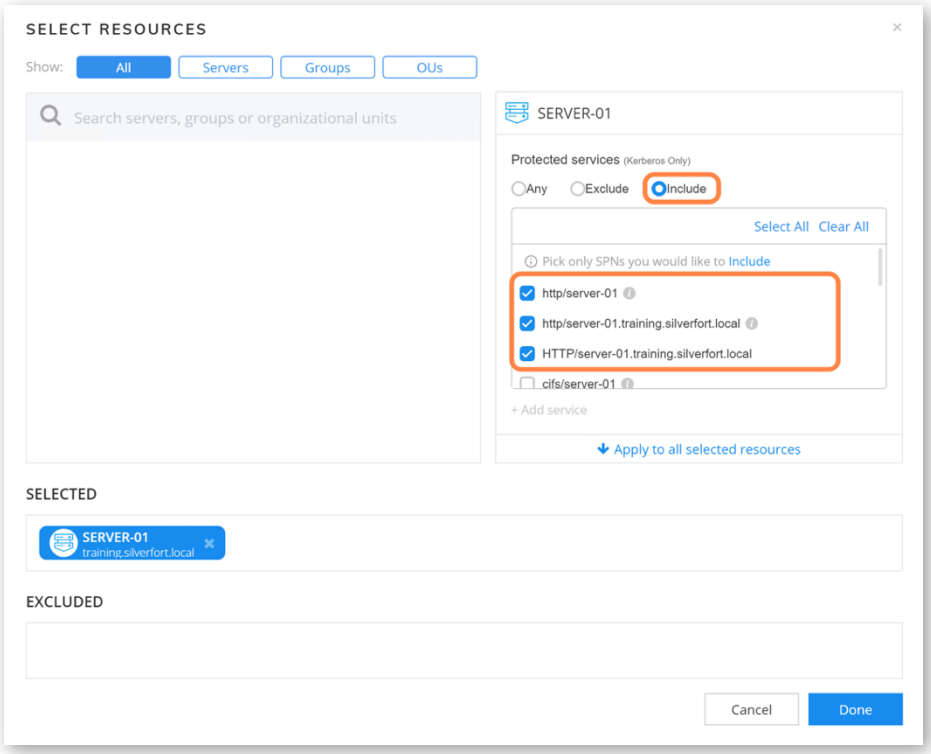
Create a Risk-Based Policy



Note: If you previously created this policy using the logs screen earlier, please edit per the following instructions.

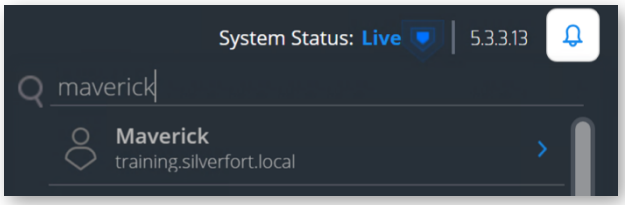
Create the Risk-Based Policy:

1. **Policy Name:** YourUsername - Risk Based Policy
2. **Auth Type:** Select Active Directory
3. **Protocol:** Select Kerberos
4. **Policy Type:** Select Risk Based
    - **Trigger:** By "**User**"
    - **Risk Level:** "**High or above**"
5. **Users And Groups:** Select your Silverfort username
6. **Source:** Leave as All Devices
7. **Destination:** Select Server-01
    - Choose "**Include**"
    - Click "**Clear All**"
    - Select all **http types**
    - Click "**Done**"
8. Click "**Save**" and "**Yes**" to activate

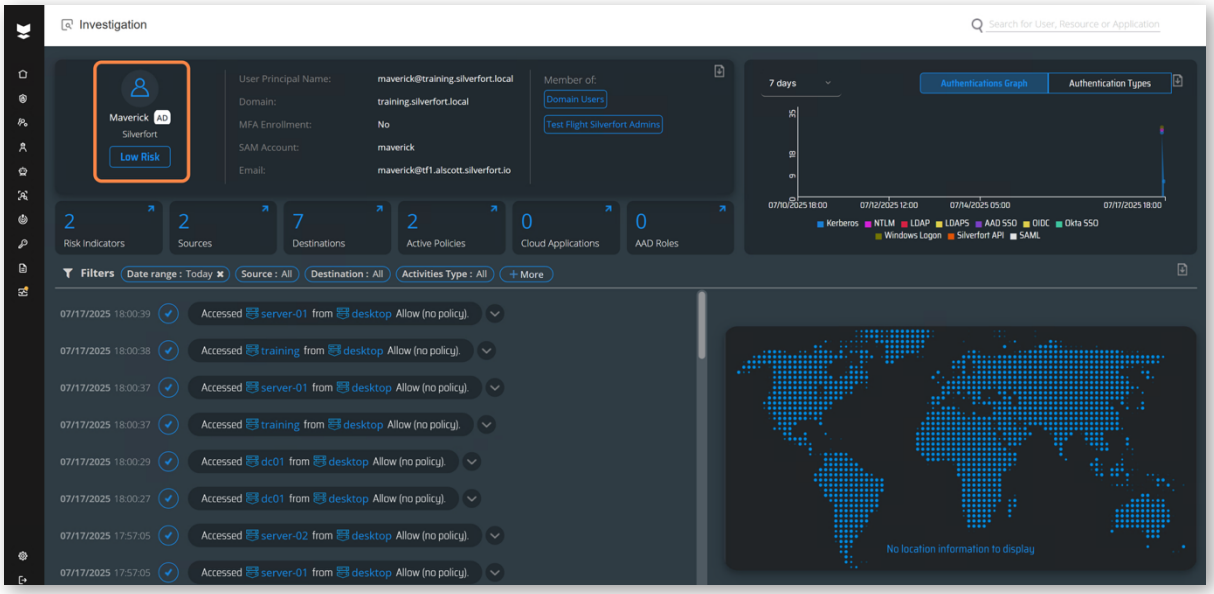


Testing the Risk-Based Policy

1. In the Silverfort dashboard, search for your username (top right search)
    - This opens your Investigation page



2. **Note your current risk score** (top left corner):



Test Normal Risk (Should not Trigger MFA):

3. Run "Exercise 6a - Risk" from the Labs folder
4. Authenticate with your credentials
5. Expected Result: No MFA required - your risk is still low
6. Close the browser window

Note: This is expected, as no policy matches were found. Use the Authentications Logs view or refresh the Investigation page for your Silverfort user; this provides a view of events for the user.

Simulate a Security Event:

7. Run "Exercise 6b - Risk" from Labs folder
8. This script simulates suspicious activity that raises your user risk score
9. Return to your Investigation page and refresh until you see the risk score increase - This may take 10-30 seconds to update

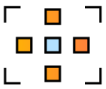
Test Elevated Risk (Should Trigger MFA):

10. Run "**Exercise 6a - Risk**" again
11. Authenticate with your credentials
12. **Now you should see an MFA prompt!**
    - The elevated risk triggered a step-up authentication request
13. Approve the MFA to access the Test Flight application

Policy Validation

Check your Authentication Logs to see:

- The first access (low risk): No MFA required
- The second access (high risk): MFA required
- The risk score that triggered the policy
- Your response time and MFA token used



Advanced Challenge: Try changing the policy action to "**Deny**" instead of MFA to see complete access blocking for high-risk scenarios.

What You've Accomplished:

- ✅ Experienced adaptive security that responds to risk
- ✅ Saw how legitimate users get seamless access when risk is low
- ✅ Experienced additional protection when risk indicators appear
- ✅ Built policies directly from observed authentication patterns

Exercise 7 - Privileged Access Security

Estimated Time: 25 minutes

Why This Matters

This is the crown jewel of identity protection. **Privileged account compromise is involved in 95% of successful breaches**, yet traditional PAM solutions are often clunky, expensive, and create gaps in coverage. Silverfort's Privileged Access Security provides comprehensive privileged access protection without the traditional PAM complexity.

Real Attack Scenario: An attacker gains access to a privileged account through credential theft, social engineering, or lateral movement. With traditional approaches, that account might have broad access across multiple tiers of infrastructure. The attacker can now access domain controllers, sensitive databases, and critical business systems.

Silverfort's Revolutionary Approach:

- **Automated Discovery:** Finds ALL privileged accounts, not just the obvious ones
- **Asset Tiering:** Enforces strict boundaries between infrastructure tiers
- **Virtual Fencing:** Limits where privileged accounts can be used
- **Just-in-Time (JIT):** Removes standing privileges entirely
- **Zero Trust:** Privileged access is earned, not granted

In this lab: You will go through three stages - Privileged Asset Classification, Virtual Fencing & Just-in-Time (JIT) removal, which effectively removes standing privileges as needed.

Part 1: Privileged Access Classification

Asset tiering is the foundation of privileged access security. We'll create a realistic tiered architecture:

- **Tier 0:** Domain Controllers, Identity Systems (Crown Jewels)
- **Tier 1:** Business Servers, Applications (Critical Infrastructure)
- **Tier 2:** User Workstations, Operational Systems (General Access)

This clear separation enables the enforcement of access restrictions, ensuring that privileged accounts are limited to their designated tiers. As a result, a compromise in a lower-tier account cannot easily compromise higher-value systems. It also helps organizations detect and prevent dangerous cross-tier authentications and build effective, risk-prioritized security controls.

Overall, asset tiering within Silverfort Privileged Access Security (PAS) is a foundational best practice for reducing the attack surface, improving visibility, and simplifying the adoption of privileged access security at scale.

DEFINING TIER 0 ASSETS

Now you will create two computer objects using a script. These computer objects will be prefixed with your Test Flight persona name and added to the “Tier 0 Computers” & “Tier 1 Computers” OUs in Active Directory. These computer objects can then be added to the Tier classifications configuration in Silverfort.

Create Your Test Infrastructure:

1. Open **Labs Folder** and run "**Exercise 7 - Create Computer Objects**"
2. This automatically creates test computer objects prefixed with your username
3. Return to Silverfort Management Console
4. Navigate to **Monitor** → **System Status**
5. Click "**Quick AD Sync**" to ingest the new computer objects
6. Wait about 1 minute for synchronization

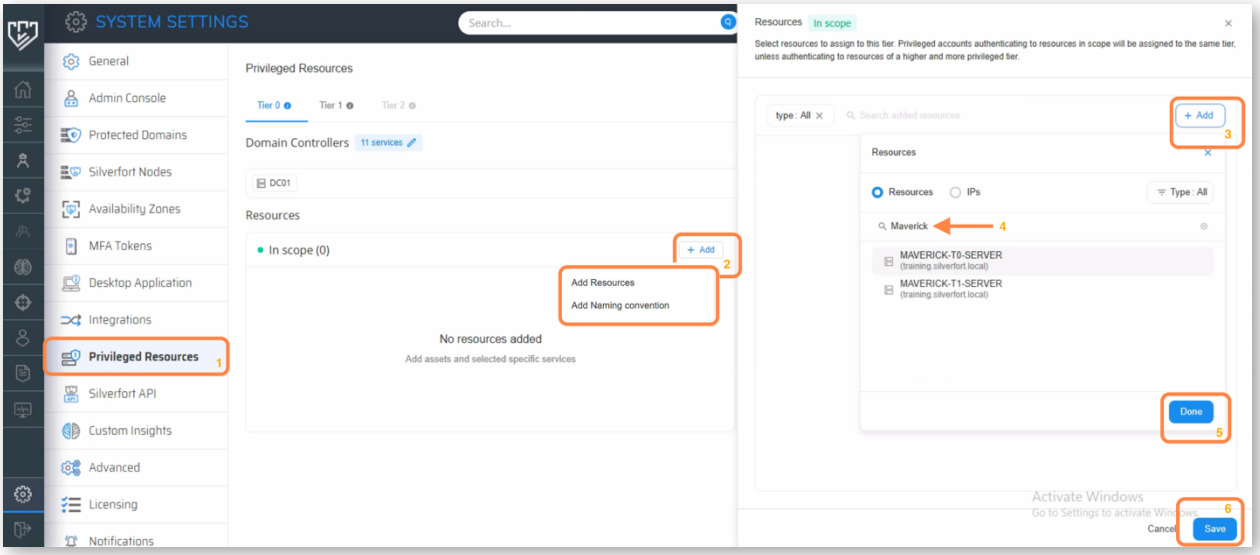
Verify Your Infrastructure:

7. *(Optional)* Use Exercise 4 shortcut to open *Active Directory Users and Computers console*
8. Navigate to: training.silverfort.com → Silverfort → Tier 0 Computers and Tier 1 Computers
9. Confirm your computer objects exist (e.g., "Maverick-T0-Server", "Maverick-T1-Server")

Configure Tier Classifications:

10. In Silverfort console, go to **Settings (gear icon)** → **Privileged Resources**

    Note: Domain Controllers are already classified as Tier 0 by default
11. In the "**In scope**" section, click "**+ Add** → **Add Resources**"
12. Click "**+ Add**" and search for your T0-Server (e.g., "Maverick-T0")
13. Select it and click "**Done**", then "**Save**"
14. **Repeat for Tier 1** using your T1-Server computer object



Review Service Principal Names (SPNs): When you add a computer object, notice the associated services (SPNs). These identify privileged activities based on authentication to these resources. You can customize which services are considered privileged for your environment. These are the Kerberos Service Principal Names (SPNs). *Note: SPNs can be unselected or new ones added to support the organization's tiering model and expected usage.*

Part 2: Privileged Access Security User Virtual Fencing

Silverfort's Privileged Access Security Virtual Fencing applies logical boundaries around sensitive assets by enforcing strict access controls based on asset classification and user privileges. This approach reduces exposure from compromised credentials, prevents unauthorized lateral movement across different resource tiers, and, should the worst happen, contains potential breaches within isolated zones. By monitoring cross-tier authentications and utilizing virtual fencing, this approach decreases the attack surface, enhances the overall security posture, and simplifies compliance efforts through the clear segregation of critical infrastructure.

Now let's create logical security boundaries around your privileged accounts:

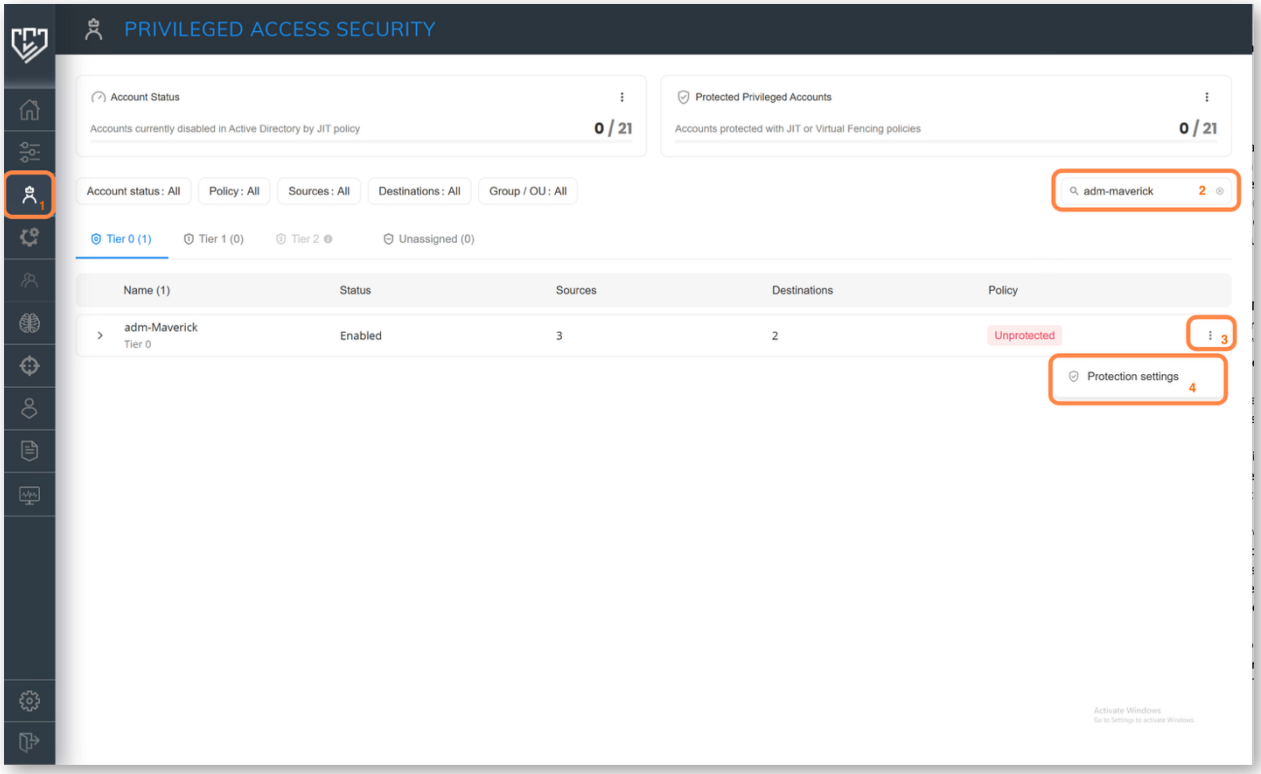
Generate a Privileged Account activity:

1. Open **Labs Folder** and run "**Exercise 7b - PAS User Access**"
2. Enter your administrator credentials when prompted (e.g., "adm-maverick")
3. From the script menu, try accessing TF-PAS-01 or TF-PAS-02 using:
    - **PowerShell option:** Returns server data when successful
    - **RDP option:** Starts remote desktop session (you'll be prompted for password)

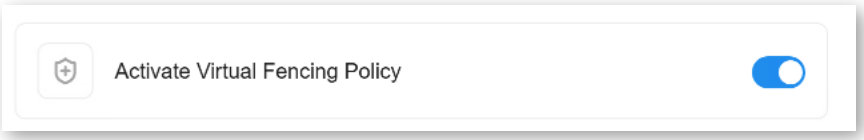
    Note: *No MFA is currently required (this is intentional)*
4. Feel free to test different options, then leave script running or close it

Configure Virtual Fencing:

5. Go to **Privileged Access** in the Silverfort menu
6. Search for your administrator account (e.g., "adm-maverick") in the top left
7. **Note:** Account shows as "**Unprotected**"
8. Click the three dots (⋮) on the far left, select "**Protection Settings**"

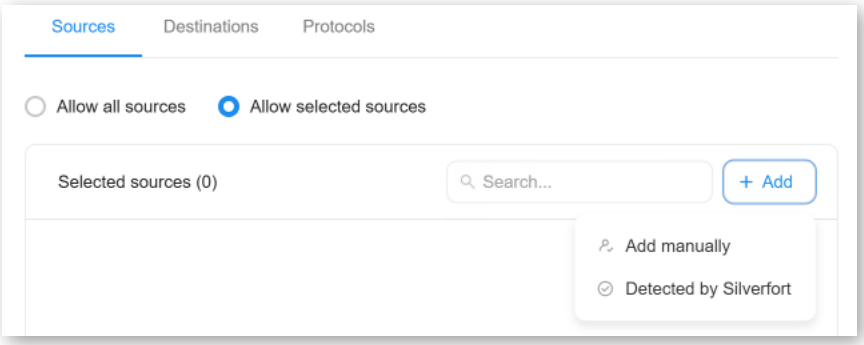


9. You'll see Virtual Fencing Policy is "**Inactive**"
10. **Move the slider to "Activate Virtual Fencing Policy"**



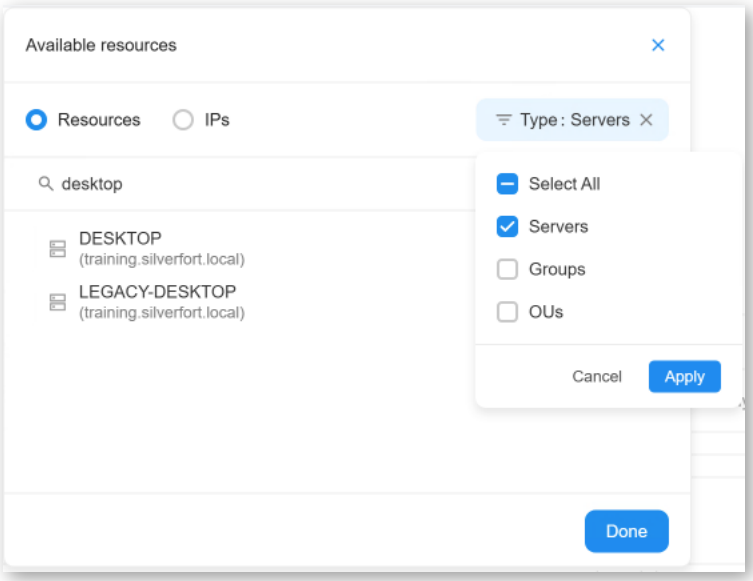
Configure allowed sources:

11. From the Sources view, select the radio button to only “allow selected sources”, and then click on the “+Add” button and use the option to “Add manually”

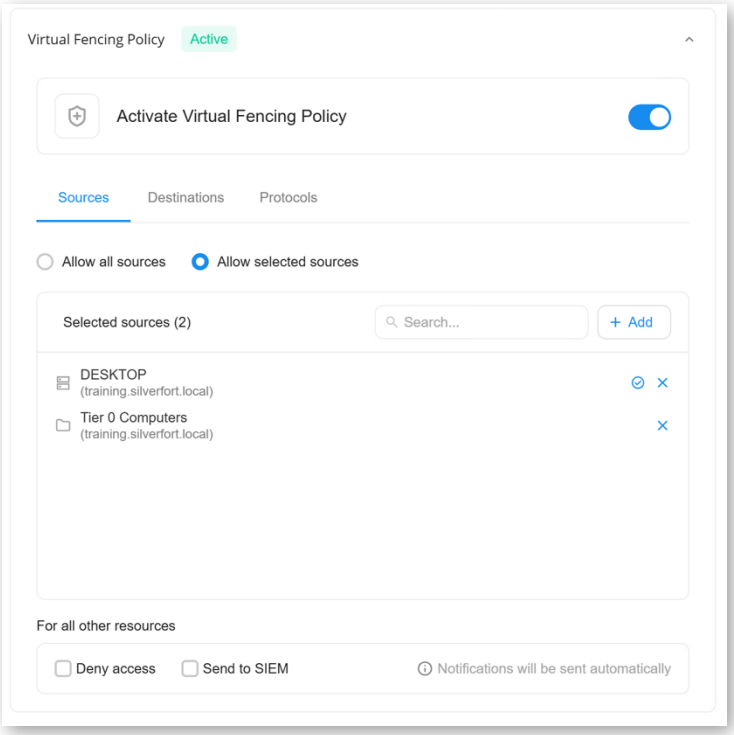


12. Click on Type, select Servers and then “Apply”

13. In the search type “Desktop”, ensure that only the DESKTOP computer object is added.

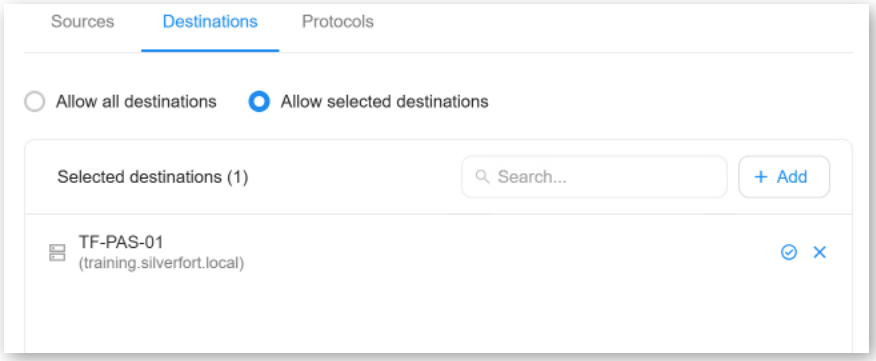


14. Click on Type, select OU and then “Apply”
15. In the search type “Tier 0” and add the “Tier 0 Computers” OU
16. Click on “Done”



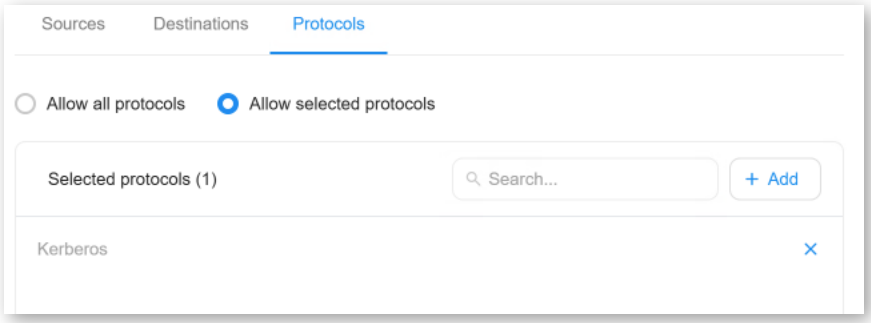
Configure Allowed Destinations:

17. Repeat for **Destinations**, adding only **ONE** of either TF-PAS-01 OR TF-PAS-02 - *This creates a policy violation for testing*



Optional: Protocol Restrictions:

18. *(Recommended for production)* Limit protocols to Kerberos only



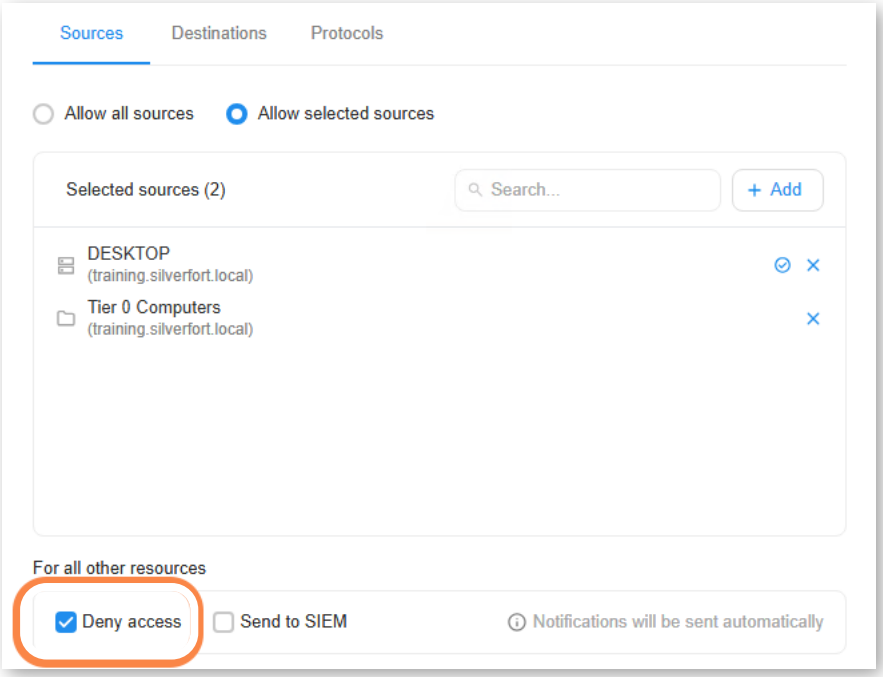
19. Click "**Save**"

    Note: *Account now shows "**Virtual Fencing**" protection status*

Test Virtual Fencing:

20. Use your script to test access to both TF-PAS-01 and TF-PAS-02
21. Expected Results: - Allowed server: ✅ Access works - Blocked server: ⚠️ Silverfort Action = "Notify" in logs
22. Check Authentication Logs to see the policy violations
23. Activate Enforcement: Change the Action to "Deny Access" and test again

24. Expected Results: Blocked server access should now fail completely

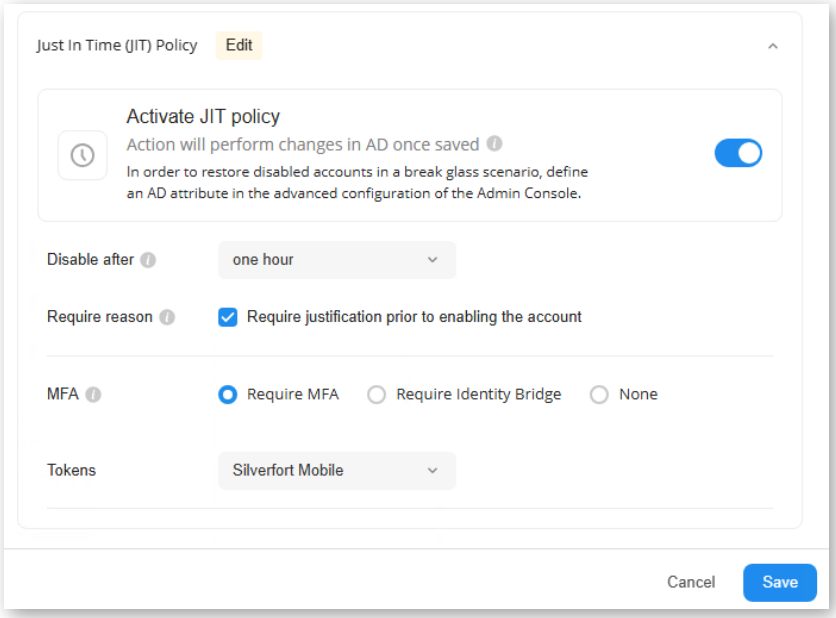


Part 3: Privileged Access Security Just-In-Time (JIT)

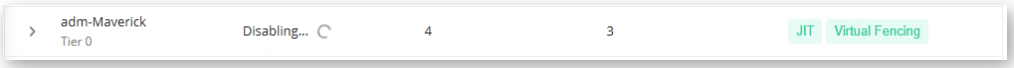
This is the ultimate privileged access protection - **zero standing privileges**. Privileged accounts are disabled by default and only activated when needed, for a limited time, with full MFA and justification requirements. Silverfort's PAS Just-In-Time (JIT) feature provides dynamic, time-limited activation of privileged accounts, enabling users to elevate access only when necessary and for a specified duration. This diminishes the risk linked with standing privileges and reduces the attack surface for critical accounts. The outcome is that privileged access remains tightly controlled, temporary, and, importantly, auditable.

1. From the Silverfort Management Console, select Privileged Access from the menu
2. Using the search field seen in the top left of the Privileged Access Security page, start typing the name of your administrator account, for example "adm-maverick"
3. Note, if you have completed the Virtual Fencing component, this account is marked as protected with "Virtual Fencing"; otherwise, the account is "Unprotected"
4. On the far left, click on the three vertical dots, and then select "Protection Settings"
5. The second half of the protection settings window is used to enable Just-in-Time (JIT) Policy and will be marked as "Inactive"
6. Move the slider to Activate a JIT policy

7. Leave the “Disable after” setting to one hour, although note that it is configurable - this is the time the account is automatically disabled again after it was enabled using JIT with Silverfort
8. Tick the box to “Require reason”; this provides an audit trail and is logged by Silverfort
9. MFA, leave the default radio button, “Require MFA”
10. Tokens, use the drop-down box to select Silverfort Mobile and then click “Apply”
11. Click on “Save”



12. Silverfort will now disable this account in Active Directory. Note the status for your administrator; it is protected with JIT and Virtual Fencing.

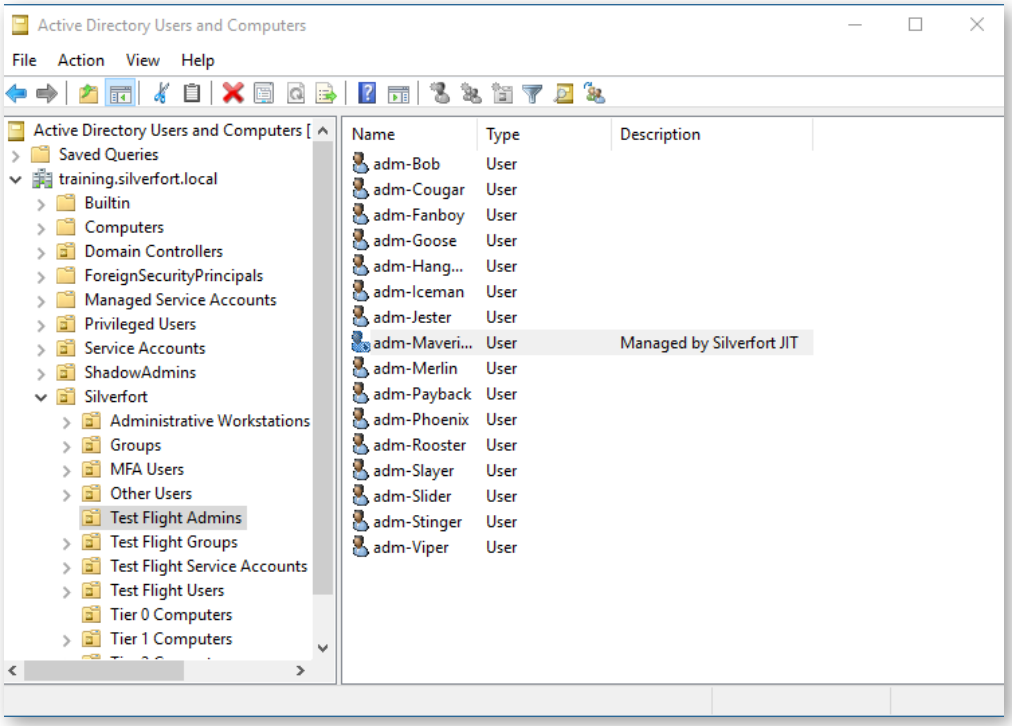


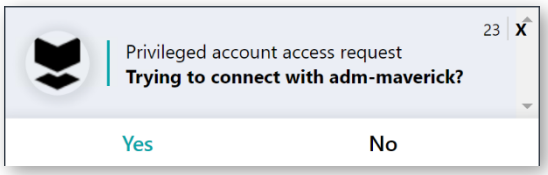
13. Rerun the **Exercise 4 shortcut** to open Active Directory Users and Computers console. In the training.silverfort.com > Silverfort > Test Flight Admins OU, see that your admin account is disabled and the description now reads “Managed by Silverfort JIT”

14. In your Lab folder, run “Exercise 7 – PAS User Access” if you’ve closed it and enter your administrator credentials, for example, “adm-maverick”
15. From the script menu, use either option to access the server that is permitted in your Virtual Fence policy

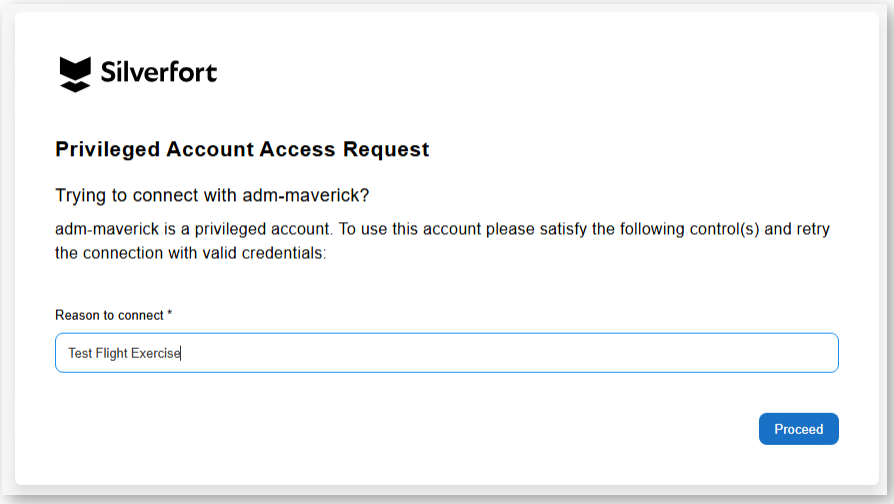
16. The session will fail as the account is disabled. As Silverfort PAS is protecting this account, the Silverfort Desktop Application prompts the user asking if it is "Trying to connect"

17. Respond Yes to start the workflow to enable the protected account

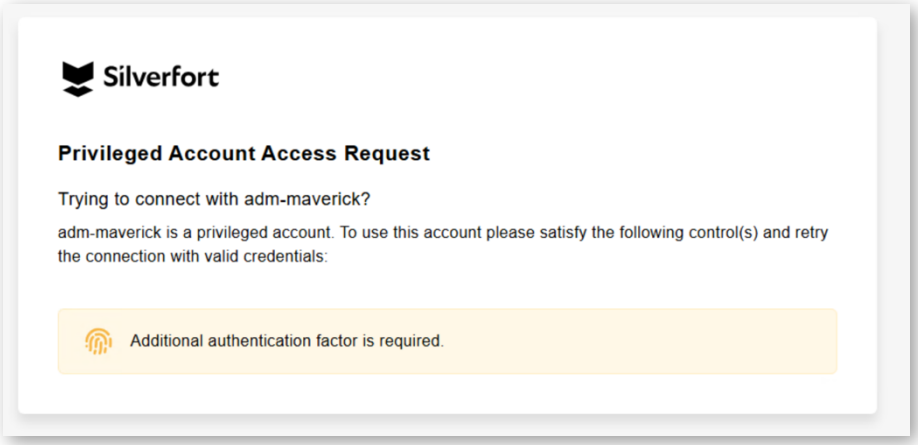




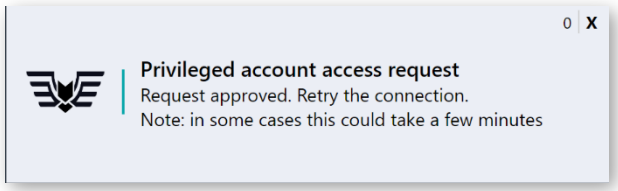
18. As the workflow requires justification, the default browser opens a page to provide this and logs it as part of the audit trail. Enter a reason and click on proceed.



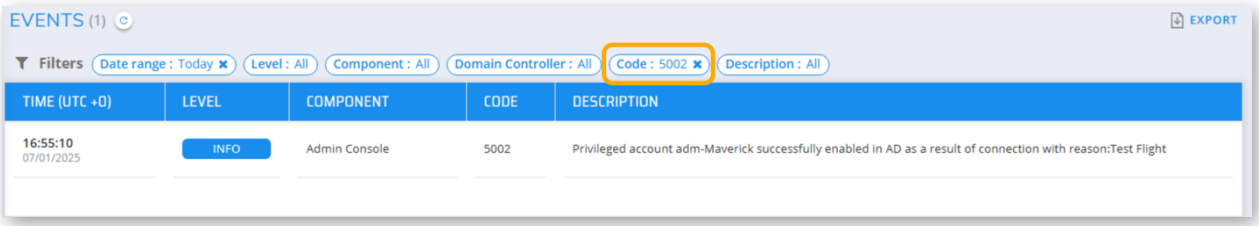
19. As MFA is also required as part of the workflow, before the account can be enabled, the browser session will be updated. Now check your phone for the MFA challenge and approve it.



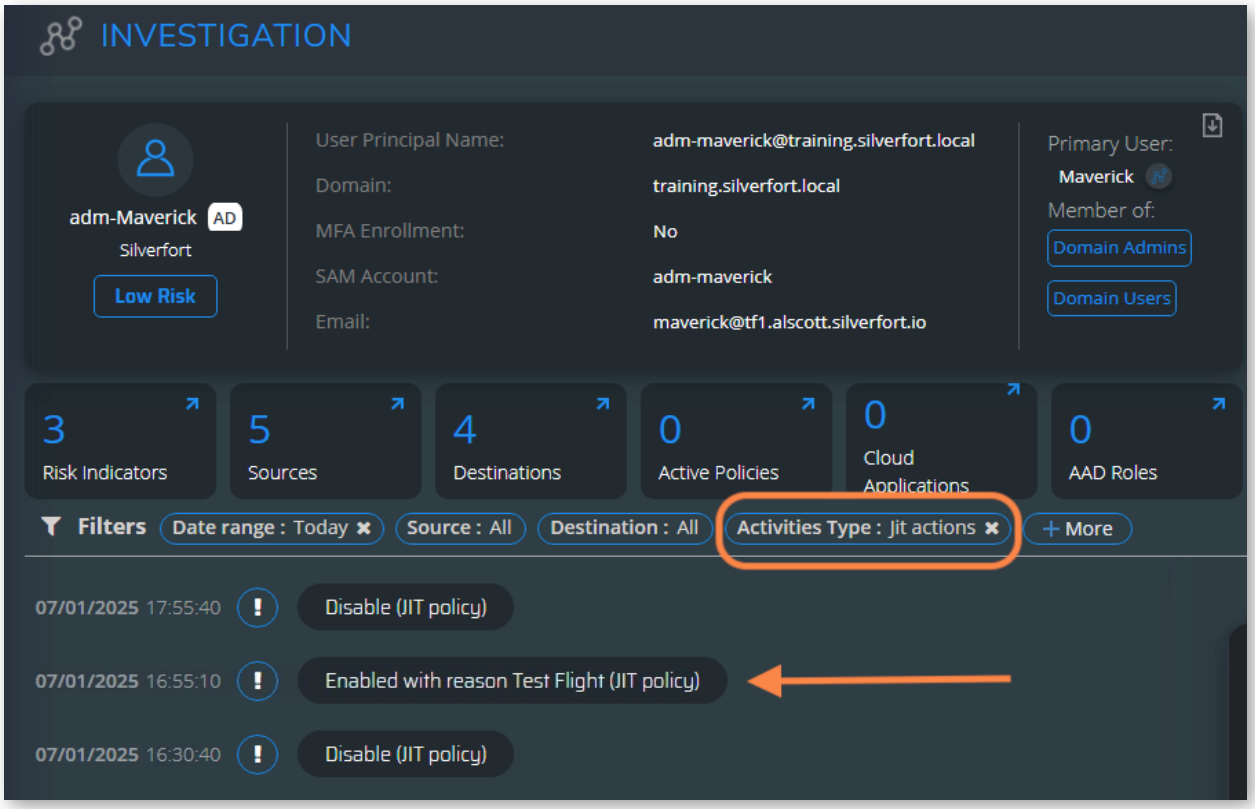
20. You will receive another desktop notification, "**Privileged account access request**", saying "**Request Approved, retry the connection."**



21. Return to the Silverfort Management Console > Privileged Access page. Search for your administrator user. Note that the account is enabled and a countdown is displayed, indicating the time remaining until Silverfort will automatically disable it, removing the standing privileges.
22. Optionally, check the user in Active Directory Users and Computers to verify that the account is enabled.
23. Using the script, try the access attempt again. Access this time works.
24. To verify the audit and the log entry, return to the Silverfort Management Console.
25. From the Menu on the left, click on 'Monitor > Notifications.
26. In the Events view, using the filter option across the top, click on "Code: All" and enter 5002, click on the + sign, or press return, to add this as a filter and then click "Apply"
27. If you wish to filter the results further, click on "Description: All" and add your administrator's username as a filter.
28. In the description, you can see the user's reason, along with the date and timestamp. This event can be sent to a SIEM or other log management tool for long-term retention and analysis.



29. This can also be viewed in the user investigation page.
30. From the Menu on the left, click on Dashboard (Home icon)
31. On the top right, use the search option by entering your administrator account name and then selecting it.
32. Use the Filters option, click on “Activities Type: All” and select “JIT actions”



Congratulations! You've Completed Advanced Privileged Access Security

What You've Accomplished:

- ✅ **Asset Classification:** Implemented tiered security architecture
- ✅ **Virtual Fencing:** Created logical security boundaries around privileged accounts
- ✅ **Just-In-Time Access (JIT):** Eliminated standing privileges entirely
- ✅ **Complete Audit Trail:** Full visibility into privileged access requests and usage
- ✅ **Real-World Protection:** Defended against 95% of successful breach techniques

Real-World Impact:

- **Before:** Privileged accounts had standing access that could be abused if compromised
- **After:** Privileged accounts are disabled by default, require justification and MFA to activate, automatically disable after use, and are restricted to specific systems

This is next-generation privileged access security!

Lab summary - Key Takeaways

You've just experienced the future of identity security. Here's what you've learned to defend against:

Exercise	Attack Prevented	Real-World Impact
RDP MFA	Initial access via stolen credentials	Stops 70% of ransomware entry points
PowerShell MFA	Lateral movement and remote execution	Breaks the #1 technique for network traversal
Console MFA	Management interface compromise	Protects your security tools from attack
Privilege Escalation	Admin credential abuse	Stops privilege elevation attacks
Service Account Fencing	Machine-to-machine lateral movement	Prevents 80% of lateral movement techniques
Risk-Based Policy	Insider threats and compromised accounts	Adaptive security based on behavior
Identity Segmentation	Complete privileged access protection	Eliminates 95% of successful breach paths

The Bottom Line: You've built a comprehensive defence that adapts to threats in real-time, protects both human and machine identities, and provides the visibility and control needed for modern cybersecurity.

Next Steps:

- Consider how these techniques apply to your environment
- Think about your current identity security gaps
- Explore how Silverfort could enhance your existing security stack

Thank you for completing the Silverfort Test Flight Lab Guide!
Your journey into next-generation identity security starts here.