

From attack to defense: Exploration of five cyber threats that can break organizations



Shehnaaz

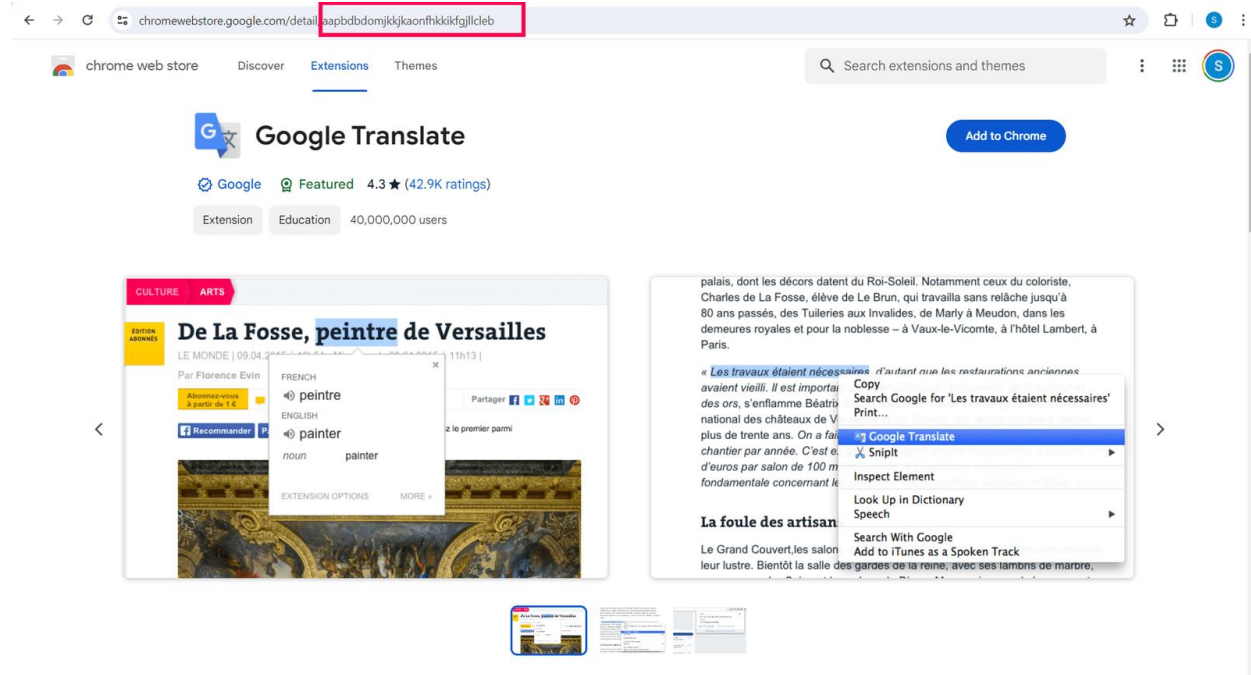
shehnaaz.n@manageengine.com



No fluff, just different attacks and their detection strategies

1. Attacking your organization using web browser based threats

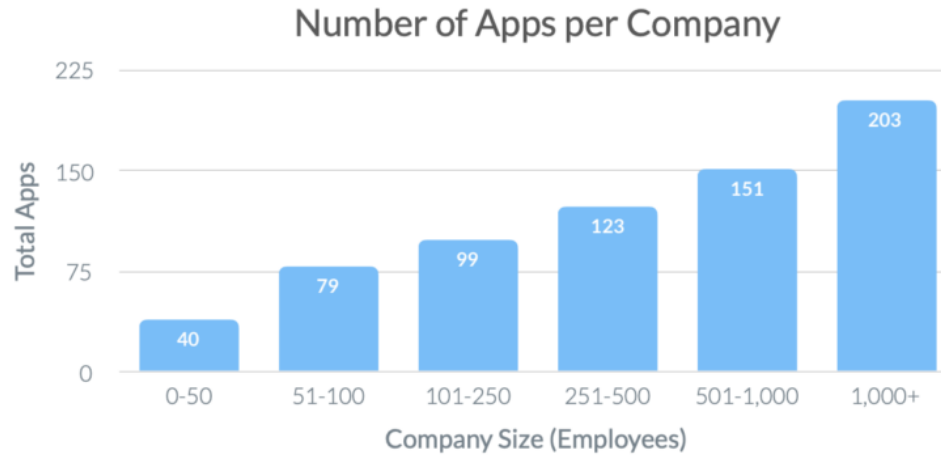
Snapshot IDs, anyone?



List of legitimate snapshot IDs

Extension Name	Extension Link
Adobe Acrobat	https://chrome.google.com/webstore/detail/adobe-acrobat/efaidnbmnnnibpcapjcgciciefndmkaj?hl=en
Checker Plus for Google Calendar	https://chrome.google.com/webstore/detail/checker-plus-for-google-c/hkhggnncdpfibdhinjiegagmopdiibha?hl=en
ChromeVox Classic Extension	https://chrome.google.com/webstore/detail/chromevox-classic-extensi/kgejghljpiefppelmijgicjbhoiplfn?hl=en
Cisco Webex Content Sharing	https://chrome.google.com/webstore/detail/cisco-webex-content-shari/iftbadgbpalmagalaclifaffakmfkac
Cisco Webex Extension	https://chrome.google.com/webstore/detail/cisco-webex-extension/jlhmfgmfgeifomenelglieieghnjghma
Cisco Webex Scheduler	https://chrome.google.com/webstore/detail/cisco-webex-scheduler/ankblejcdiejecjfagjeoelaedaoka
DeskPins (Individual approval needed)	https://efotinis.neocities.org/deskpins/
DisplayNote Extension	https://chrome.google.com/webstore/detail/displaynote-extension/dbfegieohdkcnaomlgefclpnbdiilkh
Gmail App	https://chrome.google.com/webstore/detail/gmail/pkijhegncpnkpknbcohdijoejaedia?hl=en
Google Docs Offline Chrome Extension	https://chrome.google.com/webstore/detail/google-docs-offline/ghbmnnjooekpmoecnniinbndiolhkh
Google Keep Chrome Extension	https://chrome.google.com/webstore/detail/google-keep-chrome-extens/lncadrmchfhochbanmchmfnfnhiddi

Organizations on an average use **110 SaaS applications per day.**



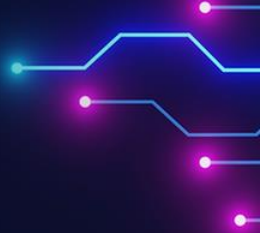
95% cloud usage happens outside your IT team's control.



Stealing data from your SaaS applications



AI-driven data leaks



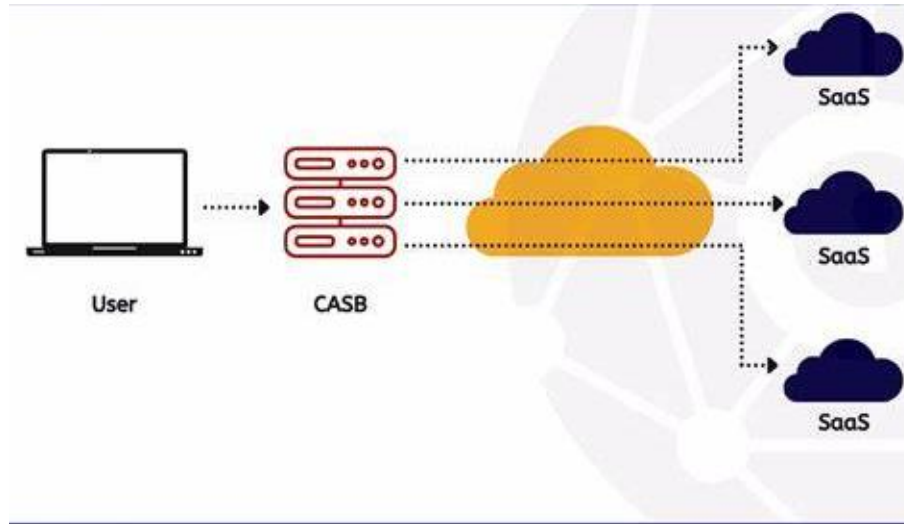
What is a CASB?

- **Gartner** defines the cloud access security broker (CASB) market as products and services that address security gaps in an organization's use of cloud services.
- They deliver differentiated, cloud-specific capabilities generally not available as features in other security controls such as web application firewalls (WAFs) and enterprise firewalls.

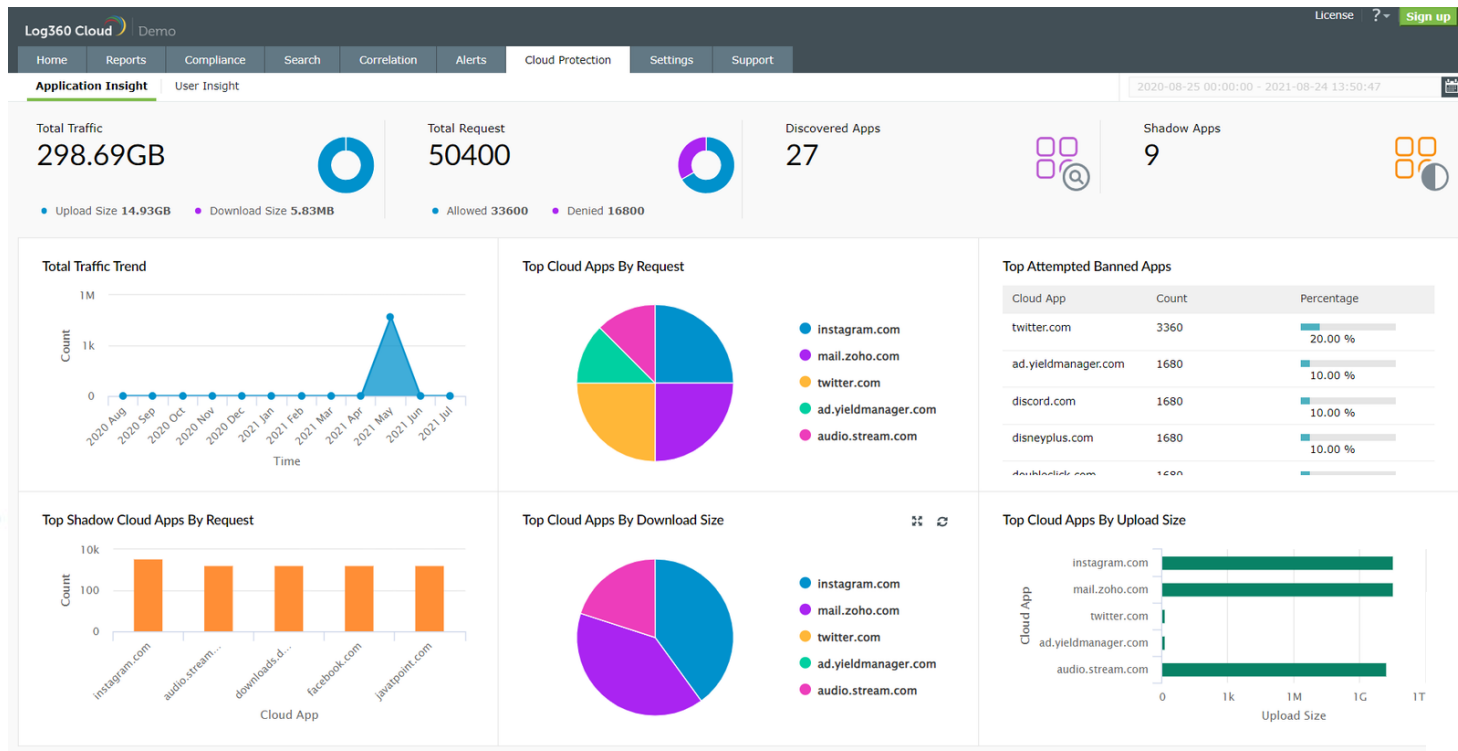


What exactly is a CASB?

A CASB is an intermediary that sits between a cloud service consumer and a cloud service provider.



Tracking application access using CASB



Getting notified for access of shadow and banned applications

The screenshot displays the Log360 Cloud interface with the 'Alert Criteria Builder' dialog box open. The background shows the 'Add Alert Profile' section with fields for Name, Severity, Log source, Criteria, and Alert Message Format. The dialog box has tabs for 'Predefined Alert Criteria', 'Compliance Alert Criteria', and 'Custom Alert Criteria'. The 'Predefined Alert Criteria' tab is active, showing a list of criteria under the 'Common Cloud Apps' group and 'Shadow Cloud Apps' category. The list includes 'Recent Sanctioned Apps Request' and 'Recent Shadow Apps Request'. The dialog box has 'Save' and 'Cancel' buttons at the bottom.

Alert Criteria Builder

Predefined Alert Criteria ? Compliance Alert Criteria ? Custom Alert Criteria ?

Group: Common Cloud Apps Category: Shadow Cloud Apps

1 - 2 of 2 10 ▾

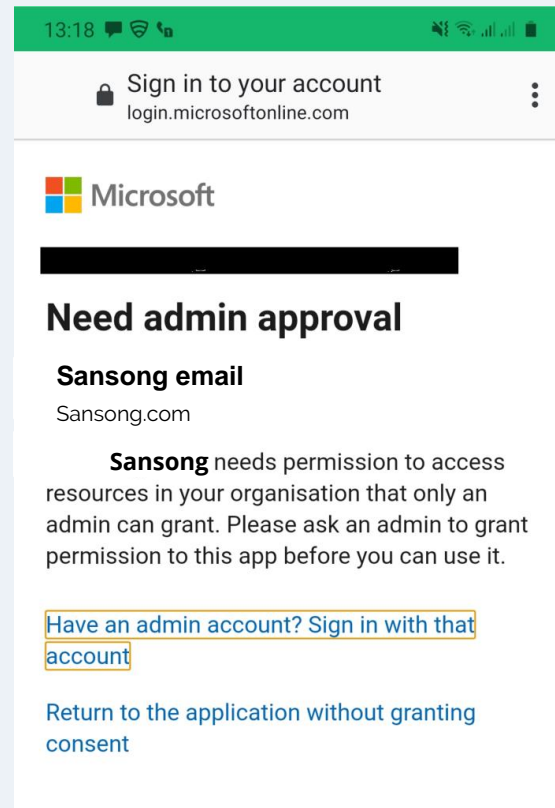
Criteria ▾	Description ▴
<input type="radio"/> Recent Sanctioned Apps Request	Recent Sanctioned Apps Request
<input type="radio"/> Recent Shadow Apps Request	Recent Shadow Apps Request

Save Cancel

2. Azure consent grant attack

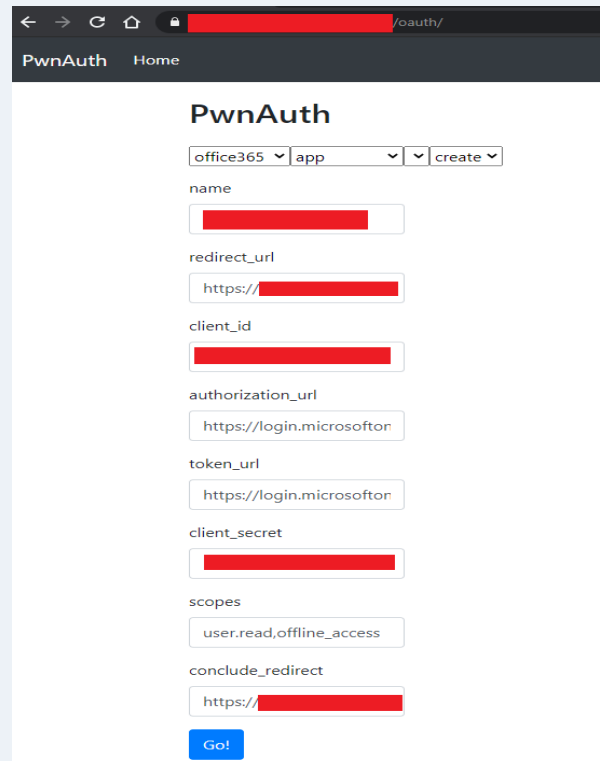
Nuanced Phishing Emails

- ✓ **Illicit consent grants:**
- ✓ Consent is the process of granting authorization to an Azure AD application to access protected resources on the users' behalf.
- ✓ Attackers can send illicit consent grant requests to users' inboxes and they get complete access if the user grants the unknown application a set of privileges.



Let's go phishing!

- ✓ **PwnAuth to launch the phishing campaign:**
- ✓ Enter the token URL and authorization URL from the source tenant and in the Scopes field, enter the names of all API permissions you (the attacker) wants to request.



The screenshot shows the PwnAuth web interface in a browser. The browser's address bar shows a redacted URL ending in "/oauth/". The page has a dark header with "PwnAuth" and "Home" links. The main content area is titled "PwnAuth" and contains a form with the following fields:

- A dropdown menu set to "office365" and a button labeled "create".
- A "name" field with a redacted value.
- A "redirect_url" field with a redacted value.
- A "client_id" field with a redacted value.
- An "authorization_url" field with the value "https://login.microsoft".
- A "token_url" field with the value "https://login.microsoft".
- A "client_secret" field with a redacted value.
- A "scopes" field with the value "user.read,offline_access".
- A "conclude_redirect" field with a redacted value.
- A blue "Go!" button at the bottom.

Illicit consent-grant attack

- ✓ **PwnAuth to launch the phishing campaign:**
- ✓ The victim will receive a consent-grant pop-up on which will prompt them to grant access to the application. All of this can happen without the user connecting to the corporate network.



Sansong.com

Need admin approval



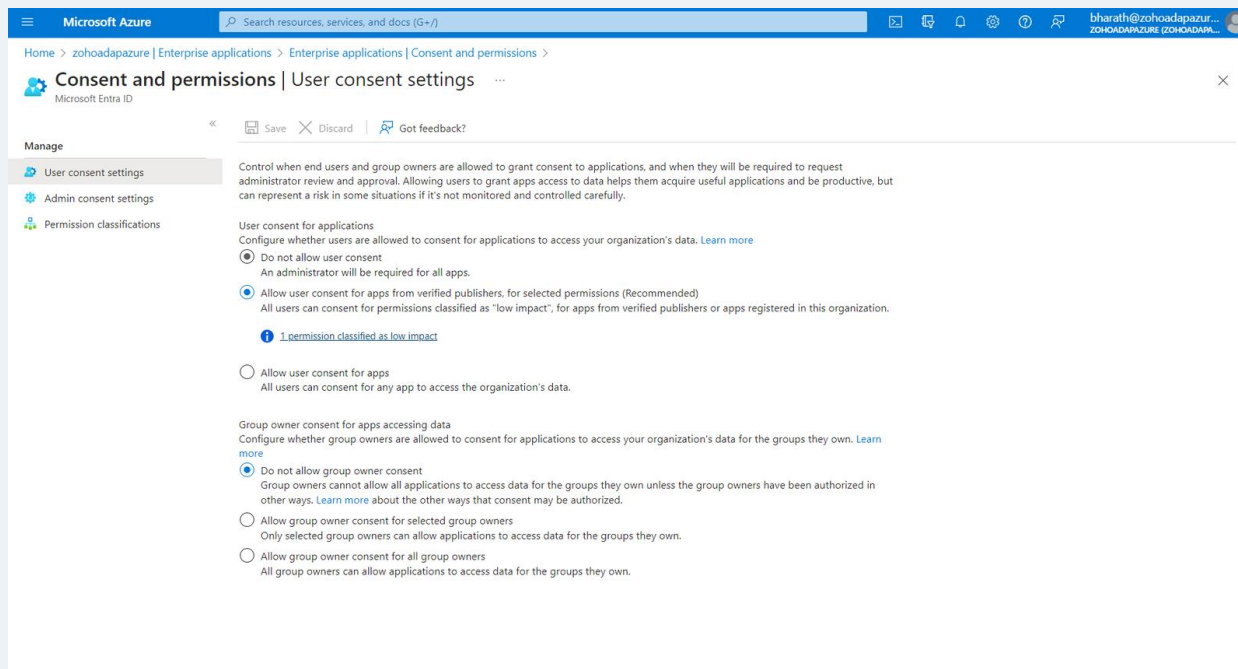
Sansong email

Sansong App needs permission to access resources in your organization that only an admin can grant. Please ask an admin to grant permission to this app before you can use it.

[Have an admin account? Sign in with that account](#)

[Return to the application without granting consent](#)

Curbing illicit access grant attempts from Azure AD: Fixing mistakes at the native tool.



Enterprise applications > Consent and permissions

Detecting consent-grant attacks using PowerShell

```
PS C:\WINDOWS\system32> Get-AzureADOAuth2PermissionGrant | Where-Object { $_.ClientId -eq 'bb313523-9bec-...' } | Select-Object -Property *

ClientId      : bb313523-9bec-4b78-...
ConsentType    : AllPrincipals
ExpiryTime     : 21-12-2022 07:54:17
ObjectId       : IzUxu-ybeEux3h_uE3mJH2j_...
PrincipalId    :
ResourceId     : 74e5ff68-284a-4d54-...
Scope          : user_impersonation
StartTime      : 01-01-0001 00:00:00

ClientId      : bb313523-9bec-4b78-...
ConsentType    : AllPrincipals
ExpiryTime     : 21-12-2022 07:47:04
ObjectId       : IzUxu-ybeEux3h_uE3mJHwD_...
PrincipalId    :
ResourceId     : 66caff00-b3f3-469a-...
Scope          : offline_access User.Read openid Application.Read.All
StartTime      : 01-01-0001 00:00:00

ClientId      : bb313523-9bec-4b78-...
ConsentType    : AllPrincipals
ExpiryTime     : 21-12-2022 07:54:17
ObjectId       : IzUxu-ybeEux3h_uE3mJH574_...
PrincipalId    :
ResourceId     : 0e8df89e-7e1a-4328-...
Scope          : user_impersonation
StartTime      : 01-01-0001 00:00:00
```

PowerShell command to identify all apps and the consent grants a user has issued:

Get-AzureADPSPermissions.ps1 | Export-csv -Path "Permissions.csv" -NoTypeInfoation

Revoking consent-grant using PowerShell

```
Account          Environment TenantId          TenantDomain      AccountType
-----
Z[REDACTED]@Gti.com AzureCloud e[REDACTED]757fb ha[REDACTED]Gti.com User

PS C:\WINDOWS\system32> # Get Service Principal using objectId
$sp = Get-AzureADServicePrincipal -ObjectId "d8c94fda-ea95[REDACTED]821b3cc"

# Get all delegated permissions for the service principal
$spOAuth2PermissionsGrants = Get-AzureADOAuth2PermissionGrant -All $true | Where-Object { $_.clientId -eq $sp.ObjectId }

# Remove all delegated permissions
$spOAuth2PermissionsGrants | ForEach-Object {
    Remove-AzureADOAuth2PermissionGrant -ObjectId $_.ObjectId
}

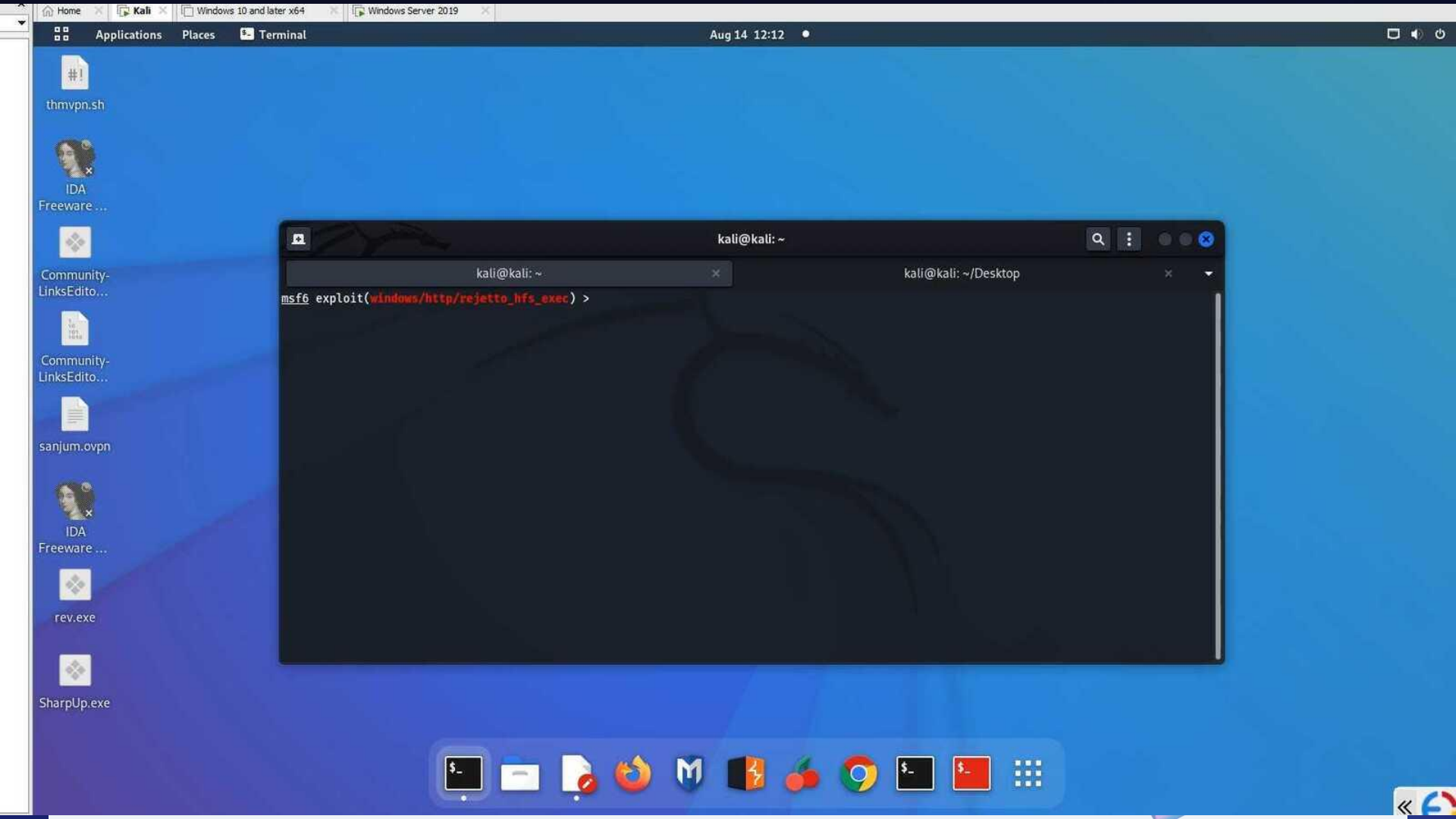
# Get all application permissions for the service principal
$spApplicationPermissions = Get-AzureADServiceAppRoleAssignedTo -ObjectId $sp.ObjectId -All $true | Where-Object { $_.Pr

# Remove all delegated permissions
$spApplicationPermissions | ForEach-Object {
    Remove-AzureADServiceAppRoleAssignment -ObjectId $_.PrincipalId -AppRoleAssignmentId $_.objectId
}
```

Revoking consent grants for applications using PowerShell commands:

Remove-AzureADOAuth2PermissionGrant

3. Domain user escalating their privilege to domain admin



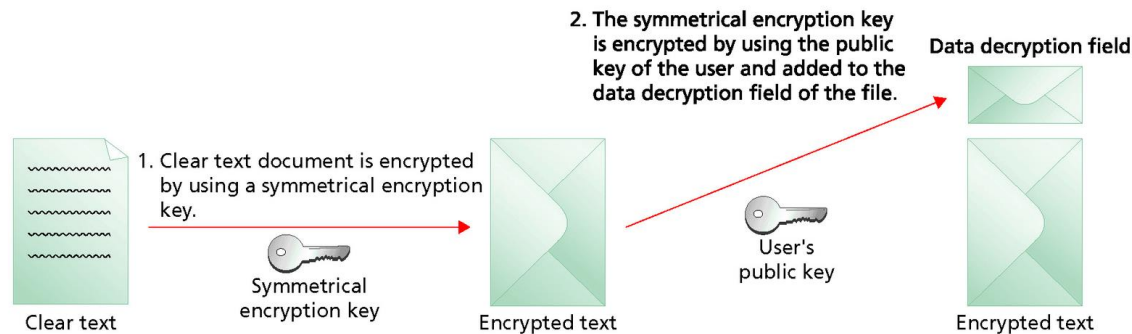
Tracking privilege escalation attempts

The screenshot displays the AD Audit Plus web interface. The top navigation bar includes tabs for Dashboard, Active Directory, Cloud Directory, File Audit, Server Audit, Endpoint, AD Backup, Analytics, Alerts, Configuration, Admin, and Support. A search bar and 'Domain Settings' link are on the right. The left sidebar lists various reports under 'Risk Assessment Reports', with 'Privilege Escalation - First time Utilizing a Privilege' selected. The main content area shows the title 'Privilege Escalation - First time Utilizing a Privilege' with a star icon, a date range '(From Jan 01,1970 07:30:00 AM to Jun 21,2024 11:40:34 AM)', and a domain dropdown set to 'admanagerplus.com'. Below this are filters for 'Period' (Before 90 Days) and 'Hours' (All [Business Hours]). Action buttons for 'Export As', 'Add to', and 'More' are present. A table titled 'Privilege Escalation - First time Utilizing a Privilege' contains three rows of data. The table has columns: CALLER USER NAME, LAST ACTIVITY TIME, PRIVILEGE UTILIZED, ACTIVITY MESSAGE, ACCOUNT NAME, SID, DOMAIN CONTROLLER, MODIFIED ATTRIBUTES, and DOMAIN. The first row shows a privilege escalation by DSP-DEMO\$. The second row shows a modification by Administrator. The third row shows a password change by Administrator.

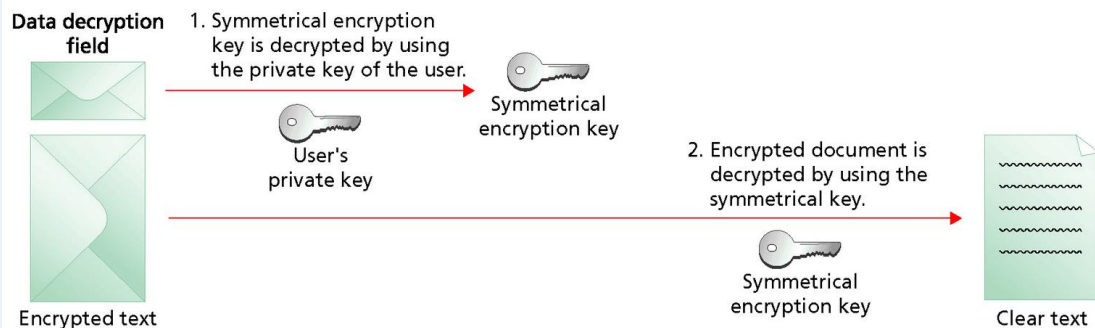
CALLER USER NAME	LAST ACTIVITY TIME	PRIVILEGE UTILIZED	ACTIVITY MESSAGE	ACCOUNT NAME	SID	DOMAIN CONTROLLER	MODIFIED ATTRIBUTES	DOMAIN
DSP-DEMO\$	Sep 22,2023 02:28:05 PM	Computer Attribute Added	Computer 'DSP-DEMO' was modified by 'ADMANAGERPLUS\DSP-DEMO\$'. Modified Properties : msDS-SupportedEncryptionTypes, Values : 28	DSP-DEMO	%(S-1-5-21-1484795863-58162057-4169609511-1370)	adapdemo.admanagerplus.com	msDS-SupportedEncryptionTypes	adman
Administrator	Sep 22,2023 02:27:25 PM	Computer Modified	Computer 'DSP-DEMO' was modified by 'ADMANAGERPLUS\Administrator'. Modified Properties : userAccountControl, Values : Account is disabled.This is a computer account for a Windows or Windows Server that is a member of this domain.	DSP-DEMO	%(S-1-5-21-1484795863-58162057-4169609511-1370)	adapdemo.admanagerplus.com	userAccountControl	adman
Administrator	Sep 22,2023 02:27:18 PM	A computer account was changed.	Computer account 'DSP-DEMO\$' was changed by 'ADMANAGERPLUS\Administrator'. Changed Attributes : 'Password Last Set'	DSP-DEMO\$	%(S-1-5-21-1484795863-58162057-4169609511-1370)	adapdemo.admanagerplus.com	Password Last Set	ADMAN

4. Abusing Windows Encrypting File System (EFS)

Encrypting a file



Decrypting a file



An insider remotely accesses a server and encrypts data

Firstly, the insider must have RDP access.

For RDP access, the user must escalate their privilege.

The attack chain of EFS ransomware attack

1. The ransomware generates a key (using AdvApi32!CryptGenKey) to be used by EFS and records the file name used by CAPI for this key.
2. The ransomware generates a certificate for this key and adds it to the personal certificate store using Crypt32!CertAddCertificateContextToStore.
3. The ransomware sets the current EFS key to this certificate using AdvApi32!SetUserFileEncryptionKey.
4. Now the ransomware can invoke AdvApi32!EncryptFile (using the generated certificate) on every file/folder to be encrypted.
5. The ransomware saves the key file to memory and deletes it from the following two folders:

%APPDATA% \Microsoft\Crypto\RSA\sid\ (where sid is the user SID)
%ProgramData% \Microsoft\Crypto\RSA\MachineKeys

The code snippet to generate a AES-key using PowerShell

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

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

PS C:\Users\shehnaaz-10133> function LoadCryptographyHelper {
>> Add-Type @"
>> using System;
>> using System.Runtime.InteropServices;
>>
>> public class CryptographyHelper {
>>     [DllImport("advapi32.dll", SetLastError=true, CharSet=CharSet.Auto)]
>>     public static extern bool CryptAcquireContext(out IntPtr hProv, string pszContainer, string pszProvider, int dwProvType, int dwFlags);
>>
>>     [DllImport("advapi32.dll", SetLastError=true)]
>>     public static extern bool CryptReleaseContext(IntPtr hProv, int dwFlags);
>>
>>     [DllImport("advapi32.dll", SetLastError=true)]
>>     public static extern bool CryptGenKey(IntPtr hProv, int Algid, int dwFlags, out IntPtr phKey);
>>
>>     public static int CALG_AES_256 = 0x6602; // AES 256-bit algorithm identifier
>>
>>     public static IntPtr CreateNewKey() {
>>         IntPtr hProv = IntPtr.Zero;
>>         IntPtr phKey = IntPtr.Zero;
>>
>>         // Acquire a cryptographic context
>>         bool result = CryptAcquireContext(out hProv, "MyKeyContainer", null, 1 /* PROV_RSA_AES */, 0x08 /* CRYPT_VERIFYCONTEXT | CRYPT_NEWKEYSET
*/);
>>         if (!result) {
>>             int error = Marshal.GetLastWin32Error();
>>             if (error == -2146893809) {
>>                 // If NTE_BAD_KEYSET error, try again with CRYPT_NEWKEYSET flag
>>                 result = CryptAcquireContext(out hProv, "MyKeyContainer", null, 1 /* PROV_RSA_AES */, 0x08 | 0x10 /* CRYPT_VERIFYCONTEXT | CRYPT_
NEWKEYSET */);
>>                 if (!result) {
>>                     throw new Exception("CryptAcquireContext failed. Error code: " + Marshal.GetLastWin32Error());
>>                 }
>>             } else {
>>                 throw new Exception("CryptAcquireContext failed. Error code: " + error);
>>             }
>>         }
>>
>>         // Generate a new AES 256-bit key
>>         if (!CryptGenKey(hProv, CALG_AES_256, 0, out phKey)) {
>>             CryptReleaseContext(hProv, 0);
>>             throw new Exception("CryptGenKey failed. Error code: " + Marshal.GetLastWin32Error());
>>         }
>>     }
>> }
```

```
Administrator: Windows Powe
>>
>>     }
>>
>>     // Generate a new AES 256-bit key
>>     if (!CryptGenKey(hProv, CALG_AES_256, 0, out phKey)) {
>>         CryptReleaseContext(hProv, 0);
>>         throw new Exception("CryptGenKey failed. Error code: " + Marshal.GetLastWin32Error());
>>     }
>>
>>     // Release the cryptographic context
>>     CryptReleaseContext(hProv, 0);
>>
>>     return phKey;
>> }
>> }
>> }
>> }
>> }
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Load the CryptographyHelper class
PS C:\Users\shehnaaz-10133> LoadCryptographyHelper
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Create a new AES 256-bit key using the helper class
PS C:\Users\shehnaaz-10133> try {
>>     $key = [CryptographyHelper]::CreateNewKey()
>>     Write-Output "AES 256-bit key created successfully."
>>     Write-Output "Key handle: $key"
>> }
>> catch {
>>     Write-Error "Error creating AES 256-bit key: $_"
>> }
>> }
AES 256-bit key created successfully.
Key handle: 2175093063984
PS C:\Users\shehnaaz-10133>
```

Rain showers
At night

Search

ENG IN 16:51 03-07-2024

Code snippet to generate a certificate and plant it in certmgr.exe using the generated key

```
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:\Users\shehnaaz-10133> # Define the AES key handle
PS C:\Users\shehnaaz-10133> $keyHandle = 2440629188752
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Create a random RSA key pair for certificate signing (RSA is used for certificate operations, not AES)
PS C:\Users\shehnaaz-10133> $rsaKey = New-Object System.Security.Cryptography.RSACryptoServiceProvider 2048
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Create a Certificate Request
PS C:\Users\shehnaaz-10133> $subjectName = "CN=MyTestCert"
PS C:\Users\shehnaaz-10133> $certRequest = New-Object System.Security.Cryptography.X509Certificates.CertificateRequest($subjectName, $rsaKey, [System.Security.Cryptography.HashAlgorithmName]::SHA256, [System.Security.Cryptography.RSASignaturePadding]::Pkcs1)
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Add basic constraints extension (usually included in self-signed certificates)
PS C:\Users\shehnaaz-10133> $certRequest.CertificateExtensions.Add((New-Object System.Security.Cryptography.X509Certificates.X509BasicConstraintsExtension($true, $false, 0, $true)))
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Self-sign the certificate
PS C:\Users\shehnaaz-10133> $cert = $certRequest.CreateSelfSigned([System.DateTime]::Now, [System.DateTime]::Now.AddYears(10))
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Convert the AES key handle to a byte array for storage in the certificate's extensions
PS C:\Users\shehnaaz-10133> $keyHandleBytes = [BitConverter]::GetBytes($keyHandle)
PS C:\Users\shehnaaz-10133> $keyHandleExtension = New-Object System.Security.Cryptography.X509Certificates.X509Extension("1.2.3.4", $keyHandleBytes, $false)
PS C:\Users\shehnaaz-10133> $cert.Extensions.Add($keyHandleExtension)
1
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> # Add the certificate to the Personal store
PS C:\Users\shehnaaz-10133> $store = New-Object System.Security.Cryptography.X509Certificates.X509Store("My", "CurrentUser")
PS C:\Users\shehnaaz-10133> $store.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadWrite)
PS C:\Users\shehnaaz-10133> $store.Add($cert)
PS C:\Users\shehnaaz-10133> $store.Close()
PS C:\Users\shehnaaz-10133>
PS C:\Users\shehnaaz-10133> Write-Output "Successfully created and added certificate to store."
Successfully created and added certificate to store.
PS C:\Users\shehnaaz-10133> Write-Output "Certificate Details:"
Certificate Details:
PS C:\Users\shehnaaz-10133> Write-Output $cert

Thumbprint                               Subject
-----
B8C3978AFF40C3A2C2B3E17B5D3A1C9B7829659B CN=MyTestCert

PS C:\Users\shehnaaz-10133>
```

certmgr - [Certificates - Current User/Personal/Certificates]

File Action View Help

Certificates - Current User

- Personal
 - Certificates
 - Trusted Root Certification Authorities
 - Enterprise Trust
 - Intermediate Certification Authorities
 - Active Directory User Objects
 - Trusted Publishers
 - Untrusted Certificates
 - Third-Party Root Certification Authorities
 - Trusted People
 - Client Authentication Issuers
 - Other People
 - Adobe CertStore
 - Smart Card Trusted Roots

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Status	Certificate Template
Adobe Content Certificate 10-5	Adobe Intermediate CA 10-3	18-06-2025	<All>	<None>		
Adobe Content Certificate 10-6	Adobe Intermediate CA 10-4	18-06-2025	<All>	<None>		
Adobe Content Certificate 10-7	Adobe Intermediate CA 10-15	05-08-2030	<All>	<None>		
Adobe Content Certificate 10-8	Adobe Intermediate CA 10-19	05-08-2030	<All>	<None>		
Adobe Intermediate CA 10-15	Adobe Root CA 10-3	04-08-2068	<All>	<None>		
Adobe Intermediate CA 10-19	Adobe Root CA 10-3	04-08-2068	<All>	<None>		
Adobe Intermediate CA 10-3	Adobe Root CA 10-3	04-08-2068	<All>	<None>		
Adobe Intermediate CA 10-4	Adobe Root CA 10-3	04-08-2068	<All>	<None>		
MyTestCert	MyTestCert	03-07-2034	<All>	<None>		
shehnaaz-10133	shehnaaz-10133	08-06-2124	Encrypting File Syst...	<None>		

Certificate

General Details Certification Path

Show: <All>

Field	Value
Version	V3
Serial number	00d2646e0b507abce3
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	MyTestCert
Valid from	03 July 2024 20:47:42
Valid to	03 July 2034 20:47:42
Subject	MyTestCert

Edit Properties... Copy to File...

OK

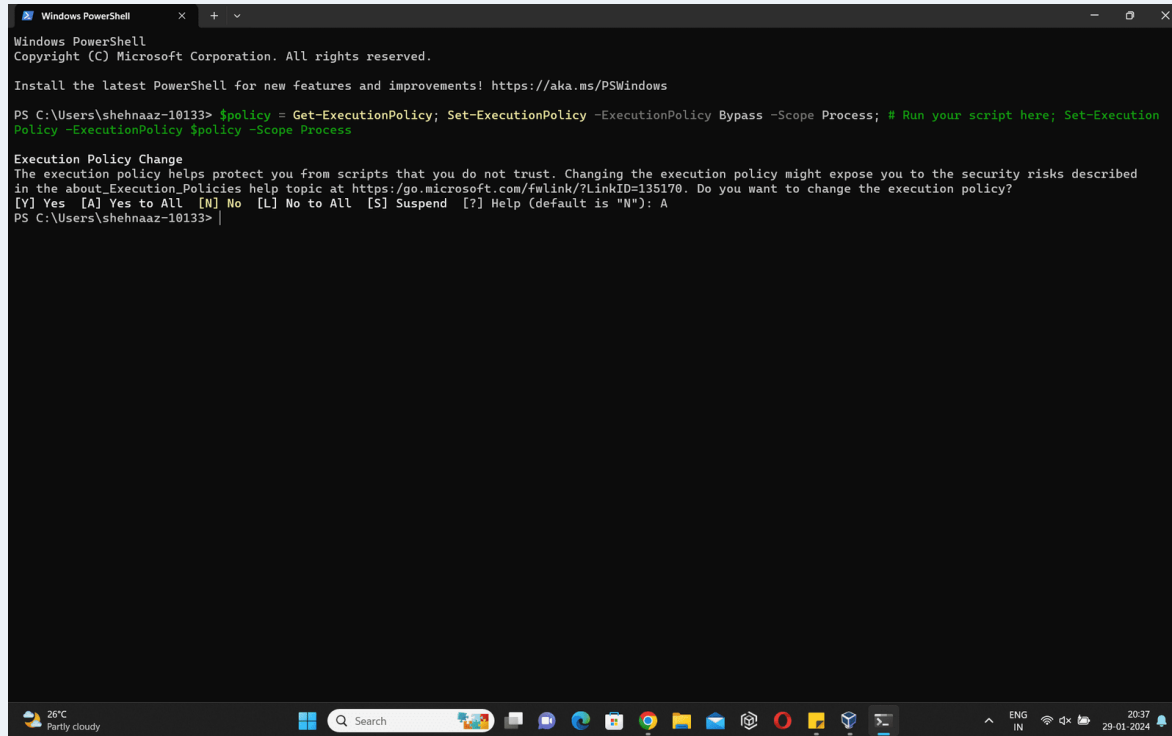
31°C Haze

Search

ENG IN

20:51 03-07-2024

PowerShell script to delete all the data:



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

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

PS C:\Users\shehnaaz-10133> $policy = Get-ExecutionPolicy; Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process; # Run your script here; Set-ExecutionPolicy -ExecutionPolicy $policy -Scope Process

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): A
PS C:\Users\shehnaaz-10133> |
```


Tracking PowerShell activities using Log360

The screenshot displays the AD Audit Plus web interface. The top navigation bar includes links for Dashboard, Reports, Azure AD, File Audit, Server Audit, Analytics, Alerts, Configuration, Admin, and Support. A search bar and a 'Domain Settings' link are also present. The left sidebar lists various audit categories, with 'Logon Audit' selected. The main content area is titled 'Powershell Audit' and shows a table of audit events for the period 'Last 30 Days'.

Powershell Audit
(From Dec 31, 2023 01:09:12 PM to Jan 30, 2024 01:09:12 PM)

Period: Hours:

Export As ★ Add to More

Advanced Search 1-25 of 60822 25 Add/Remove Columns

USER NAME	TIME GENERATED	COMMAND NAME	COMMAND TYPE	ADAP.AUDCOLUMNCONFIGURATION.DISPLAYNAME.SOURCE	EVENT NUMBER	HOST APPLICATION	HOST NAME	TOTAL NUMBER OF EVENTS
dsp	Jan 30, 2024 12:20:45 PM	-	-	DSP-DEMO	4104	-	-	1
dsp	Jan 30, 2024 12:20:45 PM	-	Script	DSP-DEMO	4103	powershell.exe \FapEvent.ps1 -file C:\DSP\Scripts\file_audit_script\adapdc3.csv -count 100	ConsoleHost	0
dsp	Jan 30, 2024 12:20:45 PM	Write-Host	Cmdlet	DSP-DEMO	4103	powershell.exe \FapEvent.ps1 -file C:\DSP\Scripts\file_audit_script\adapdc3.csv -count 100	ConsoleHost	0

Tracking encryption using the correlation module

The screenshot displays the Log360 'Create Custom Action' dialog box. The dialog has a title bar with 'Create Custom Action' and a close button. It features a 'Back' button in the top right corner. The main content area includes an 'Action name' field with the value 'Sensitive privilege use' and an 'Add description' link. Below this is a criteria selection area with a dropdown for 'Event ID', a comparison operator 'Equals', and a value '4673'. A green '+' button is next to the value. The criteria pattern is displayed as '(Event ID : 4673)' with an 'Add group' link. At the bottom of the dialog are 'Create' and 'Cancel' buttons. The background shows the Log360 interface with a sidebar menu containing categories like Overview, SIEM, AD Audit, MSIS, UEBA, Exchange, AD Management, and Cloud Security. The main content area shows a search bar and a list of events.



5. One more way to take down your organization



Questions ?

Bingo Code: 1503





Thank you



Shehnaaz

shehnaaz.n@manageengine.com

