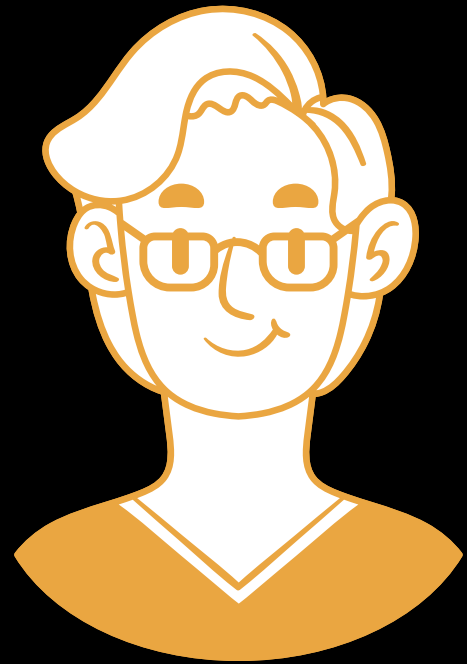
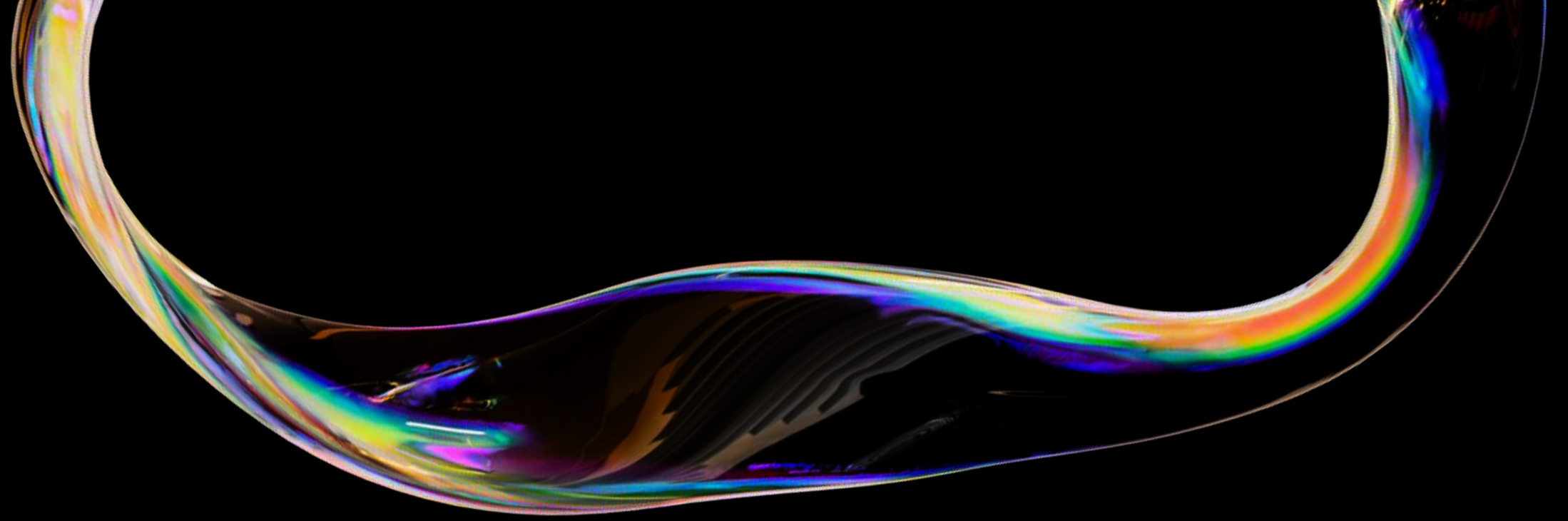


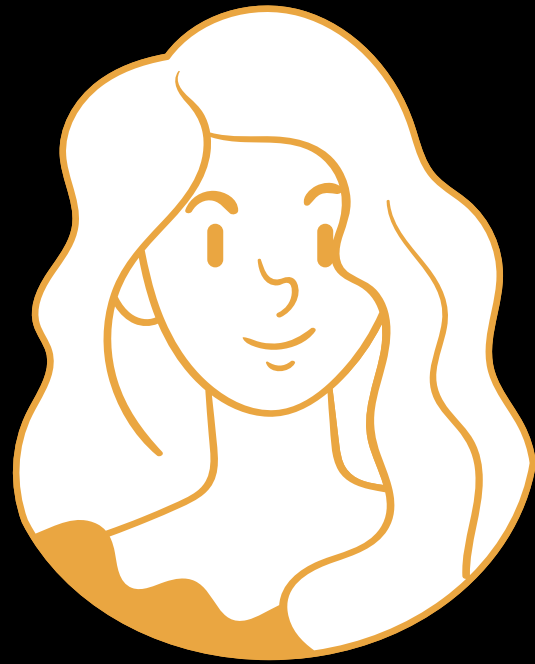
# RED TEAM CAPSTONE CHALLENGE



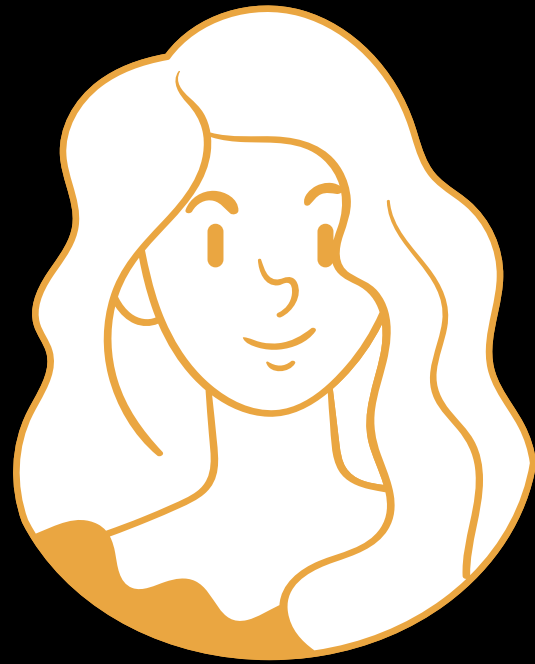
# MEET THE TEAM



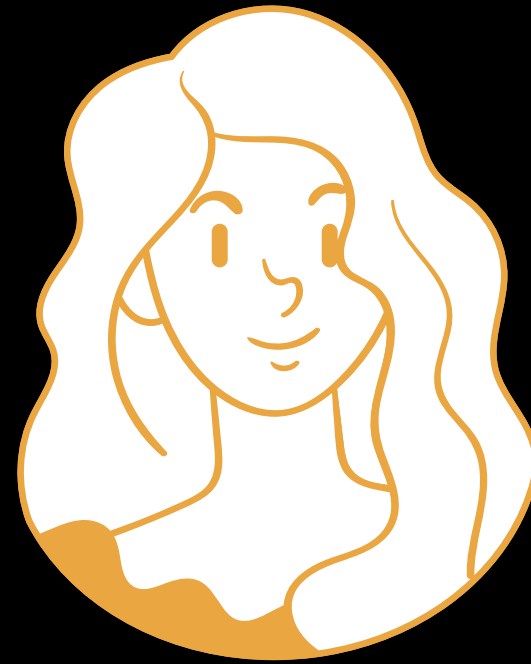
MUHAMAD  
SHOKRY



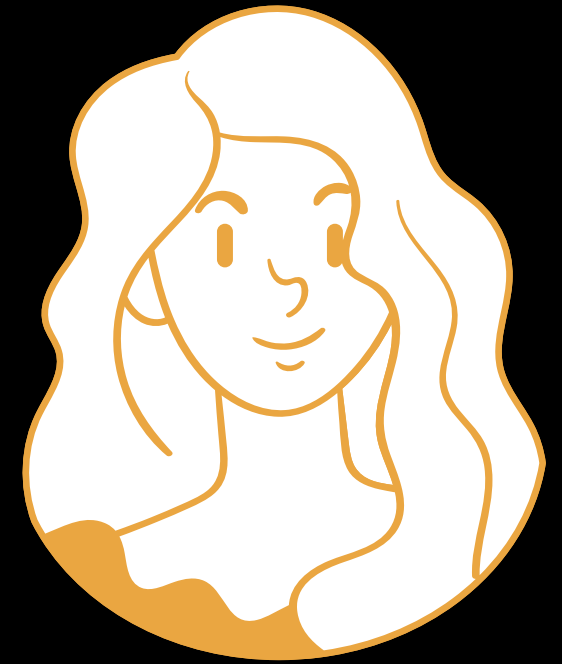
HALA  
DAIHOOM



AYA  
REDA



OLA  
GABER



ROWAYDA  
MOHSEN

# TODAY'S AGENDA

- Introduction
- Initial Access
- AV Evasion
- Lateral Movement
- Privilege Escalation
- Post-Compromise Exploitation





# INTRODUCTION

We, as the Red Team, have been tasked by the government of Trimento – a small but wealthy island nation in the Pacific due to foreign investments – to perform a comprehensive security assessment of the Reserve Bank, known as TheReserve. The bank has two key divisions: one serving corporate clients and foreign investors, and the other handling financial transfers between global banks. The government is concerned that the corporate division may pose a threat to the sensitive financial transfer operations due to insufficient segregation between the two divisions. Our assessment will cover both internal and external networks and will determine whether it is better to separate the two divisions into distinct entities.



# Red Team Capstone The Reserve Bank

This covers a red team engagement for Trimento's Reserve Bank.





# Project Overview & Goal

## Scope

Assess corporate and bank divisions for security risks.

## Goal

Simulate fraudulent transfer via SWIFT backend access.

## SWIFT System

Isolated backend with internal web app for transfers.

# Transfer Process & Security

## 1 Step 1

Customer requests transfer and receives a code.

## 2 Step 2

Employee with capturer role authenticates and captures transfer.

## 3 Step 3

Approver reviews and approves transfer from jump host.

## 4 Separation of Duties

No single employee can capture and approve the same transfer.

# Secure & Banking



Worldwide sole badge un  
nious cleat bankrige



arroate the preflice un  
nious band branking



asconaline aduce un  
nious cleat bankrige





# Project Scope & Rules

## In-Scope

- Internal and external network testing
- OSINT on corporate website
- Phishing and mailbox attacks
- Simulated fraudulent transfer

## Out-of-Scope

- External OSINT
- Mail server config changes
- Attacks outside subnet
- VPN and e-Citizen platform testing



# Tools & Registration

1

## Tools Provided

Password policies and common tool lists included.

2

## Registration

Register via e-Citizen SSH portal using provided credentials.

3

## Proof of Compromise

Perform specific steps on compromised hosts to prove access.





# Phase 1: OSINT – Gathering Public Intelligence

## Objective

Collect public intelligence and user credentials using OSINT techniques.

## Information Collected

- Domain: thereserve.loc
- NetBIOS Name: THERESERVE
- Valid Credentials for multiple users

## Tools & Techniques

Used theHarvester, Hunter.io, and custom Google dorking to gather data.

Password policy observed:  
minimum 8 characters, including numbers and special characters.



## 🔍 Phase 1: OSINT (Simulated)

### Goal

Gather public intelligence and user credentials using open-source intelligence techniques.

### 🔍 Collected Information:

- Domain Name: thereserve.loc
- NetBIOS Name: THERESERVE
- Valid Credentials Obtained:
  - laura.wood@corp.thereserve.loc : Password1@
  - mohammad.ahmed@corp.thereserve.loc : Password1!
  - muhamadsabek@corp.th3reserve.loc : jjaGRX\_YnANjztr3





## Observed Password Policy:

Minimum of 8 characters, must contain at least one number and one special character (!@#\$%^).

## Result:

OSINT phase completed successfully, revealing valid user credentials, the internal domain structure, and authentication format.



# Phase 2: Enumeration & Fuzzing


## Goal

Identify internal hosts, open ports, and running services to map the attack surface.

## Tools Used

- Nmap for port scanning
- ffuf for HTTP fuzzing
- Manual enumeration techniques

## Findings

- Multiple hosts with SSH, HTTP, SMTP, MySQL, and RDP services
  - Internal mail server and domain controller identified
- 



# Initial Network Exploration



## VPN Server

SSH and HTTP services open.



## WebMail Server

Multiple mail protocols and MySQL open.



## Web Server

SSH and HTTP services active, hosting company site.



# Exploring The Network

## Nmap

### VPN

```
root@ip-10-10-46-88:~# nmap 10.200.116.12 -sV -sC
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-12 05:20 BST
Nmap scan report for 10.200.116.12
Host is up (0.0022s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 17:f3:c2:89:2a:eb:25:90:02:f9:e0:c1:a8:6f:b3:3c (RSA)
|   256 53:8c:34:1c:e2:5d:2d:2f:69:df:b9:4f:1d:13:fa:18 (ECDSA)
|_  256 02:3f:29:8d:a6:58:51:0e:c9:ee:5f:f3:1a:04:92:24 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: VPN Request Portal
Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.59 seconds
[11] Done thunderbird
```

## Result:

### VPN 10.200.116.12

22/tcp open ssh    OpenSSH 7.6p1 Ubuntu 4ubuntu0.5

80/tcp open http    Apache httpd 2.4.29 ((Ubuntu))

## WebMail

```
root@ip-10-10-46-88:~# nmap 10.200.116.11 -sV -sC
```

## WebMail 10.200.116.11

22 ssh

25 stmp

110

135

139

143

587

3306/tcp open mysql

3389/tcp open ms-wbt-server Microsoft Terminal Services



# WEB machine.

```
root@ip-10-10-46-88:~# nmap 10.200.116.13 -sV -sC
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-12 05:45 BST
Nmap scan report for 10.200.116.13
Host is up (0.0072s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 ac:ae:01:d7:8e:da:bf:5c:ff:b5:69:93:79:94:2b:52 (RSA)
|   256  81:5a:9e:79:a5:70:00:cf:8d:d0:8a:18:6a:37:67:91 (ECDSA)
|_  256  53:4d:82:5f:b3:f5:ee:d6:e5:35:d8:f6:b4:cf:24:99 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit
Nmap done: 1 IP address (1 host up) scanned in 7.96 seconds
```

## Result:

Web 10.200.116.13

22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.7

80/tcp open http Apache httpd 2.4.29 ((Ubuntu))

Target\_Name: THERESERVE

| NetBIOS\_Domain\_Name: THERESERVE

| NetBIOS\_Computer\_Name: MAIL

| DNS\_Domain\_Name: thereserve.loc

| DNS\_Computer\_Name: MAIL.thereserve.loc



## Tools Used:

Nmap, Manual Enumeration Techniques

## Discovered Hosts & Services:

IP Address	Host	Services Detected
10.200.116.11	WebMail	SSH (22), SMTP (25), MySQL (3306), RDP (3389), etc.
10.200.116.12	VPN Server	SSH (22), HTTP (80) - Apache 2.4.29
10.200.116.13	Web Server	SSH (22), HTTP (80) - Apache 2.4.29

## Additional Discovery:

- Internal Mail Server: MAIL.thereserve.loc
- Domain Controller: THERESERVE.LOC

## Result:

Enumeration was successful. Multiple internal services and hosts were identified, providing multiple entry points for exploitation.





# Phase 3: Phishing & Initial Foothold



## Phishing Campaign

Sent a phishing email with a link to a vulnerable page on the VPN server.



## Exploitation

Remote Code Execution via unsanitized input in requestvpn.php on VPN server.



## Result

Reverse shell triggered successfully, granting internal network access.



# Post-Exploitation: SSH Access & Email Extraction

## SSH Access

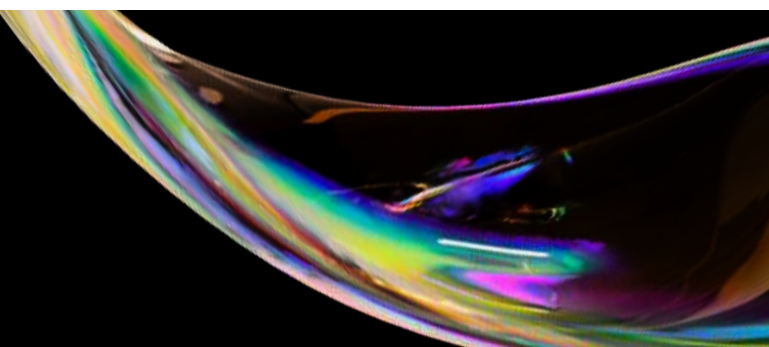
Connected to internal system via SSH using compromised credentials.

Enabled further reconnaissance and data gathering.

## Email Access

Used Thunderbird over SSH tunneling to access internal emails.

Extracted sensitive data including passwords and internal communications.





# Internal Role & Domain Mapping

Name	Email	Role
Paula Bailey	paula.bailey@corp.thereserve.loc	CEO
Christopher Smith	christopher.smith@corp.thereserve.loc	CIO
Charlene Thomas	charlene.thomas@corp.thereserve.loc	CMO
Mohammad Ahmed	mohammad.ahmed@corp.thereserve.loc	Developer (compromised)

This mapping provided insight into key personnel and their access levels within the organization.







# Summary of Red Team Engagement

## Phases Executed

- OSINT and credential harvesting
- Enumeration and service discovery
- Phishing attack and exploitation
- Post-exploitation lateral movement and data extraction

## Outcome

Successful initial access and internal network control demonstrated.

Valuable insights gained on security weaknesses and attack vectors.



# Key Tactics Demonstrated



OSINT & Credential  
Harvesting



Port Scanning &  
Service Discovery



HTTP Fuzzing



Phishing & Payload  
Delivery



Reverse Shell  
Exploitation



SSH Internal Access



Email Extraction



Internal Role &  
Domain Mapping



# AV EVASION



Objective : Gain reverse shell access to the target server while bypassing antivirus or endpoint detection systems.

## Approach Chosen:

Performed reconnaissance to identify active hosts and services.

Used brute force via Hydra on a login portal to obtain credentials.

Generated a custom reverse shell payload with AV evasion techniques.

Established persistent access using SSH key injection.

Accessed and moved laterally to VPN (10.200.118.12) and WRK1 (10.200.118.21).



# PRIVILEGE ESCALATION





# ENGAGEMENT OVERVIEW

This red team exercise aimed to simulate a real-world attacker escalating from limited internal access to full control over an Active Directory domain. The objective was to test the environment's resilience against stealthy privilege escalation, credential dumping, and persistence techniques. I executed all actions in this segment, focusing on evasion, stealth, and post-exploitation dominance.

# ESTABLISHING INITIAL FOOTHOLD VIA CHISEL

To establish internal access, I used the Chisel tunneling tool to create a reverse SOCKS proxy. The Chisel server was launched on an external machine using the `--reverse` flag, listening on port 8000. On the compromised internal host, the Chisel client initiated the reverse tunnel, effectively bypassing egress restrictions.

This encrypted tunnel enabled secure command and control while avoiding detection by network defenses. It also laid the foundation for stealthy lateral movement and enumeration.



# INTERNAL RECONNAISSANCE USING KERBEROASTING

After achieving initial access, I conducted SPN enumeration using GetUserSPNs.py from Impacket, routed through proxychains to maintain OPSEC.

Valid domain credentials were used to request service tickets for SPNs on the Domain Controller. These tickets were stored offline to later perform password cracking — a technique known as Kerberoasting.

The goal was to extract weak service account credentials and use them to escalate privileges.





# PRIVILEGE ESCALATION VIA CRACKED SERVICE ACCOUNT

The extracted service tickets were cracked offline, revealing the password for a service account named (svcScanning).

To validate access, I used xfreerdp with proxychains to open a Remote Desktop session to the internal host. This confirmed that the credentials were valid and gave me interactive GUI access for further enumeration and post-exploitation.

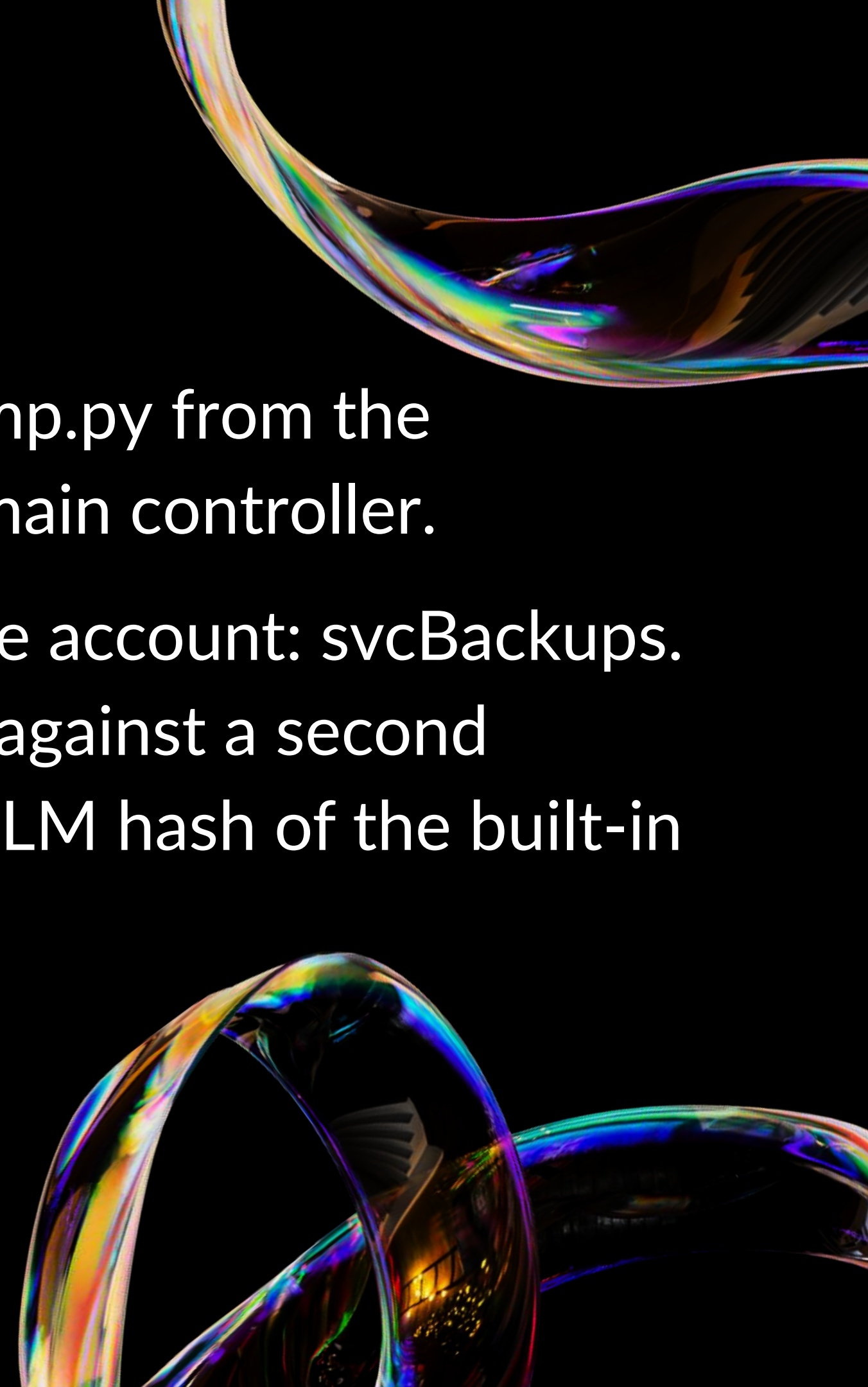


# CREDENTIAL DUMPING WITH SECRETSDUMP.PY

With service-level access, I leveraged secretsdump.py from the Impacket suite to dump credentials from the domain controller.

This operation exposed another privileged service account: svcBackups. Using it, I performed an additional secrets dump against a second domain controller, successfully extracting the NTLM hash of the built-in Administrator account (RID 500).

This hash became the key to accessing any system using Pass-the-Hash techniques.





# PASS-THE-HASH TO GAIN ADMINISTRATOR ACCESS

I executed a Pass-the-Hash attack using Evil-WinRM, authenticating to the domain controller using the Administrator's NTLM hash.

I executed a Pass-the-Hash attack using Evil-WinRM, authenticating to the domain controller using the Administrator's NTLM hash.





# GUI-BASED CONTROL WITH RDP

To improve usability, I switched to a graphical interface using xfreerdp3. Logging in as Administrator with a known password (Muhamad999), I gained full GUI access to the domain controller.

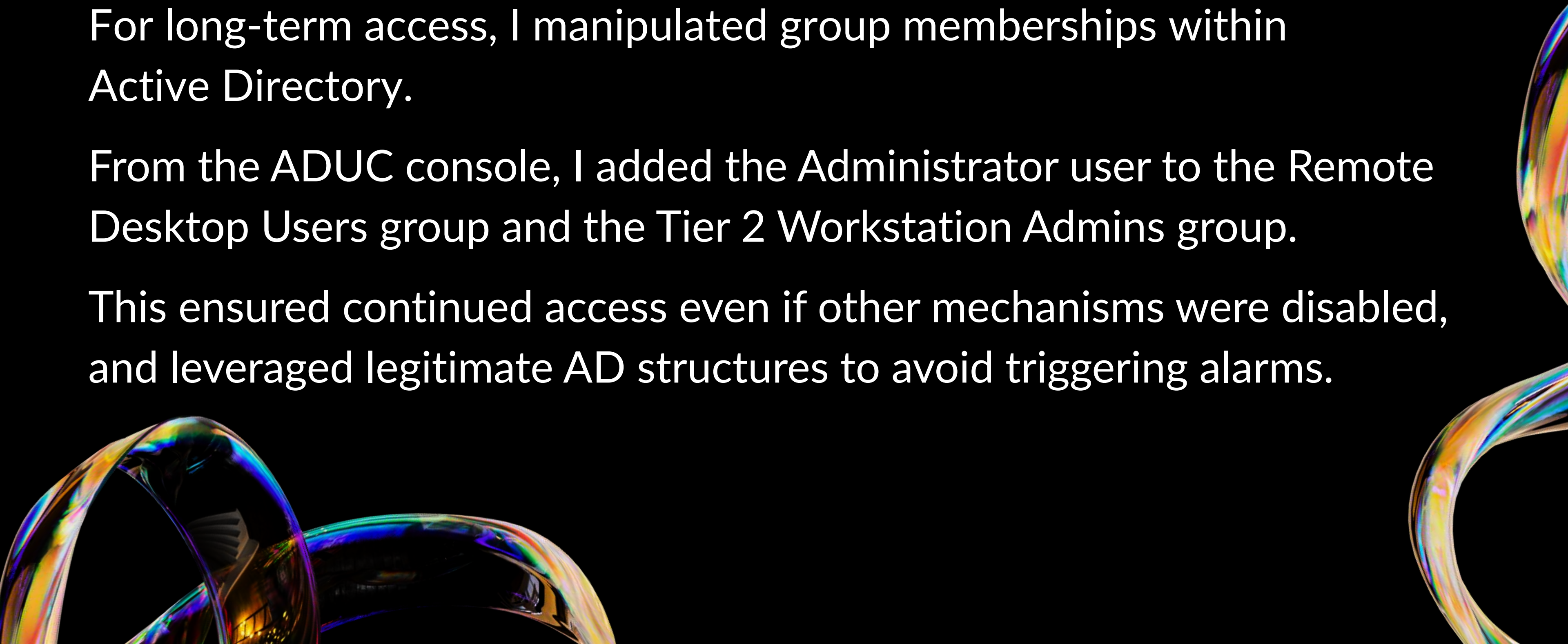
From this interface, I could launch Active Directory tools, browse files, and deploy post-exploitation payloads with ease, all through the encrypted SOCKS proxy.

# PERSISTENCE THROUGH GROUP MEMBERSHIP MANIPULATION

For long-term access, I manipulated group memberships within Active Directory.

From the ADUC console, I added the Administrator user to the Remote Desktop Users group and the Tier 2 Workstation Admins group.

This ensured continued access even if other mechanisms were disabled, and leveraged legitimate AD structures to avoid triggering alarms.





# DEPLOYING POST-EXPLOITATION TOOLS

To perform advanced Kerberos attacks, I securely transferred Mimikatz and Rubeus to the compromised machine.

Using SMB or RDP clipboard, the tools were moved discreetly and executed in-memory where possible. This allowed me to extract Kerberos tickets and prepare for Golden Ticket operations while evading endpoint detection.

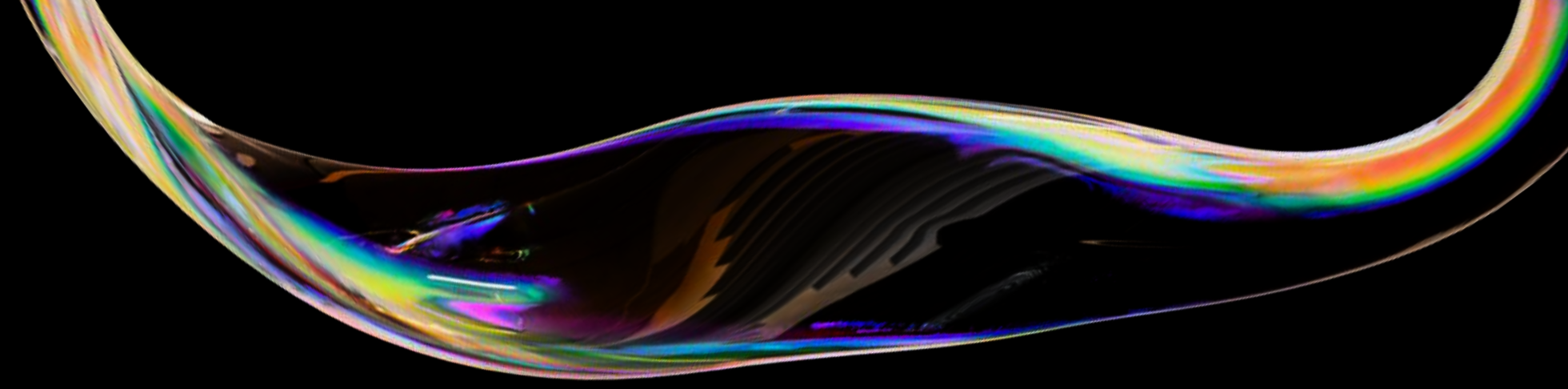


# ACTIVE DIRECTORY RECONNAISSANCE

Before launching Golden Ticket attacks, I performed AD recon using PowerShell.

Commands like `Get-ADComputer` and `Get-ADGroup` were used to identify high-value targets, such as domain controllers and Enterprise Admins.

Understanding the AD hierarchy was critical for planning targeted movements and ensuring successful impersonation.

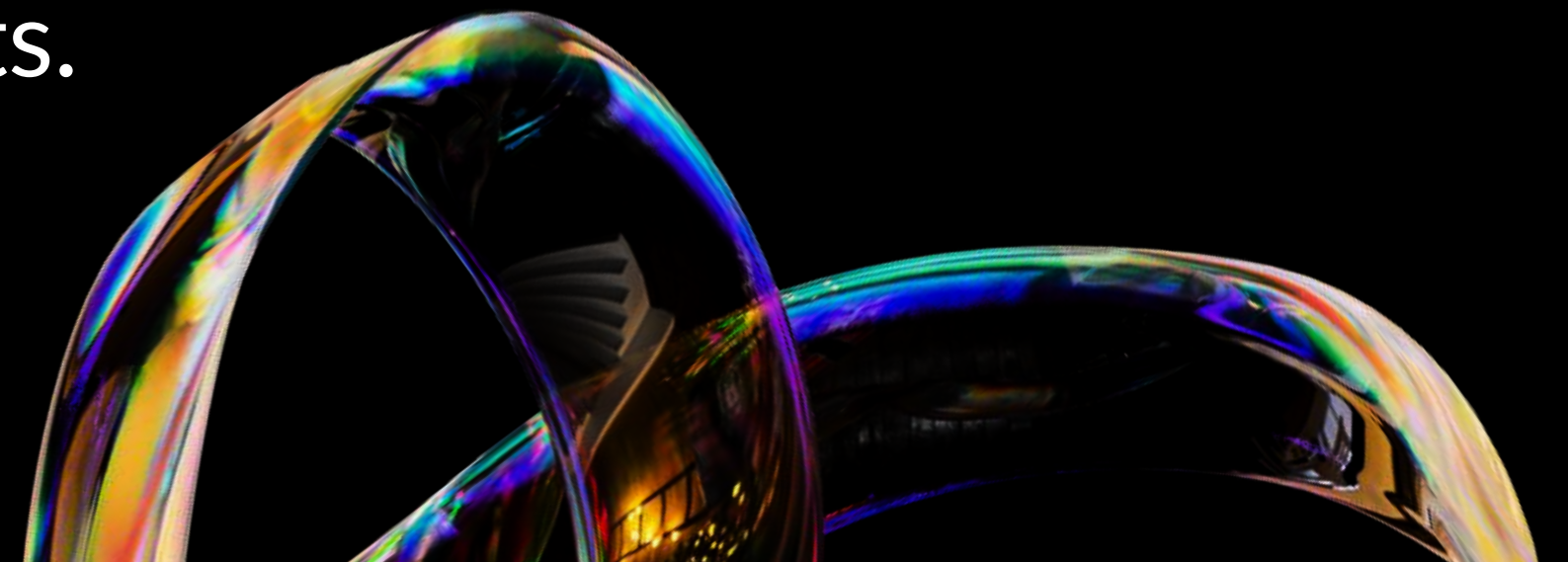


# GOLDEN TICKET ATTACK EXECUTION

Using lsadump::dcsync in Mimikatz, I retrieved the krbtgt NTLM hash — the most critical credential in the domain.

With this hash, I forged a valid Kerberos Ticket Granting Ticket (TGT) impersonating the Administrator.

The forged TGT was injected into memory using Mimikatz and Rubeus, granting me full, undetectable access to any Kerberos-protected resource without authentication prompts.

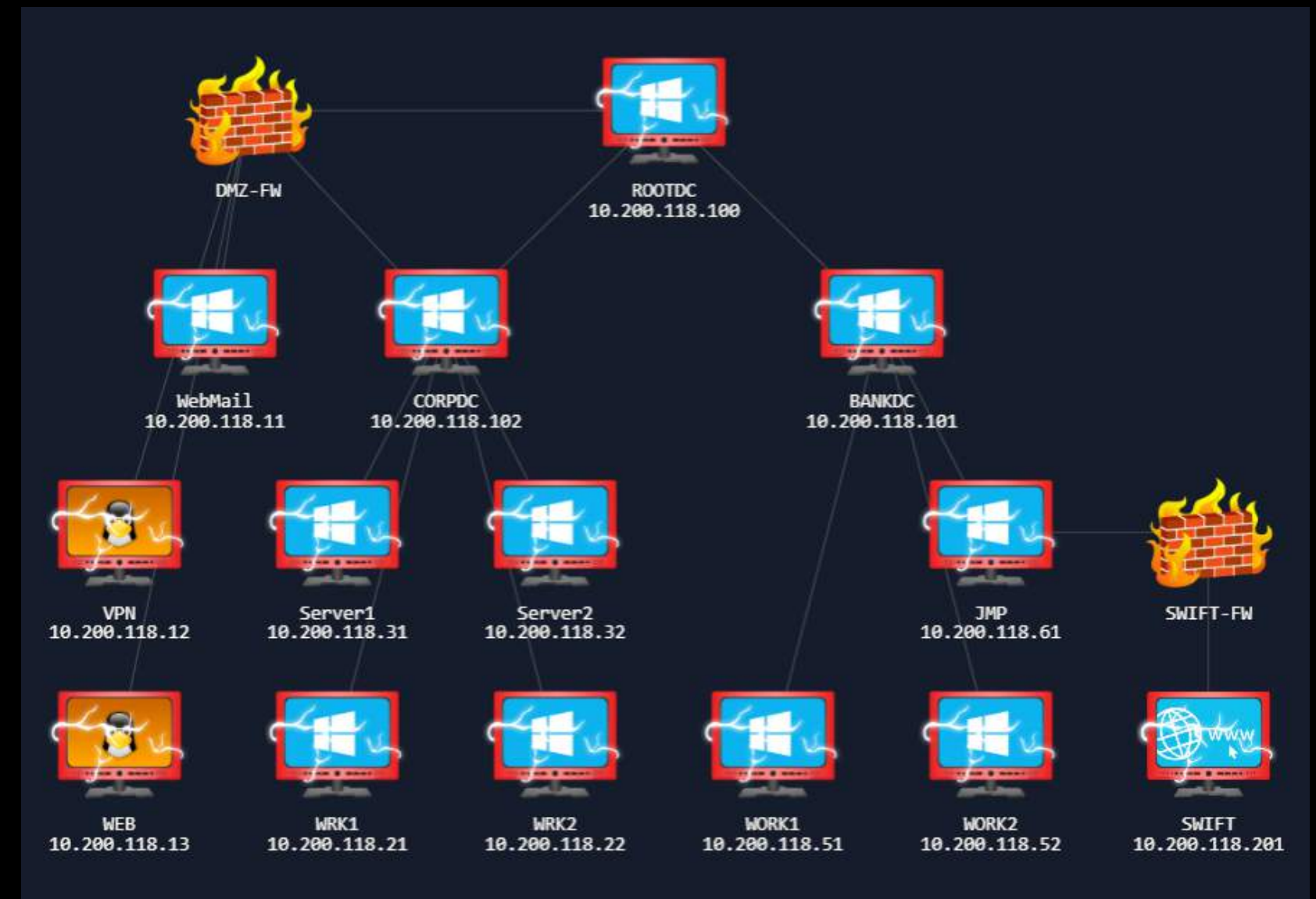




# LATERAL MOVEMENT AND DOMAIN-WIDE COMPROMISE

With forged tickets and admin access, I moved laterally to multiple critical systems including

- RootDC
- BankDC
- Swift
- JMP



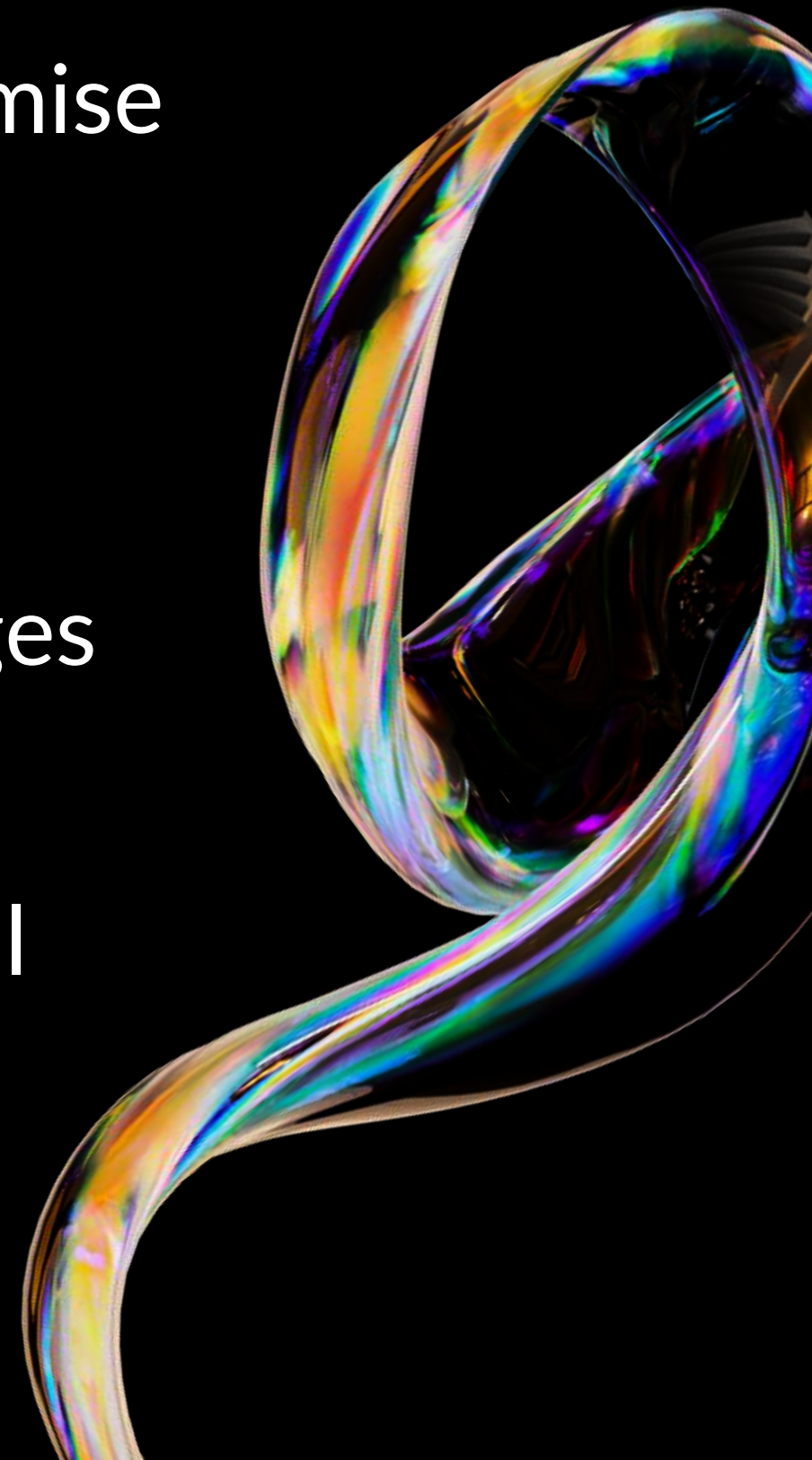
# IMPACT ANALYSIS

The engagement demonstrated complete domain compromise through chained misconfigurations

- Weak SPN passwords
- Lack of segmentation between admin tiers
- No monitoring for ticket anomalies or AD group changes

Risk Level: ● Critical

The attacker was able to escalate from basic access to total domain control with minimal resistance.







# REMEDIATION RECOMMENDATIONS

- Enforce strong, complex passwords for all service accounts
- Monitor and alert on SPN ticket requests and anomalies
- Audit group membership changes regularly
- Implement network segmentation to isolate privilege tiers
- Use SIEM rules to detect tools like Mimikatz and Rubeus



**THANK YOU**

