# Assessment Details

**Scope**

The scope of this assessment was limited to **one (1)** IPv4 address within **Timelapse Inc**.’s internal network. Being a black-box style engagement, the hostname of the target, and its associated domain, were not disclosed to us directly. These values were appended to the scope upon discovery as agreed upon by **Timelapse Inc.** Below is a comprehensive list of all items within scope of the engagement.

|  |  |
| --- | --- |
| **In-Scope Assets** | |
| **10.10.11.152** | Target IP Address |
| **DC01.timelapse.htb** | Target Hostname |
| **timelapse.htb** | Target Domain |

**Internal Network Assessment Summary**

**Stop the Xploit** began all testing activities from a black-box perspective, emulating an unauthenticated user on the internal network. **Timelapse Inc.** provided **Stop the Xploit** with a single IPv4 address, but did not provide additional information such as operating system, domain, or hostname. Utilizing common network enumeration tools, such as **ping** and **nmap**, we were able to quickly map out our environment and determine that our target was a Windows Domain Controller, **DC01.timelapse.htb**, in the **timelapse.htb** domain.

**Key Findings**

During the course of our assessment period, **Stop the Xploit** discovered a total of **five (5)** findings which possess a threat to **Timelapse Inc.** information systems. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. These insights indicate areas for betterment and don't equate to immediate security risks. The table below categorizes these insights by their criticality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Finding Severity** | | | | |
| **Critical** | **High** | **Medium** | **Low** | **Total** |
| **0** | **1** | **4** | **0** | **5** |

<FINDINGS SUMMARY>

.

.

.

.

.

.

.

.

.

</>

## **Testing Timeline**

**Domain Controller Compromise Overview**

During the course of the engagement, an analyst for **Stop the Xploit** was able to gain an initial foothold on the network’s Domain Controller. Through a series of attacks, our analyst was then able to escalate their privileges to a Domain Administrator account, granting our analyst full control over the **timelapse.htb** domain. The steps below demonstrate the attack chain utilized to achieve this, from gaining initial access to compromising the Domain Controller. This attack chain aims to represent the shortest path to compromise an adversary could take, and thus contains only the actions necessary to achieve that goal. The purpose of demonstrating these procedures is to not only exhibit the overall impact each vulnerability has, but to better convey the risks associated with not remediating these weaknesses as well.

The section below is **not** a comprehensive summary of our findings, and any discoveries not related to this attack chain are listed in the **Findings** section. Leveraging additional vulnerabilities in this section may lead to similar levels of compromise, however, we find the attack chain detailed below to possess the greatest potential of being exploited.

**Attack Narrative**

*Stop the Xploit performed the following steps to fully compromise the* ***timelapse.htb*** *domain.*

1. Our analyst used **smbclient** to gain anonymous access to a network share where they were able to exfiltrate a password-protected zip archive.

2. The zip archive’s password was successfully cracked using the hash cracking tool, **JohnTheRipper**. A password-protected pfx file was extracted, which our analyst was also able to crack using the same tool. The public and private key pair for the **legacyy** user was extracted from the pfx file with **openssl**, and our analyst was then able to login to the system using **evil-winrm**.

3. While the **legacyy** user had limited privileges on the system, our analyst was able to locate a clear text password for the **svc\_deploy** account in **legacyy**’s PowerShell history.

4. With the newly acquired **svc\_deploy** credentials, our analyst used **bloodhound-python** to run a series of authenticated LDAP queries against the Domain Controller. Our analyst then imported this data into **BloodHound CE**, where they discovered that the **svc\_deploy** account is in the **LAPS\_READERS** group. This group grants members the ability to read the Local Administrator password via the **ReadLAPSPassword** permission.

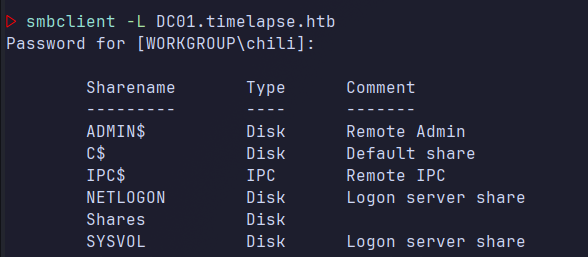
6. Using the multi-protocol enumeration tool, **crackmapexec**, our analyst was able to take advantage of the **ReadLAPSPassword** ability and retrieve the Local Administrator password for the Domain Controller.

7. Using **evil-winrm**, our analyst was able to successfully authenticate to the Domain Controller as **Administrator**. With the validity of the discovered administrator credentials confirmed, our analyst had now fully compromised the **timelapse.htb** domain.

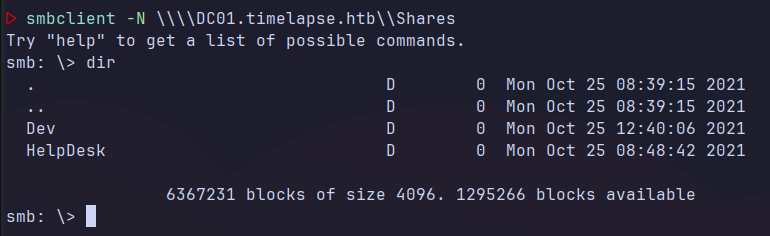
**Attack Chain Reproduction Steps**

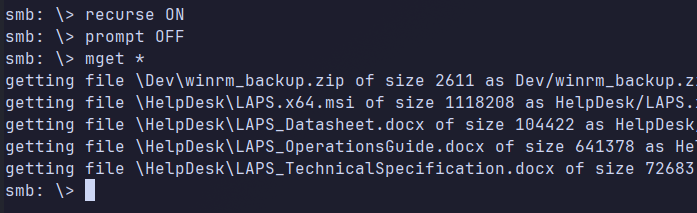
*The following section consists of the steps required to reproduce the attack chain described above.*

After initial enumeration of our target was complete, our analyst used **smbclient** to list any SMB shares present on the host.

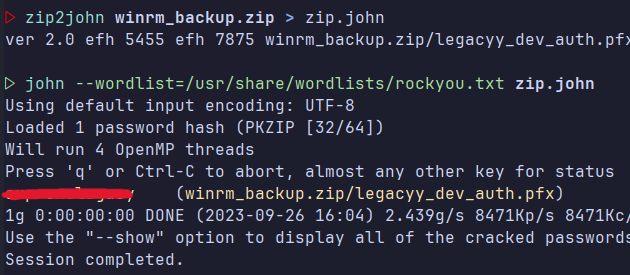


Upon further investigation, our analyst discovered that they were able to gain access to the “Shares” SMB share via Anonymous Login. Anonymous Login allows an individual without credentials to access shares anonymously. The analyst was then able to recursively download every file within the share to their local machine for further analysis.

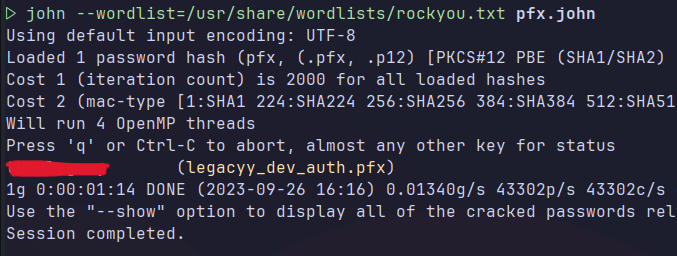




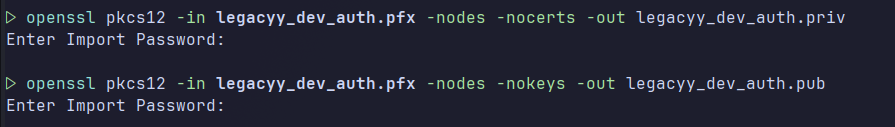
Upon review of the extracted SMB share, the analyst was able to discover a password protected zip archive titled, “winrm\_backup.zip”. Using a tool called **zip2john**, the analyst was able to retrieve the archive’s password hash and then crack it using **JohnTheRipper**.



The analyst was able to extract a Personal Information Exchange (PFX) file from the archive, which they were then able to crack using **pfx2john** and **JohnTheRipper**. PFX files are extremely sensitive as they contain both an entity's public and private keys.

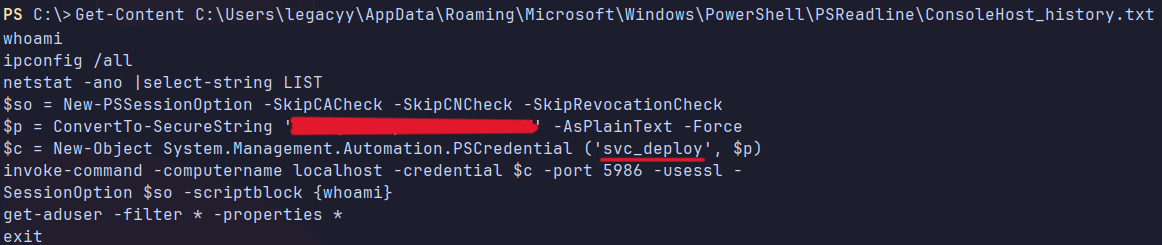


After obtaining the PFX file’s password, our analyst used **openssl** to extract the public and private keys into their own separate files. They were then able to use this key pair in **evil-winrm** to gain a foothold on the Domain Controller as the **legacyy** user.

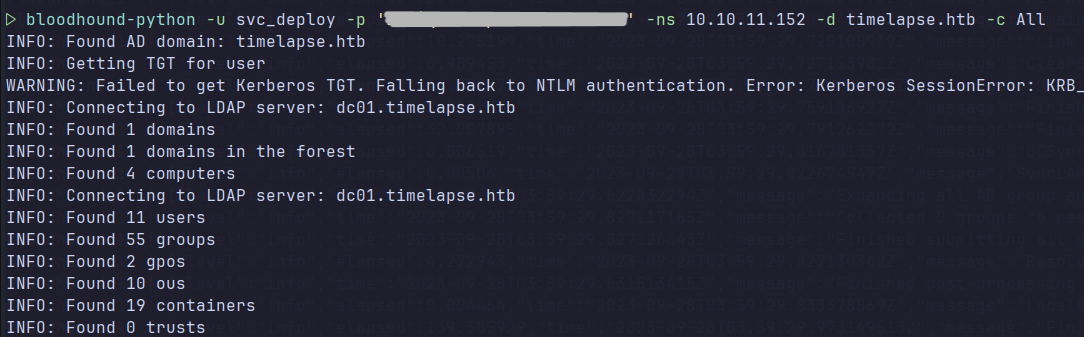




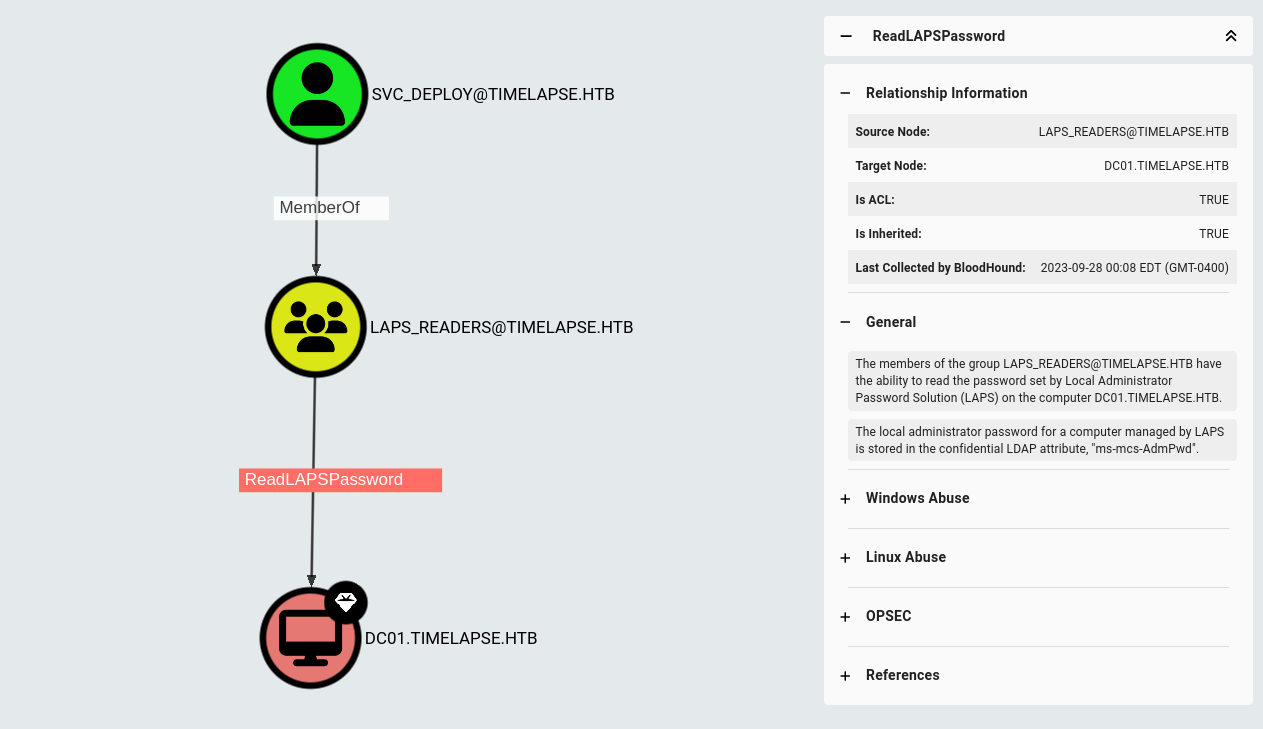
Our analyst next began enumerating the **legacyy** user’s privileges as well as searching for any potential vectors for lateral movement. Upon dumping the contents of the **legacyy** user’s PowerShell history, our analyst discovered that cleartext credentials for another user, **svc\_deploy**, were stored inside.



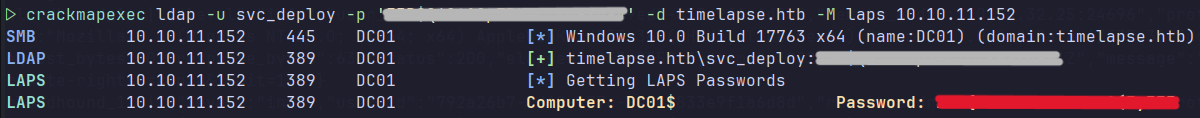
With the newly acquired service credentials, our analyst used **bloodhound-python** to run a series of authenticated LDAP queries against the Domain Controller. Light-Weight Directory Access Protocol (LDAP) allows for rapid queries of directory objects, such as users, workstations, policies, etc., over the network.



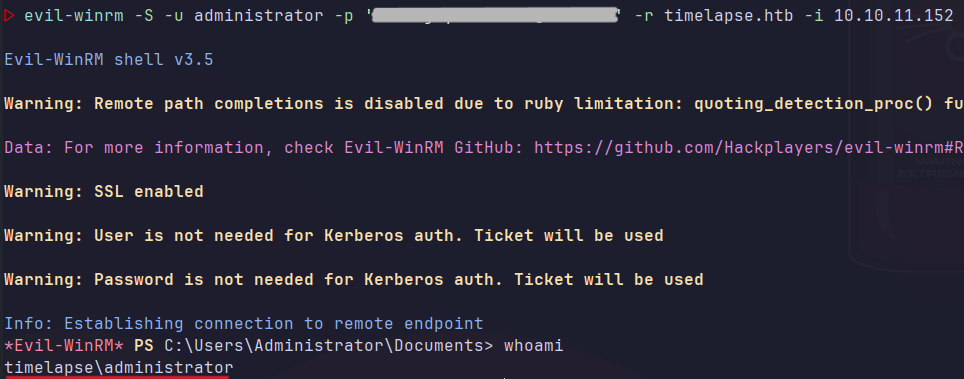
Using the Active Directory enumeration tool, **BloodHound CE**, our analyst ingested the LDAP query data to begin looking for ways to escalate their privileges. It was discovered that the **svc\_deploy** account is a member of the **laps\_readers** group, which has the privilege “ReadLAPSPassword” enabled. Objects with this privilege have the ability to read the password set by the Local Administrator Password Solution (LAPS).



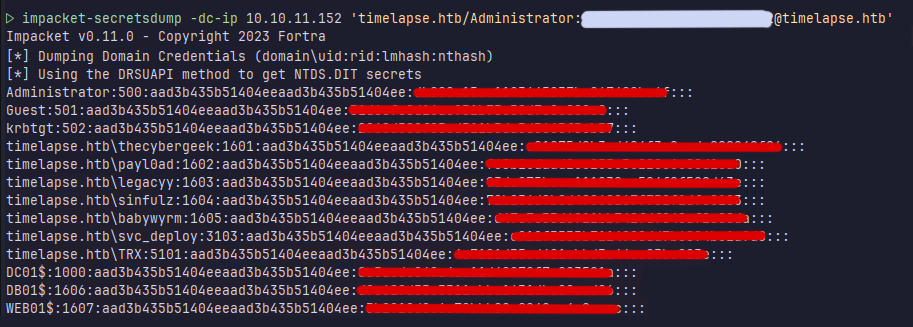
To confirm that it was possible to read the Local Administrator password, our analyst sent a specialized LDAP query to the Domain Controller using a tool called **crackmapexec**.



Using **evil-winrm**, our analyst was able to successfully authenticate to the Domain Controller as **Administrator**. With the validity of the discovered administrator credentials confirmed, our analyst had now fully compromised the Domain Controller and the **timelapse.htb** domain was now under their complete control.



With full control of the DC, it was possible to retrieve NTLM hashes for all users in the domain using **impacket-secretsdump**. While not in scope for this engagement, our analyst could now attempt to crack the hashes locally with **hashcat,** or try another tactic such as a pass-the-hash attack.



**Remediation Summary**

Due to several misconfigurations on the Domain Controller, the **timelapse.htb** domain is at high risk of being compromised should an unauthorized person gain access to the internal network. This also makes the domain highly susceptible to insider-threat attacks, as only initial access to the internal network is required to initiate the attack chain. With a few minor adjustments, the overall security posture of **Timelapse Inc.**’s internal networkcan be vastly improved.

**Recommendations**

**Stop the Xploit**’s recommendations for resolving these issues can be found below. Items with the shortest remediation time are presented at the top of the list, while longer-term solutions appear at the bottom. To help prevent unwanted downtime or data loss, **Timelapse Inc.** should sufficiently test **any** modifications in a separate environment prior to implementing them in production. Backups of any configured policies and/or snapshots of the system should also be created to allow for easier rollback if problems occur.

**Short Term**

* Revoke the certificate associated with the **legacyy** user’s PFX file and generate new keys.
* Remove the zip archive containing the **legacyy** user’s PFX file from the public share.
* Purge any PowerShell history files containing cleartext credentials or sensitive data.
* Renew and replace the expired Domain Controller certificate.
* Disable deprecated TLS v1.0 and TLS v1.1 Protocols
* Enforce a password change for all users and service accounts.

**Medium Term**

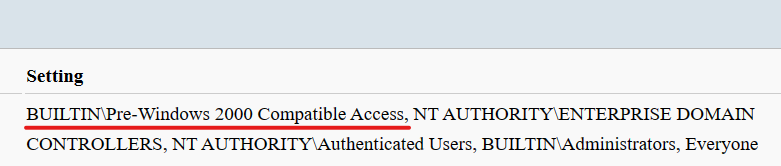
* Enforce the usage of secure strings when passing sensitive data through powershell or in scripts.
* Implement a solution such as the Microsoft Local Administrator Password Solution" (LAPS)
* Consider implementing an enterprise PKI Management Solution
* Perform a network file share audit
* Enhance network logging and monitoring
* Renew and replace all expired or invalid cryptographic keys still being used within the domain.

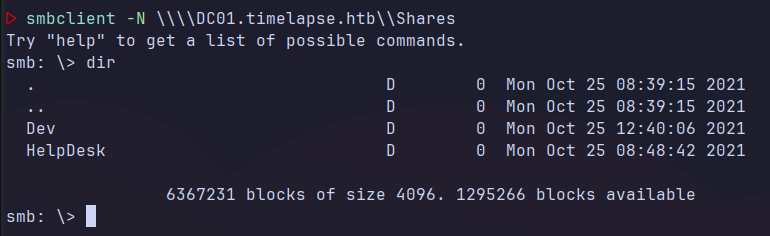
**Long Term**

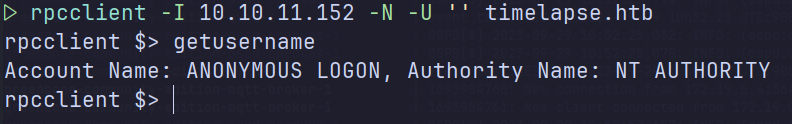
* Perform ongoing internal network vulnerability assessments and domain password audits
* Perform periodic Active Directory security assessments
* Educate systems and network administrators and developers on security hardening best practices
* Enhance network segmentation to isolate critical hosts and limit the effects of an internal compromise

**Findings**

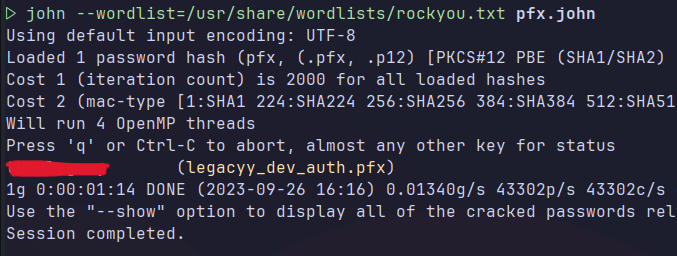
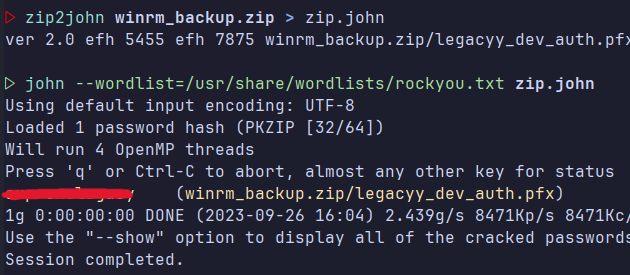
|  |  |
| --- | --- |
| **SMB/NETBIOS NULL Session Authentication** | |
| **CWE** | [CWE-287](https://cwe.mitre.org/data/definitions/287.html) |
| **CVSS v3.1 Score** | 7.3 -High |
| **CVSS v3.1 Vector** | AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L/E:H/RL:O/RC:C |
| **Description** | In Microsoft Windows environments, network file sharing occurs over the Server Message Block (SMB) protocol. Utilizing SMB, a user or application may read, create, and update resources located on a remote server.  In situations where remote resources must be accessible to unauthenticated individuals, SMB can be configured to allow NULL Session Authentication. This legacy authentication method allows clients to access shares as an Anonymous User without providing any login credentials. NULL Session Authentication is disabled by default in newer Windows Operating Systems and enabling it may lead to the exposure of sensitive data.  SMB shares may be enumerated with common tools such as **smbclient** and **smbmap**. |
| **Impact** | When SMB Null Session Authentication is enabled, remote shares may be enumerated or accessed by unauthenticated users. This allows attackers to discover network resources anonymously, making it very difficult to provide any accountability for the actions performed on the server. This issue is further exacerbated if anonymous sessions are able to create, modify, or delete resources within any shares. If an anonymously accessible share contains sensitive information, such as credentials, keys, PII, etc., this may lead to further system compromise as well as an increased chance of data exposure.  While all modern versions of Windows ship with these settings disabled by default, a problematic group in Active Directory environments can bypass Group Policy restrictions set for anonymous users. This group is titled **Pre-Windows 2000 Compatible Access** and its description reads “A backward compatibility group which allows read access on all users and groups in the domain.” |
| **Affected Host** | DC01.TIMELAPSE.HTB |
| **Remediation** | Check you Group Policy settings to see if the **BUILTIN\Pre-Windows 2000 Compatible Access** group is enabled and has any network access permissions. If so, remove **Anonymous** and **Everyone** from the **PreWin2k Compatible Access** group on your Domain Controller.  It is also recommended to remove the **Access this computer from the network** permission from this group as well. For Domain Controllersthe best practice is to grant this right only to authenticated users, enterprise domain controllers, and administrators.  If completely disabling Anonymous User access isn't possible in your environment, an alternate solution would be to implement explicit access controls and restricted permissions over what resources Anonymous users can access.  **We suggest testing any Group Policy modifications in a seperate environment prior to pushing them to your Domain Controller. This will help prevent any unwanted loss of availability to resources on the network. We also recommend creating a backup of current Group Policy settings to allow for easy rollback should any problems occur.** |
| **External References** | [MITRE ATT&CK T1135](https://attack.mitre.org/techniques/T1135/)  [Pre-Win2kGroup](https://www.semperis.com/blog/security-risks-pre-windows-2000-compatibility-windows-2022/)  [Microsoft: Restrict Anonymous](https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-do-not-allow-anonymous-enumeration-of-sam-accounts-and-shares) |



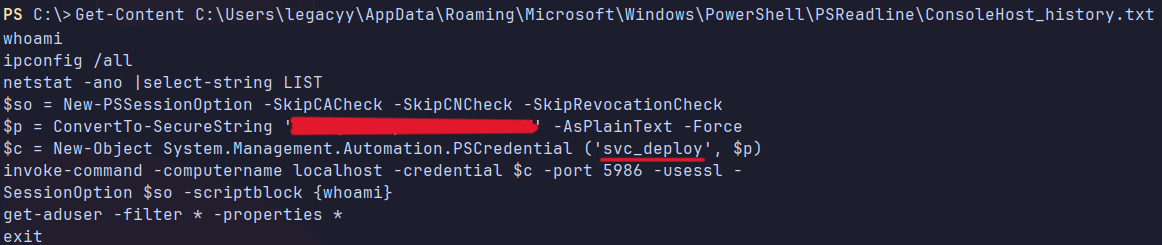




|  |  |
| --- | --- |
| **Insecure Storage of Cryptographic Keys in a Public Share** | |
| **CWE** | [CWE-922](https://cwe.mitre.org/data/definitions/922.html) |
| **CVSS v3.1 Score** | 6.5 |
| **CVSS v3.1 Vector** | AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N |
| **Description** | A PFX file was discovered in a public network share accessible by anonymous users. While the PFX file was password protected and stored in an encrypted zip archive, both layers of protection were easily bypassed using a publicly available wordlist and common hash cracking tools.  A common PKI object in Windows environments are Personal Information Exchange (PFX) files, which bundle an identity's certificate and private key into one file. This key pair may be utilized for many different purposes, including but not limited to, software signing, nonrepudiation, authentication, and encryption. It is essential to implement secure storage and management practices for these keys to help mitigate the chances of them falling into the wrong hands. |
| **Impact** | Public Key Infrastruture (PKI) is essential in providing trustworthy and secure communication over a network. PKI can be the deciding factor between a threat  If an attacker were to obtain a user's public/private key pair, the attacker could potentially use them to gain access to restricted resources, modify the integrity of transmitted data, as well as expose confidential conversations encrypted with that key. |
| **Affected Host** | DC01.TIMELAPSE.HTB |
| **Remediation** | Revoke usage of the compromised Certificate and generate new keys.  Remove the PFX file from the public share.  Implement a keystore or other key management solution that supports strong encryption.  If not using a keystore, password-protect your keys with strong, high complexity passwords.  Avoid using password protected archives as a secure storage solution for Cryptographic Keys.  If an added layer of encryption is necessary, encrypt the password-protected key using a secure cipher-suite with a utility like openssl. |
| **External References** | [Microsoft: PFX Files](https://learn.microsoft.com/en-us/windows-hardware/drivers/install/personal-information-exchange---pfx--files)  [Microsoft: Revoke Certificate](https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-cersod/c2144dad-a72b-4829-b9de-b5944b697d9e) |



|  |  |
| --- | --- |
| **Cleartext Password in PowerShell History** | |
| **CWE** | [CWE-313](https://cwe.mitre.org/data/definitions/313.html) |
| **CVSS v3.1 Score** | 6.3 - Medium |
| **CVSS v3.1 Vector** | AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L |
| **Description** | In Windows environments, PowerShell keeps a cumulative record of each user containing all the commands they have executed in each session. If a user supplies a cleartext credential in any PowerShell commands, the unencrypted credential will be stored in the PowerShell history file as well.  Access to the history file is limited to the user it pertains too, minus certain higher-privilege accounts. However, if an adversary is able to remotely impersonate a user or gain access to the account locally, the attacker would then have full permission to display the contents of the victim's PowerShell history. |
| **Impact** | The sensitive information could be read by attackers with access to the file, or with physical or administrator access to the raw disk. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information |
| **Affected Host** | DC01.TIMELAPSE.HTB |
| **Remediation** | Avoid the usage of plain text credentials, or other sensitive data, in scripts and commands. These values should be referenced using Secure Strings instead by using the command below.  *$Secure = Read-Host -AsSecureString*  Additionally, you may choose to purge the PowerShell History log or disable it completely with the following commands:  Clear PowerShell History:  *del (Get-PSReadlineOption).HistorySavePath*  Disable PowerShell History:  *Set-PSReadlineOption –HistorySaveStyle SaveNothing*  **Purging or disabling the console history may reduce visibility and accountability in your environment. These actions should only be used in specific scenarios and only after the associated risks have been taken into account.** |
| **External References** | [MITRE ATT&CK T1](https://attack.mitre.org/techniques/T1005/)005  [Microsoft: ConvertTo-SecureString](https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.security/convertto-securestring?view=powershell-7.3) |



|  |  |
| --- | --- |
| **Deprecated TLSv1.0 and TLSv1.1 Protocol Detection** | |
| **CWE** | [CWE-327](https://cwe.mitre.org/data/definitions/327.html) |
| **CVSS v3.1 Score** | 5.9- Medium |
| **CVSS v3.1 Vector** | AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N |
| **Description** | It was discovered that this system still accepts connections encrypted with the deprecated TLSv1.0 and TLSv1.1 protocols. |
| **Impact** | An adversary may be able to abuse known cryptographic flaws in order to eavesdrop on client/server connections. This may result in the attacker intercepting sensitive data being transmitted over the secured channel.  Additionally, both protocols are end-of-life and are no longer receiving any security patches or updates. This makes it very difficult to prevent against new vulnerabilities discovered within these protocols. |
| **Affected Host** | DC01.TIMELAPSE.HTB |
| **Remediation** | Remove TLS 1.0/1.1 dependencies in their environments and disable TLS 1.0 and 1.1 at the operating system level where possible. |
| **External References** | [CAPEC-97](https://capec.mitre.org/data/definitions/97.html)  [Microsoft: Disable Legacy TLS](https://learn.microsoft.com/en-us/security/engineering/disable-legacy-tls) |

Testing SSL server timelapse.htb on port 5986 using SNI name timelapse.htb

SSLv2 disabled

SSLv3 disabled

TLSv1.0 enabled

TLSv1.1 enabled

TLSv1.2 enabled

TLSv1.3 disabled

|  |  |
| --- | --- |
| **Expired SSL/TLS Certificate** | |
| **CWE** | [CWE-298](https://cwe.mitre.org/data/definitions/298) |
| **CVSS v3.1 Score** | 5.3 - Medium |
| **CVSS v3.1 Vector** | AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:L/A:N |
| **Description** | The remote server's SSL/TLS certificate has already expired. |
| **Impact** | Certificates act as a way for clients and servers to display their identity to each other, prove ownership of resources, and more. When a certificate becomes invalid, so does the digital identity of the entity who owns it. With no way to verify the identity of the server, the integrity and confidentiality of communications is lost. R  Availability of resources may also be impacted in situations where client devices perform certificate validation checks before connecting to a server. Accessing resources on a server containing an expired certificate may recquire additional interaction from the client to allow the insecure request. If the client is unable to bypass these validation checks, they may not be able to access the server at all. |
| **Affected Host** | DC01.TIMELAPSE.HTB |
| **Remediation** | Purchase or generate a new SSL/TLS certificate to replace the existing one.  Implement a PKI Management Policy to ensure certificates and keys are properly managed. |

Signature Algorithm: sha256WithRSAEncryption

RSA Key Strength: 2048

Subject: dc01.timelapse.htb

Issuer: dc01.timelapse.htb

Not valid before: Oct 25 14:05:29 2021 GMT

Not valid after: Oct 25 14:25:29 2022 GMT

**<Work in Progress>**

**Appendices**

**Appendix A – Qualitative Severity Ranking Methodology**

Probability x Impact = Severity

High

Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited

Medium

Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Low

Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Security Strengths

Strong Password Complexity

SMB v1 Disabled

SMB Signing Enabled

User Permissions

Antivirus Solution Enabled on Endpoint (Windows Defender)

PS C:\Users\Administrator\Documents> Get-MpThreatDetection

ActionSuccess : True

AdditionalActionsBitMask : 0

AMProductVersion : 4.18.2202.4

CleaningActionID : 2

CurrentThreatExecutionStatusID : 1

DetectionID : {15267089-211C-40CC-B5D1-7EBC1796CC48}

DetectionSourceTypeID : 3

DomainUser : TIMELAPSE\Administrator

InitialDetectionTime : 9/30/2023 3:57:17 AM

LastThreatStatusChangeTime : 9/30/2023 3:57:30 AM

ProcessName : C:\Windows\System32\wsmprovhost.exe

RemediationTime : 9/30/2023 3:57:30 AM

Resources : {file:\_C:\Users\Administrator\Documents\seat.exe}

ThreatID : 2147755226

ThreatStatusErrorCode : 0

ThreatStatusID : 3

PSComputerName :

ActionSuccess : True

AdditionalActionsBitMask : 0

AMProductVersion : 4.18.2202.4

CleaningActionID : 2

CurrentThreatExecutionStatusID : 1

DetectionID : {66EB472E-B580-4A66-884D-F10FFEDEC078}

DetectionSourceTypeID : 10

DomainUser : TIMELAPSE\Administrator

InitialDetectionTime : 9/30/2023 3:29:35 AM

LastThreatStatusChangeTime : 9/30/2023 3:29:51 AM

ProcessName : C:\Windows\System32\wsmprovhost.exe

RemediationTime : 9/30/2023 3:29:51 AM

Resources : {amsi:\_\Device\HarddiskVolume4\Windows\System32\wsmprovhost.exe}

ThreatID : 2147755226

ThreatStatusErrorCode : 0

ThreatStatusID : 3

SYSTEMINFO

Host Name: DC01

OS Name: Microsoft Windows Server 2019 Standard

OS Version: 10.0.17763 N/A Build 17763

OS Manufacturer: Microsoft Corporation

OS Configuration: Primary Domain Controller

OS Build Type: Multiprocessor Free

Registered Owner: Windows User

Registered Organization:

Product ID: 00429-00521-62775-AA158

Original Install Date: 10/23/2021, 10:27:45 AM

System Boot Time: 9/28/2023, 9:12:30 AM

System Manufacturer: VMware, Inc.

System Model: VMware7,1

System Type: x64-based PC

Processor(s): 1 Processor(s) Installed.

[01]: Intel64 Family 6 Model 85 Stepping 7 GenuineIntel ~2295 Mhz

BIOS Version: VMware, Inc. VMW71.00V.16707776.B64.2008070230, 8/7/2020

Windows Directory: C:\Windows

System Directory: C:\Windows\system32

Boot Device: \Device\HarddiskVolume3

System Locale: en-us;English (United States)

Input Locale: en-us;English (United States)

Time Zone: (UTC-08:00) Pacific Time (US & Canada)

Total Physical Memory: 4,095 MB

Available Physical Memory: 1,832 MB

Virtual Memory: Max Size: 5,503 MB

Virtual Memory: Available: 3,492 MB

Virtual Memory: In Use: 2,011 MB

Page File Location(s): C:\pagefile.sys

Domain: timelapse.htb

Logon Server: N/A

Hotfix(s): 6 Hotfix(s) Installed.

[01]: KB5009472

[02]: KB4512577

[03]: KB4535680

[04]: KB4589208

[05]: KB5011503

[06]: KB5009642

Network Card(s): 1 NIC(s) Installed.

[01]: vmxnet3 Ethernet Adapter

Connection Name: Ethernet0

DHCP Enabled: No

IP address(es)

[01]: 10.10.11.152

[02]: fe80::3548:f3f6:1ddc:849f

[03]: dead:beef::3548:f3f6:1ddc:849f

[04]: dead:beef::1cb

Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

By the end of the engagement, a **Stop the Xploit** analyst was able to successfully gain access to the target, **DC01.timelapse.htb**, and eventually escalate their privileges to the local administrator account. This resulted in **Stop the Xploit** obtaining virtually full control over the entire **timelapse.htb** domain. With this level of access, an adversary would be unconstrained in their actions, severely impacting the confidentiality, integrity, and availability of any assets associated with the domain.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Probability** | | |  |  |  |
| Low | Medium | High |  |  |  |
| **Impact** | High | Medium  Risk | High  Risk | Critical  Risk |  |  |  |
| Medium | Medium  Risk | Medium  Risk | High  Risk |  |  |  |
| Low | Info  Risk | Low  Risk | Medium  Risk |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | **CVSS v3.1** | |
|  |  |  |  |  |  | **Rating** | **Score** |
|  |  |  |  |  |  | None | 0 |
|  |  |  |  |  |  | Low | 0.1 - 3.9 |
|  |  |  |  |  |  | Medium | 4.0 - 6.9 |
|  |  |  |  |  |  | High | 7.0 - 8.9 |
|  |  |  |  |  |  | Critical | 9.0 - 10.0 |