

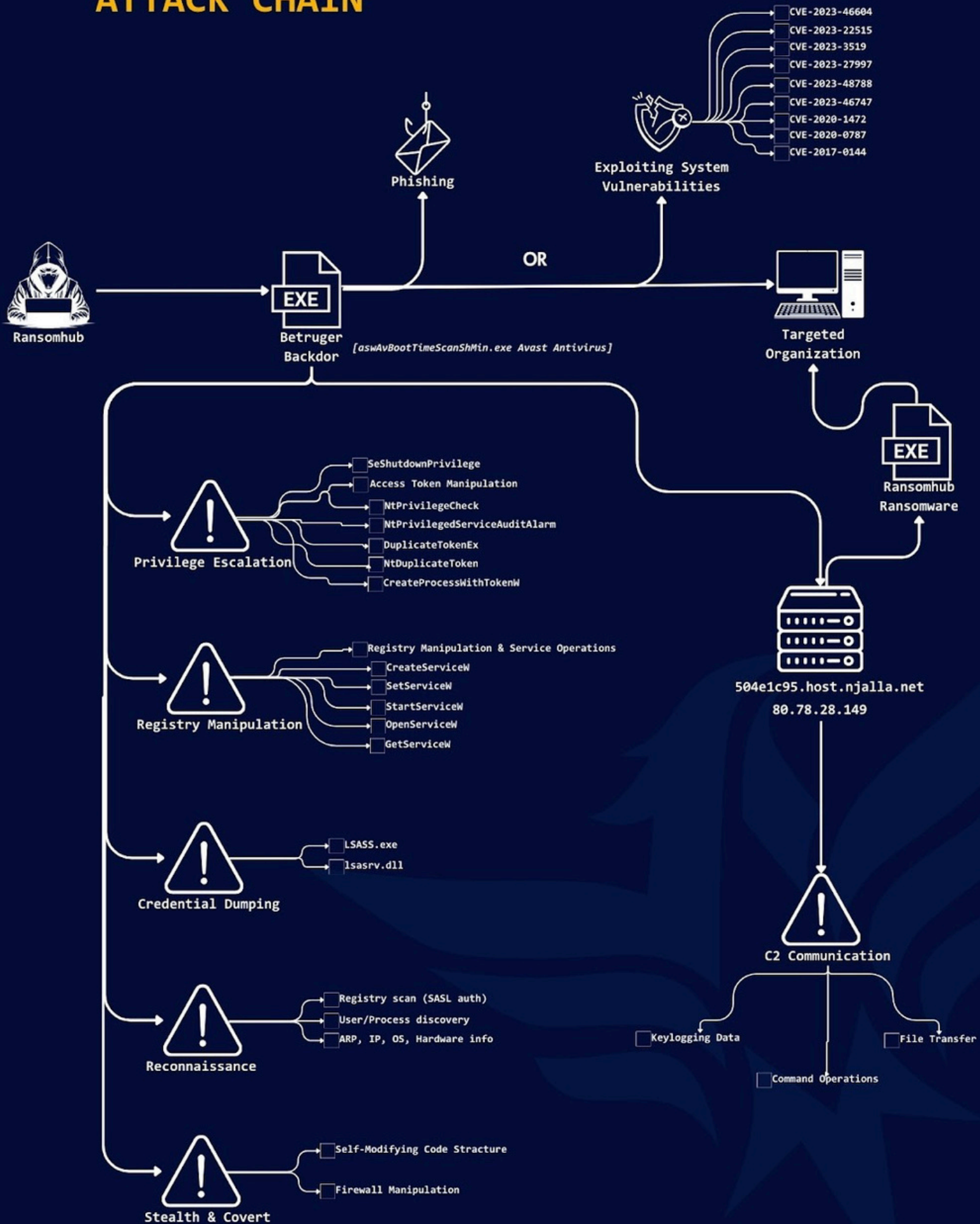


ThreatMon
Under Cyber Wings



RANSOMHUB GROUP & NEW BETRUGER BACKDOOR

ATTACK CHAIN



DIAMOND MODEL

Ransomhub Ransomware As a Service (RaaS)
Financially Motivated
Dark Web Presence

Betruger Backdoor
(Rust-based)
Credential Dumping
Keylogging
Self-modifying code
Privilege escalation
File exfiltration
Command and Control
(C2) Communication
Possible Ransomware
Infection
Masquerading
Multi-Functionality
Firewall Manipulation
Registry & Service
Manipulation
Discovery



504e1c95.host.njalla.net

80.78.28.149

WSS over TLS/SSL
communication

ransomxifxwc5eteopdobyonjctkxx
vap77yqifu2emfbecgbqdw6qd.onion

4D598799696AD5399FABF7D40C4D1BE9F05D74
CFB311047D7391AC0BF64BED47B56EEE66A528

Primary: USA (53.06%)

Secondary: Canada (14.29%)

More Countries - Global Attacks

Technology (20.83%)

Manufacturing (20.83%)

Healthcare (20.83%)

More Sectors - Global Attacks



About Ransomhub Group

RansomHub

[Home](#) / [About](#) / [Contact](#) /

 4D 19h 7m 25s Visits: 580 Data Size: 30GB Last View: 03-22 16:50:51 2025-03-20 08:44:50	 5D 19h 7m 25s Visits: 737 Data Size: 90 GB Last View: 03-22 16:49:30 2025-03-21 16:59:22	 5D 19h 7m 25s Visits: 608 Data Size: 1TB Last View: 03-22 16:52:13 2025-03-21 16:57:50
 5D 19h 7m 25s Visits: 592 Data Size: 61 GB Last View: 03-22 16:49:24 opdobyronjctkooxap77yqifu2emfbecgbqdw6qd.onion	 5D 19h 7m 25s Visits: 606 Data Size: 66 GB Last View: 03-22 16:49:24	 5D 19h 7m 25s Visits: 588 Data Size: 4 GB Last View: 03-22 16:49:24

Ransomhub Official Dark Web Website

RansomHub is a **ransomware group** that targets organizations **worldwide**, encrypting their data and demanding payment for decryption. Operating as a **Ransomware-as-a-Service (RaaS)** platform, the group collaborates with affiliates who carry out attacks using RansomHub's tools and infrastructure. In addition to encrypting data, RansomHub threatens to leak stolen information if victims refuse to pay, increasing pressure on targeted organizations.

Like most ransomware groups, RansomHub is **financially motivated**, seeking to maximize its profits by extorting victims. The group often demands payments in **cryptocurrency** to make transactions harder to trace, and it carefully selects targets that are likely to pay large sums to **recover their data**. By compromising **networks** and **exfiltrating** sensitive information, RansomHub increases its leverage, making it more difficult for organizations to refuse their demands.

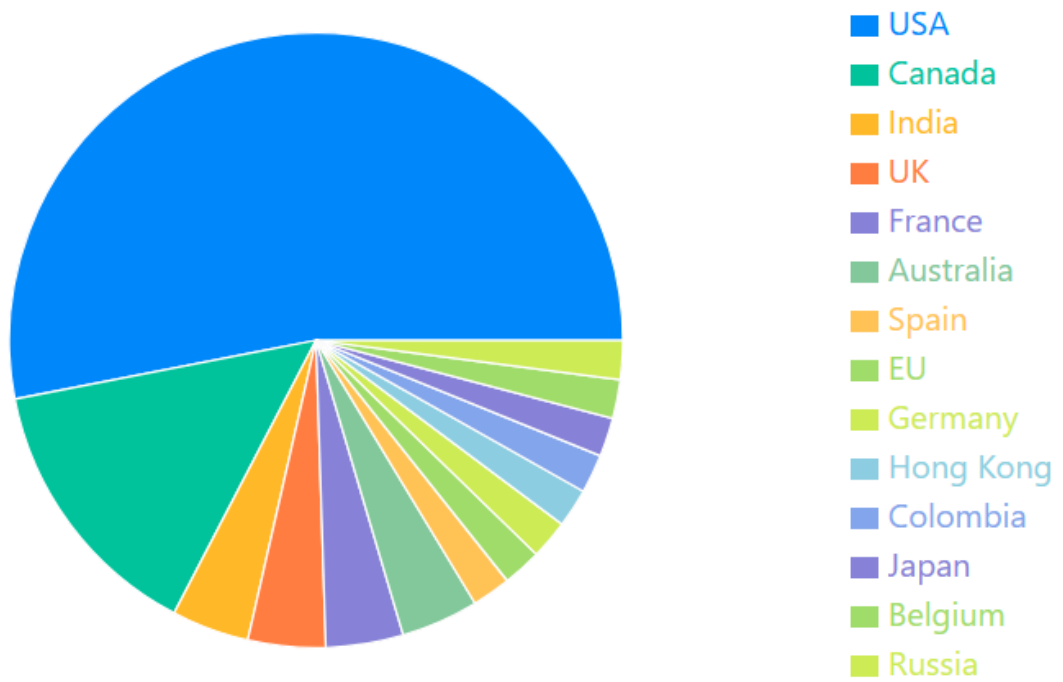
The group employs various techniques to infiltrate systems, including **exploiting security vulnerabilities**, **phishing attacks**, and abusing **remote access tools**. Their targets often include **large corporations**, **government agencies**, and **critical infrastructure providers**, as these entities are more likely to suffer significant operational and **financial damage** if their data is compromised.



Ransomhub Threat Analysis / 2025

Targeted Country Distribution / March 2025

Country Distribution (Pie Chart)



Country Distribution Graph

Based on the country information, the United States is by far the most targeted country (53.06%), followed by Canada (14.29%). Together, these North American countries account for over 67% of all attacks..

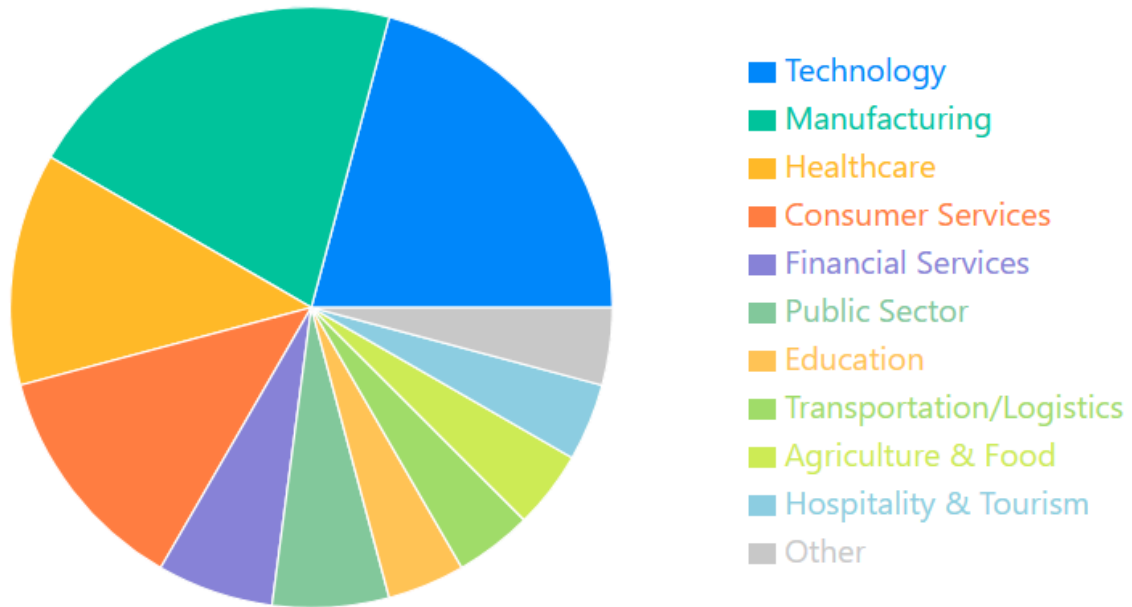
Key Findings - Country Analysis

Findings & Analysis
<ul style="list-style-type: none">North American Focus: USA and Canada are the primary targets (67.35% combined)
<ul style="list-style-type: none">Global Reach: Attacks span at least 14 different countries across multiple continents
<ul style="list-style-type: none">Secondary Markets: India, UK, France, and Australia each appear twice in the data
<ul style="list-style-type: none">English-Speaking Countries: A notable preference for targeting English-speaking nations



Targeted Sectoral Distribution / March 2025

Sector Distribution (Pie Chart)



Sectoral Distribution Graph

Based on the detailed sector information, Technology and Manufacturing are tied as the most targeted sectors (20.83% each), followed by Healthcare and Consumer Services (12.50% each).

Key Findings - Sector Analysis

Findings & Analysis
<ul style="list-style-type: none">• Technology & Manufacturing: These sectors are the most targeted (10 victims each)
<ul style="list-style-type: none">• Healthcare & Consumer Services: Second most targeted sectors (6 victims each)
<ul style="list-style-type: none">• Financial Services & Public Sector: These critical infrastructure sectors represent the third tier of targets
<ul style="list-style-type: none">• Diverse Targeting: Ransomhub attacks span at least 12 different sectors, showing broad targeting
<ul style="list-style-type: none">• Critical Services: Many targeted organizations provide essential services or possess sensitive data



Tools Used - Ransomhub

Source: ransomware.live

Tool Name	Category	Description
AnyDesk	RMM Tools	Remote desktop software that allows access and control of computers from a distance.
Atera	RMM Tools	All-in-one RMM and PSA platform for IT professionals to monitor, manage, and support networks remotely.
N-Able	RMM Tools	Comprehensive RMM solution that provides network monitoring, patch management, and remote support capabilities.
ScreenConnect	RMM Tools	Remote support software (now called ConnectWise Control) that enables technicians to connect to and control remote devices.
Splashtop	RMM Tools	Remote desktop solution that allows users to access and control computers from various devices.
Angry IP Scanner	Discovery	Fast and lightweight network scanner that pings IP addresses and ports to find active hosts.
Nmap	Discovery	Powerful network discovery and security auditing tool for scanning networks and identifying running services.
Softperfect Netscan	Discovery	Network scanner that can discover devices, scan ports, and map network resources.



Metasploit	Offsec	Advanced open-source framework for developing, testing, and executing exploits against remote targets.
Cobalt Strike	Offsec	Commercial penetration testing tool that enables adversary simulations and red team operations.
CrackMapExec	Offsec	Post-exploitation tool designed to assess and exploit Windows networks.
Sliver	Offsec	Cross-platform adversary emulation/red team framework.
Kerbrute	Offsec	Tool designed to perform Kerbero's pre-auth bruteforcing to enumerate valid Active Directory accounts.
Impacket	Offsec	Collection of Python classes for working with network protocols, used for network penetration testing.
PSCP	Exfiltration	Command-line tool that securely transfers files between computers using SSH.
RClone	Exfiltration	Command-line program for managing files on cloud storage, which can synchronize, copy, and transfer data.
WinScp	Exfiltration	Free SFTP, SCP, Amazon S3, WebDAV, and FTP client for Windows, used for secure file transfer.
BITSAAdmin	LOLBAS	Command-line tool that creates and monitors download or upload jobs using the Background Intelligent Transfer Service.
Psexec	LOLBAS	Lightweight tool that lets you execute processes on remote systems without manually installing client software.
Mimikatz	Credential Theft	Post-exploitation tool that can extract plaintext passwords, hash, PIN codes and kerberos tickets from memory.



Vulnerabilities Used - Ransomhub

VENDOR	PRODUCT	CVE	SEVERITY	SOURCE
Apache	ActiveMQ	CVE-2023-46604	Critical	cisa.gov, ransomware.live
Atlassian	Confluence Data Center & Server	CVE-2023-22515	Critical	cisa.gov, ransomware.live
Citrix	NetScaler ADC & Gateway	CVE-2023-3519	Critical	cisa.gov, ransomware.live
Fortinet	FortiOS SSL-VPN & FortiProxy	CVE-2023-27997	Critical	cisa.gov, ransomware.live
Fortinet	FortiClientEMS	CVE-2023-48788	Critical	cisa.gov, ransomware.live
F5	BIG-IP	CVE-2023-46747	Critical	cisa.gov, ransomware.live
Windows	NetLogon	CVE-2020-1472	Medium	cisa.gov, ransomware.live
Windows	BITS	CVE-2020-0787	High	cisa.gov, ransomware.live
Windows	SMBv1	CVE-2017-0144	High	cisa.gov, ransomware.live



TTP List - Ransomhub

Source: *ransomware.live*

Execution (TA0002)	Defense Evasion (TA0005)	Lateral Movement (TA0008)	Impact (TA0040)
<p>Windows Management Instrumentation (T1047) The ransomware deletes shadow copies using the WMIC.exe utility.</p>	<p>Indicator Removal: Clear Windows Event Logs (T1070.001) The ransomware clears the victim machine's application, system, and security event logs using the wevtutil.exe utility.</p>	<p>Lateral Tool Transfer (T1570) Affiliates were identified using: psexec.exe, PsExec.exe, and smbexec.exe for lateral movement.</p>	<p>Service Stop (T1489) The Windows IIS service stop command is executed using iisreset.exe. Allows for encryption of web applications hosted on IIS servers as files linked to these applications are typically locked while IIS is running.</p>
<p>Command and Scripting Interpreter: Windows Command Shell (T1059.003) The ransomware utilizes cmd.exe to execute various Windows utilities to implement various other techniques.</p>	<p>Impair Defenses: Disable or Modify Tools (T1562) Threat actors use files such as: STONESTOP and POORTRY to load drivers for the purpose of disabling and deleting AV files.</p>		<p>Inhibit System Recovery (T1490) The ransomware deletes system shadow copies to inhibit system recovery.</p>
			<p>Service Stop (T1489) The Windows IIS service stop command is executed using iisreset.exe. Allows for encryption of web applications hosted on IIS servers as files linked to these applications are typically locked while IIS is running.</p>

Active Website - Ransomhub

Onion Link
ransomxifxwc5eteopdobynonjctkxxvap77yqifu2emfbecgbqdw6qd.onion



Contact Information - Ransomhub

Tox ID
4D598799696AD5399FABF7D40C4D1BE9F05D74CFB311047D7391AC0BF64BED47B56EEE66A528

Active Malicious URLs & Domains - Ransomhub / 2025

Download the Full List:

<https://github.com/ThreatMon/ThreatMon-Reports-IOC>

Malicious URLs - Ransomhub in 2025	Malicious Domains - Ransomhub in 2025
URL hxxp[:]//40031[.]co	Domain 40031[.]co
URL hxxp[:]//12301230[.]co	Domain 12301230[.]co
URL hxxp[:]//samuelelena[.]co	Domain samuelelena[.]co
URL hxxp[:]//504e1c95.host[.]njalla{.}net	Domain 504e1c95.host[.]njalla{.}net

Active Malicious IPs - Ransomhub / 2025

Download the Full List:

<https://github.com/ThreatMon/ThreatMon-Reports-IOC>

Malicious URLs - Ransomhub in 2025	
IPv4	34{.}[132]{.}[102[.]6
IPv4	34[.]136[.]111[.]81
IPv4	80[.]178[.]128[.]149
IPv4	5[.]18[.]163[.]178



Active Malicious Hash List - Ransomhub / 2025

Download the Full List:

<https://github.com/ThreatMon/ThreatMon-Reports-IOC>

Malicious Sha256 - Ransomhub in 2025	
Sha256	ae35a3ee27cb81230a3f546253641bece5f4f6b72490e26fd3d019fbc4b8ec1
Sha256	83654c500c68418142e43b31ebbec040d9d36cfbbe08c7b9b3dc90fab414801a
Sha256	ae35a3ee27cb81230a3f546253641bece5f4f6b72490e26fd3d019fbc4b8ec1
Sha256	f95e46257efbdc06b1e3297420257aca75bc8e9e82f4ffa97430370729be01f3
Sha256	ec77a2d8ba5409d9dbf4d36a4ce511687f66e3d5cc3db022614ba5fd1f3489ba
Sha256	0d886ec0eeb7e0197591bf69a6090ba7d9a26ae1b1bb6e571a445c520952a84e
Sha256	ea9f0bd64a3ef44fe80ce1a25c387b562a6b87c4d202f24953c3d9204386cf00
Sha256	36e5be9ed3ec960b40b5a9b07ba8e15d4d24ca6cd51607df21ac08cda55a5a8e
Sha256	490a7515445c2ebf8dfbe3791383ac050cd7fca5dcf0c4539f276ad21aa1afe7
Sha256	595cd80f8c84bc443eff619add01b86b8839097621cdd148f30e7e2214f2c8cb
Sha256	104b22a45e4166a5473c9db924394e1fe681ef374970ed112edd089c4c8b83f2
Sha256	6bf404c84f05a15c62b1108336ef1b56dd9b47283deeea6412074747bf6b12cb
Sha256	94f76f44c9e0e20c0041234164eff18658ef1b961e5d1d40e9f2a2e967a059f5
Sha256	80a1ce8db3c3c02080b844418f344e386f77622d02a10b5a5f23681580ce1437
Sha256	3e54451622d485b9e0f5d953109f537b4e1dd119446f32f8fb9c3aca4cc3ea47
Sha256	e654ef69635ab6a2c569b3f8059b06aee4bce937afb275ad4ec77c0e4a712f23



Sha256	50d4a24a25be1e16b7805c6aff4436bae13623c5e26658871863875b2960771d
Sha256	2f3d82f7f8bd9ff2f145f9927be1ab16f8d7d61400083930e36b6b9ac5bbe2ad
Sha256	7114288232e469ff368418005049cf9653fe5c1cdcfcd63d668c558b0a3470f2
Sha256	7539bd88d9bb42d280673b573fc0f5783f32db559c564b95ae33d720d9034f5a
Sha256	34e479181419efd0c00266bef0210f267beaa92116e18f33854ca420f65e2087
Sha256	342b7b89082431c1ba088315c5ee81e89a94e36663f2ab8cfc27e17f7853ca2b
Sha256	f6663774f9e46cb408c6865f725b27dcb314efbf5c9f3191484caeed32ead66f
Sha256	8f59b4f0f53031c555ef7b2738d3a94ed73568504e6c07aa1f3fa3f1fd786de7

About Betruger Backdoor

BETRUGER BACKDOOR: THREAT PROFILE

TECHNICAL ANALYSIS - MARCH 2025

Operated by: RansomHub | Attack Method: Ransomware-as-a-Service

FILE PROPERTIES	TECHNICAL DETAILS	INDICATORS OF COMPROMISE
<ul style="list-style-type: none"> • Size: ~5MB • Original filename: aswAvBootTimeScanShMin.exe • Alternative names: mailer.exe, turbomailer.exe • Masquerades as Avast Antivirus 	<ul style="list-style-type: none"> • Written in Rust language • Self-modifying code techniques • Anti-analysis capabilities • Username: "malcolmbetruger" • WSS protocol with TLS/SSL 	<ul style="list-style-type: none"> • C2 Domain: 504e1c95.host.njalla.net • C2 IP Address: 80.78.28.149 • SHA256: ae7c31d4547dd293ba3fd3982b715c65...

PRIMARY CAPABILITIES

<p>Credential Theft:</p> <ul style="list-style-type: none"> • LSASS process targeting • Keylogging capabilities <p>System Manipulation:</p> <ul style="list-style-type: none"> • Firewall rule manipulation 	<p>Reconnaissance:</p> <ul style="list-style-type: none"> • System information gathering • Network discovery <p>Persistence:</p> <ul style="list-style-type: none"> • Windows service installation 	<p>Data Exfiltration:</p> <ul style="list-style-type: none"> • File transfer functionality • Encrypted C2 communication <p>Ransomware Preparation:</p> <ul style="list-style-type: none"> • RSA key generation
--	---	---



"**Betruger**" is a multi-functional **backdoor** named by security researchers and identified as being used by a **RansomHub** member. It is primarily deployed before **ransomware infections**, facilitating ransomware attacks by establishing **initial access** and **gathering critical system information**. Written in **Rust**, it features **self-modifying code**, making detection and analysis more challenging. It targets **LSASS** for **credential theft**, includes **keylogging capabilities**, and performs network reconnaissance to map potential attack vectors. Betruger also communicates with its **command-and-control (C2)** server using encrypted channels, ensuring stealthy data exfiltration.

Additionally, the malware masquerades as **Avast Antivirus**, able to **manipulate the firewall rules** to maintain access, and leverages **privilege escalation** techniques to gain higher system control. By executing these actions, it enables ransomware deployment by preparing the environment, including generating RSA keys for encryption.

Technical Analysis of the malware is covered in the below title.

Technical Malware Analysis

The screenshot shows the CFF Explorer VIII interface for the file `avast.exe`. The left pane displays the file's structure, including headers, sections, and various directories. The right pane shows two property tables.

Property	Value
File Name	C:\Users\lab\Desktop\avast.exe
File Type	Portable Executable 64
File Info	Microsoft Visual C++ 8.0 (DLL)
File Size	5.05 MB (5298176 bytes)
PE Size	5.05 MB (5298176 bytes)
Created	Friday 21 March 2025, 21:46:57
Modified	Friday 21 March 2025, 17:39:22
Accessed	Saturday 22 March 2025, 18:43:36
MD5	5675A7773F6D3224BFEFDC01745F8411
SHA-1	C0E5E4B5FCBD0A30B042E602D99A6EE81AD5D8D7

Property	Value
CompanyName	Gen Digital Inc.
LegalCopyright	Copyright © 2024 Gen Digital Inc. All rights reserved.
FileDescription	Avast Antivirus
FileVersion	24.8.9372.0
InternalName	aswAvBootTimeScanShMin
OriginalFilename	aswAvBootTimeScanShMin.exe
ProductName	Avast Antivirus
ProductVersion	24.8.9372.0



SignerCertificate	Status	Path
-----	NotSigned	avast.exe

Background processes (38)						
Avast Antivirus	0%	9.8 MB	0 MB/s	0 Mbps	Very low	Very low
COM Surrogate	0%	2.3 MB	0 MB/s	0 Mbps	Very low	Very low
COM Surrogate	0%	2.9 MB	0 MB/s	0 Mbps	Very low	Very low

Basic Characteristics & Digital Sign & Background Process

The malicious software takes up around 5.05MB of space on the disk. Its original file name is **aswAvBootTimeScanShMin.exe**, and it imitates Avast Antivirus security program. Although it does not have a digital signature, an average computer user may perceive it as the original Avast Antivirus software. In particular, the file's assembly information closely replicates that of Avast Antivirus.

Research shows that the malware can change depending on the targeted company, business, or institution. Besides imitating Avast, its assembly properties may vary according to the developed attack scenario. It is also known to attempt attacks under the names **mailer.exe** and **turbomailer.exe**, in addition to Avast.

Zaman	Olay	Yerel Adres	Uzak Adres	Protokol	Durum
12:35:36	NEW_CONNECTION	192.168.38.159:1847	80.78.28.149:443	TCP	SYN_SENT
11:01:07	CLOSED_CONNECTION	192.168.38.156:65271	80.78.28.149:443	TCP	CLOSED
11:01:07	NEW_CONNECTION	192.168.38.156:65327	80.78.28.149:443	TCP	SYN_SENT
11:00:47	NEW_CONNECTION	192.168.38.156:65271	80.78.28.149:443	TCP	SYN_SENT
10:57:49	CLOSED_CONNECTION	192.168.38.156:64813	80.78.28.149:443	TCP	CLOSED
10:57:49	NEW_CONNECTION	192.168.38.156:64864	80.78.28.149:443	TCP	SYN_SENT
10:57:28	CLOSED_CONNECTION	192.168.38.156:64757	80.78.28.149:443	TCP	CLOSED
10:57:28	NEW_CONNECTION	192.168.38.156:64813	80.78.28.149:443	TCP	SYN_SENT
10:57:08	NEW_CONNECTION	192.168.38.156:64757	80.78.28.149:443	TCP	SYN_SENT
10:53:15	CLOSED_CONNECTION	192.168.38.156:64522	80.78.28.149:443	TCP	CLOSED
10:53:15	NEW_CONNECTION	192.168.38.156:64575	80.78.28.149:443	TCP	SYN_SENT

Image of Anka Malware Analysis Tool

After the malware is executed on the system, it establishes a connection with a remote server at IP address **80.78.28.149**, which has the hostname **504e1c95.host.njalla.net**.



Hostname:	504e1c95.host.njalla.net
IP Address:	80.78.28.149

```
Listing: avast.exe - (8 addresses selected)
MOV     EAX,dword ptr [RBP + local_2c8[4]]
ADD     EAX,-0x2
TEST    EAX,0xffffffff
JZ      LAB_140054053
XOR     R8D,R8D
XOR     EDX,EDX
MOV     param_1,RBX
CALL    qword ptr [->ADVAPI32.DLL::StartServiceW]
TEST    EAX,EAX
JNS     LAB_140054053
CALL    qword ptr [->KERNEL32.DLL::GetLastError] = 00115
CMP     EAX,0x420
JNZ     LAB_140054251
LAB_140054053
CMP     byte ptr [RBP + local_2d8],0x0

Decompile: FUN_1400539c0 - (avast.exe)
241
242   if ((local_7c8 != (HKEY)0x0) && (DVar5 = RegCloseKey(local_7c8), DVar5 !=
243     SetLastError(DVar5);
244   )
245   if ((char)lstack_2f0 == '\0') {
246     pNVar8 = pNVar10;
247   }
248   if ((pNVar8 != (HKEY)0x0) && (DVar5 = RegCloseKey(pNVar8), DVar5 != 0)) {
249     SetLastError(DVar5);
250   }
251   }
252   else {
253     local_2d8[0].unused_0_i_ = '\x01';
254   }
255   if (((local_2c8.dwCurrentState - 2 & 0xffffffff) == 0) ||
256     (DVar4 = StartServiceW(hService,0,(LPCWSTR *)0x0), DVar4 != 0)) ||
257     (DVar5 = GetLastError(), DVar5 == 0x420)) {
258     if ((char)local_2d8[0].unused == '\0') {
259       local_328 = local_2d8;
260       FUN_1400547d0((longlong *)&local_328);
261     }
262     if (7 < local_68) {
58   local_7d8 = 0;
59   hSCManager = OpenSCManagerW((LPCWSTR)0x0,L"ServicesActive",1);

00007FFBADEEB760 48:83EC 48      sub rsp,48
00007FFBADEEB764 48:8B4424 78      mov rax,qword ptr ss:[rsp+78]
00007FFBADEEB769 48:894424 30      mov qword ptr ss:[rsp+30],rax
00007FFBADEEB76E 48:8B4424 70      mov rax,qword ptr ss:[rsp+70]
00007FFBADEEB773 C74424 28 01000000 mov dword ptr ss:[rsp+28],1
00007FFBADEEB77B 48:894424 20      mov qword ptr ss:[rsp+20],rax
00007FFBADEEB780 E8 0F000000 call mssock.7FFBADEEB794
00007FFBADEEB785 48:83C4 48      add rsp,48
00007FFBADEEB789 C3      ret
```

Image of Ghidra & X64dbg

Analysis confirms that the Betruger Backdoor can modify the structure of an existing service with service manipulation. While no changes were observed during execution in a virtual machine, **X64dbg** and **Ghidra** reveal its ability to alter service configurations via **SetServiceW**. This capability can be exploited for **privilege escalation, AV evasion, and ransomware deployment.**

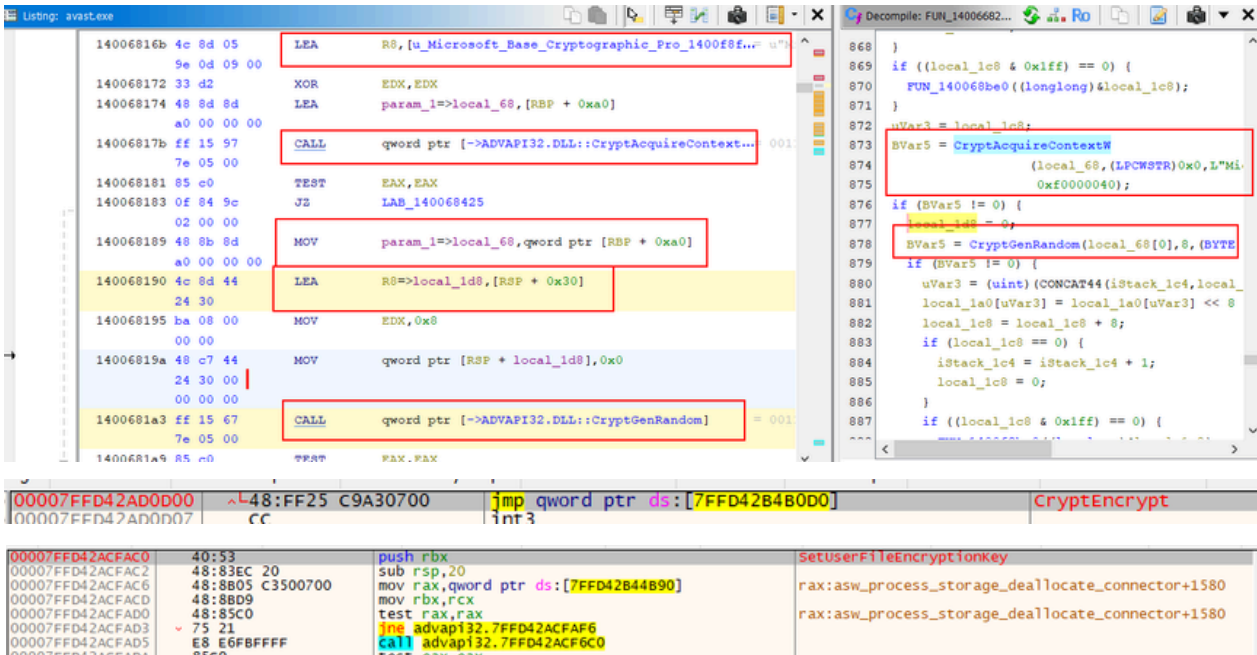
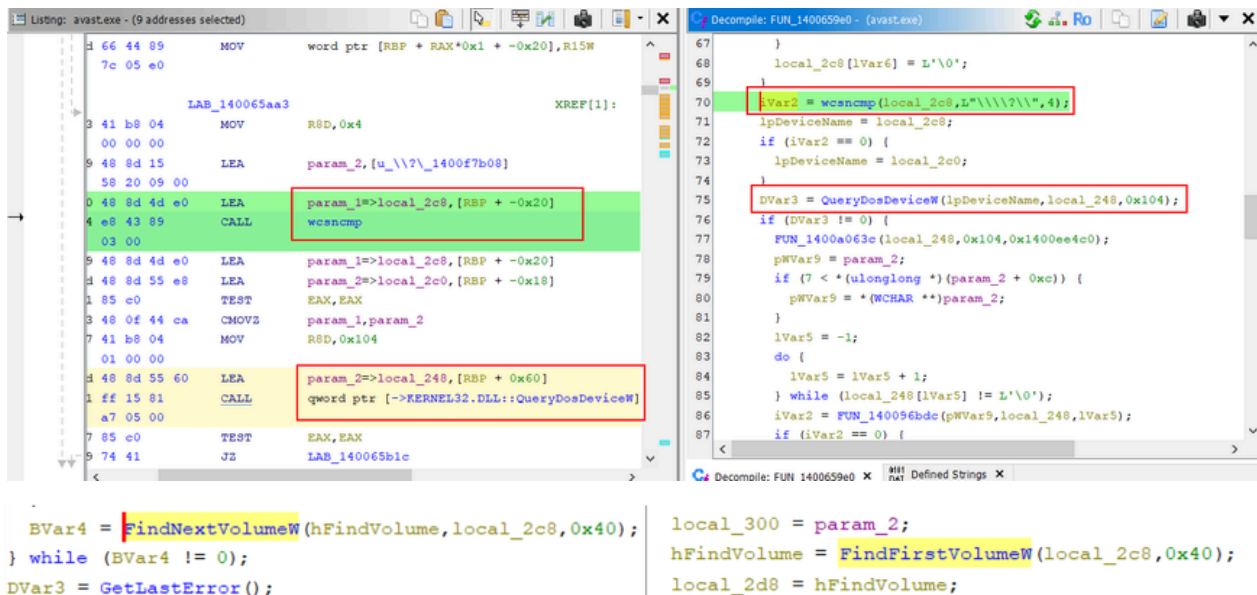


Image of Ghidra & x64dbg

During the static/dynamic analysis process, the backdoor malware exhibits behaviors that prepare for ransomware deployment. Specifically, tracking APIs such as **CryptAcquireContextW**, **CryptGenRandom**, **SetUserEncryptionKey**, and **CryptEncrypt** within the stub suggests that the backdoor may have the capability to generate an **RSA key for the ransomware**.



Images of Ghidra

Prior to the ransomware infection, another possible activity was detected. The Betruger backdoor malware was engaged in **disk discovery** and **disk access** activities on the compromised device



D43353F30	48:895C24 08	mov qword ptr ss:[rsp+8],rbx	GetAsyncKeyState
D43353F35	57	push rdi	
D41B81EE8	0F1F8400 00000000	nop dword ptr ds:[rax+rax],eax	
D41B81EF0	4C:8BD1	mov r10,rcx	NTUserGetKeyboardState
D41B81EF3	B8 76100000	mov eax,1076	
D4337F860	48:83EC 38	sub rsp,38	SetWindowsHookExA
D4337F864	C74424 20 02000000	mov dword ptr ss:[rsp+20],2	
D4337F86C	E8 BF60FEFF	call <user32.SetWindowsHookExAW>	
D4337F871	48:83C4 38	add rsp,38	
D4337F875	C3	ret	

Images of X64dbg

A **keylogging** structure has been detected within the **backdoor**. After the malware is executed on the system, it calls APIs such as **GetAsyncKeyState**, **GetKeyState**, **GetKeyboardState**, and **SetWindowsHookEx**. Specifically, before calling SetWindowsHookEx, the idHook value being set to 2 indicates the **WH_KEYBOARD** hook type, which is a clear **keylogging behavior**.

```

00007FFD40DE8470 <mswsock.TransmitFile>
mov r11, rsp
mov qword ptr ds:[r11+8], rbx
mov qword ptr ds:[r11+10], rbp
mov qword ptr ds:[r11+18], rsi
push rdi
sub rsp, 60
and qword ptr ds:[r11-28], 0
lea rax, qword ptr ds:[r11-18]
and qword ptr ds:[r11-30], 0
mov edi, r9d
mov qword ptr ds:[r11-38], rax
mov esi, r8d
lea rax, qword ptr ds:[r11-10]
mov dword ptr ss:[rsp+28], 8
mov rbp, rdx
mov qword ptr ds:[r11-48], rax
mov r9d, 10
lea r8, qword ptr ds:[7FFD40E166D0]
mov edx, C8000006
mov rbp, rdx
call qword ptr ds:[kwsAIoctls]
xor eax, eax
cmp eax, FFFFFFFF
jmp mswsock.7FFD40DE8479

mswsock.00007FFD40DE84D9
mov ecx, dword ptr ss:[rsp+A0]
mov r9d, edi
mov rax, qword ptr ss:[rsp+58]
mov r8d, esi
mov dword ptr ss:[rsp+30], ecx

mswsock.00007FFD40DE84D5
jmp mswsock.7FFD40DE8515

00007FFD42541760 48:8BC4 mov rax, rsp
00007FFD42541763 48:8958 08 mov qword ptr ds:[rax+8], rbx
00007FFD42541767 48:8970 10 mov qword ptr ds:[rax+10], rsi
00007FFD4254176B 48:8978 18 mov qword ptr ds:[rax+18], rdi
  
```

Images of X64dbg

It has been clearly analyzed that **file transfer operations** can take place within the **backdoor**. This situation, especially in ransomware attacks, not only involves the encryption of data but also creates the risk of **data leaks**. It is known that Ransombh can **publicly share** the data if the ransom payment is refused.

0000000000014AB8	00000000F4685703		
0000000000014ABC0	0000000000000003		
0000000000014ABC8	00000000319E5092		
0000000000014ABD0	000000001813A696		
0000000000014ABD8	0000000006EB4CFC6		
0000000000014ABE0	000000000014C848		
0000000000014ABE8	000000000261761F		return to 000000000261761F from 00000000025ABA9
0000000000014ABF0	0000000000000008		
0000000000014ABF8	000000000014C848		
0000000000014AC00	0000000000000000		
0000000000014AC08	0000000000000000		
0000000000014AC10	00000000027E6CD8		&"src/. /windows_entry.rs"
0000000000014AC18	0000000000000000		
0000000000014AC20	0000000000000000		
0000000000014AC28	0000000000000000		

sub eax, 2267981	0000000002267981	"o-6f17d22bba15001f/gif-0.12.0/src/common.rs"
add eax, 26DB900	00000000026DB900	"01f/tokio-1.39.3/src/runtime/park.rsinconsis"

Images of X64dbg



During the dynamic analysis of the malware, a reference to **src/windows_entry.rs** was observed in memory. This file manages the malware's Windows-specific entry point and system integration, indicating that **Betruger Malware** was developed in the **Rust programming language**.

```
String Address String
000000000266CC63 "1.2/src/dfa.rs/Users/malcolmbetruger/.cargo/registry/src/index.crates.io-6f17d22bba15001f/aho-cor
00000000026D342E "e64 is always valid HeaderValue/Users/malcolmbetruger/.cargo/registry/src/index.crates.io-6f17d22
```

Images of X64dbg

Memory dump strings confirm the project was developed in Rust, indicated by **.rs** files and crates.io references. Debug symbols left during compilation suggest it was built on a computer with the username "**malcolmbetruger**." While this name belongs to a game character, it may also be (not definitively) a **Ransohub** member's alias.

The screenshot displays the X64dbg interface. The assembly window shows the following instructions:

```

00007FFBF5759832 48:8D15 F7900100 lea rdx,qword ptr ds:[7FFBF5772930]
00007FFBF5759839 48:C7C1 02000080 mov rcx,FFFFFFFF80000002
00007FFBF5759840 48:FF15 81870100 call qword ptr ds:[<RegOpenKeyExW>]
00007FFBF5759847 0F1F4400 00 nop dword ptr ds:[rax+rax],eax
00007FFBF575984C 45:33ED      xor r13d,r13d
00007FFBF575984F 85C0        test eax,eax
00007FFBF5759851 0F85 8C010000 jne sspic11.7FFBF57599E3
00007FFBF5759857 6548:8B0C25 60000000 mov rcx,qword ptr ds:[60]
00007FFBF5759860 33D2        xor edx,edx
00007FFBF5759862 41:88 10040000 mov r8d,410
00007FFBF5759868 48:8B49 30    mov rcx,qword ptr ds:[rcx+30]
00007FFBF575986C 48:FF15 2D880100 call qword ptr ds:[<RT1A1locateHeap>]
00007FFBF5759873 0F1F4400 00 nop dword ptr ds:[rax+rax],eax
00007FFBF5759878 48:8BD8      mov rbx,rax
00007FFBF575987B 48:85C0      test rax,rax
00007FFBF575987E 0F84 4F010000 je sspic11.7FFBF57599D3
00007FFBF5759884 4C:8DA0 08020000 lea r12,qword ptr ds:[rax+208]
00007FFBF5759888 41:8BF5      mov esi,r13d
00007FFBF575988E 48:8B4D 60    mov rcx,qword ptr ss:[rbp+60]
00007FFBF5759892 48:8D45 48    lea rax,qword ptr ss:[rbp+48]
00007FFBF5759896 48:894424 38    mov qword ptr ss:[rsp+38],rax
00007FFBF575989B 4C:8D4D 50    lea r9,qword ptr ss:[rbp+50]
00007FFBF575989F 48:8D45 58    lea rax,qword ptr ss:[rbp+58]
00007FFBF57598A3 4C:896424 30    mov qword ptr ss:[rsp+30],r12
00007FFBF57598A8 48:894424 28    mov qword ptr ss:[rsp+28],rax
00007FFBF57598AD 4C:8BC3      mov r8,rbx
00007FFBF57598B0 8BD6        mov edx,esi
00007FFBF57598B2 4C:896C24 20    mov qword ptr ss:[rsp+20],r13
00007FFBF57598B7 C745 50 04010000 mov dword ptr ss:[rbp+50],104

```

The memory dump window shows the following string:

```

7D8
07FFBF5772930 L"system\\CurrentControlSet\\Control\\SecurityProviders\\Sas1Profiles"]=74007300790

```

Images of X64dbg

SASL is a framework that forms the authentication infrastructure for many critical protocols (**LDAP, SMTP, IMAP**), and the Betruger Backdoor directly targets this framework through the registry path;

System\CurrentControlSet\Control\SecurityProviders\saslProfiles.

It accesses the authentication configuration using the **RegOpenKeyExW** API. This indicates that the Betruger Backdoor exhibits **lateral movement** capabilities.

```
add byte ptr ds:[rax+22c0009],a1 00000000022c0009 "halasinhahasogdSogdiansogdiansogorasora_
add byte ptr ds:[rdx+rsi*4+22c0029],ch 00000000022c0029 "osorasora_sompengorasompengsoyoSoyombosoy
```

Images of X64dbg



It has been determined that the **self-modifying code** technique is used for **anti-analysis** functionality. This technique is utilized by the backdoor using **memory manipulation** on addresses **0x22C0009** and **0x22C0029**, modifying string data during runtime to generate dynamic payloads and implement encrypted command and control communication while evading detection.

```

00000000025C80EA 49:8D87 8C000000 lea rax,qword ptr ds:[r15+8C]
00000000025C80F1 48:898424 E8000000 mov qword ptr ss:[rsp+E8],rax
00000000025C80F9 41:0FB687 8C000000 movzx eax,byte ptr ds:[r15+8C]
00000000025C8101 48:8D0D 38B92000 lea rcx,qword ptr ds:[27D3A40]
00000000025C8108 48:630481 movsxd rax,dword ptr ds:[rcx+rax*4]
00000000025C810C 48:01C8 add rax,rcx
00000000025C810F - FFE0 jmp rax
00000000025C8111 49:8DBF D8020000 lea rdi,qword ptr ds:[r15+2D8]
00000000025C8118 41:0FB687 D8020000 movzx eax,byte ptr ds:[r15+2D8]
00000000025C8120 4D:8DA7 90000000 lea r12,qword ptr ds:[r15+90]
00000000025C8127 48:8D0D 2EB92000 lea rcx,qword ptr ds:[27D3A5C]
00000000025C812E 48:630481 movsxd rax,dword ptr ds:[rcx+rax*4]
00000000025C8132 48:01C8 add rax,rcx
00000000025C8135 - FFE0 jmp rax
00000000025C8137 48:897C24 78 mov qword ptr ss:[rsp+78],rdi
00000000025C813C 4D:8D87 D0020000 lea r14,qword ptr ds:[r15+2D0]
00000000025C8143 41:0FB687 D0020000 movzx eax,byte ptr ds:[r15+2D0]
00000000025C8148 49:8DBF 20010000 lea rdi,qword ptr ds:[r15+120]
00000000025C8152 48:8D0D 13B92000 lea rcx,qword ptr ds:[27D3A6C]
00000000025C8159 48:630481 movsxd rax,dword ptr ds:[rcx+rax*4]
00000000025C815D 48:01C8 add rax,rcx
00000000025C8160 - FFE0 jmp rax
00000000025C8162 48:8D0D 87E52100 lea rcx,qword ptr ds:[27E66F0]
00000000025C8166 E8 F2650800 call 264E760
00000000025C816E E9 1C0D0000 jmp 25C8E5F
00000000025C8173 4C:8BB424 30020000 mov r14,qword ptr ss:[rsp+230]
00000000025C817B 40:8AAC24 38020000 mov bpl,byte ptr ss:[rsp+238]
00000000025C8183 8B8424 39020000 mov eax,dword ptr ss:[rsp+239]
00000000025C818A 898424 00010000 mov dword ptr ss:[rsp+100],eax

```

r12:&"C:\\Users\\fab\\Desktop\\avast.exe", [r15+90]

00000000027E66F0:&"src/global_state.rs"

```

6C
8C]=[000000000014926C]=5C1CF000000004

```

```

000000000149268 0000000400000001 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149270 00000000005C1CF0 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149278 0000000000000023 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149280 00000000005C1CF0 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149288 0000000000000023 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149290 0000000000000000 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
000000000149298 0000000000000008 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
0000000001492A0 0000000000000000 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"
0000000001492A8 0000000000000023 "wss://504e1c95.host.njalla.net:443/ab_3928x86)\\Common Files"

```

Images of X64dbg

It has been determined that the **WSS** protocol is used for **Server-Client** communication. WSS is a real-time, bidirectional communication protocol and is utilized by Betruger in an encrypted manner using **TLS/SSL** for purposes such as file transfers, keylogger data transmission, and command execution.





```
4C:88DC      mov r11,rsp
48:83EC 68      sub rsp,68
48:888424 B0000000  mov rax,qword ptr ss:[rsp+80]
49:8943 F0      mov qword ptr ds:[r11-10],rax
48:888424 A8000000  mov rax,qword ptr ss:[rsp+A8]
49:8943 E8      mov qword ptr ds:[r11-18],rax
48:888424 A0000000  mov rax,qword ptr ss:[rsp+A0]
49:8943 E0      mov qword ptr ds:[r11-20],rax
48:888424 98000000  mov rax,qword ptr ss:[rsp+98]
49:8943 D8      mov qword ptr ds:[r11-28],rax
888424 90000000  mov eax,dword ptr ss:[rsp+90]
894424 38      mov dword ptr ss:[rsp+38],eax
4D:8948 C8      mov qword ptr ds:[r11-38],r9
4D:8943 C0      mov qword ptr ds:[r11-40],r8
894424 20      mov dword ptr ss:[rsp+20],edx
48:8D15 03900700  lea rdx,qword ptr ds:[7FFBF68ED648]
4C:8BCA      mov r9,rdx
4C:8BC2      mov r8,rdx
E8 0C000000  call advapi32.7FFBF687465C
```

Images of X64dbg

With the **SeShutdownPrivilege** structure, **shutdown** and **restart** privileges are obtained. Additionally, the usage of the **CreateProcessWithTokenW** API from **advapi32.dll** and attempts to access the **System32** directory are observed, which are used for **privilege escalation**.

```
rpqrt4.00007EFBE631BA38
lea rdx,qword ptr ds:[7FFBF6401820] ; 00007FFBF6401820:L"lsass.exe"
mov rcx,qword ptr ds:[rax+18] ; rcx:ZwQueryInformationThread+14
mov rax,qword ptr ds:[rcx+10] ; rcx+10:ZwOpenProcess+4
mov rcx,qword ptr ds:[rax+60] ; rcx:ZwQueryInformationThread+14
call qword ptr ds:[<_wcsicmp>]
nop dword ptr ds:[rax+rax],eax
test eax,eax
je rpqrt4.7FFBF631BAFC

rpqrt4.00007FFBF631BAFC
cmp dword ptr ds:[7FFBF641FEC0],0
je rpqrt4.7FFBF631BA5F

rpqrt4.00007FFBF631BB09
jmp rpqrt4.7FFBF638F3F4

rpqrt4.00007FFBF638F3F4
lea rcx,qword ptr ds:[7FFBF64061F8] ; rcx:ZwQueryInformationThread+14, 00007FFBF64061F8:L"lsasrv.dll"
call qword ptr ds:[<GetModuleHandleW>]
nop dword ptr ds:[rax+rax],eax
mov rbx,rax ; rbx:"LdrpInitializeProcess"
test rax,rax
je rpqrt4.7FFBF631BA5F
```

Images of X64dbg

The **Betruger Backdoor** targets the **LSASS** process. It uses **ZwOpenProcess** to access **LSASS.exe**, and loads **lsasrv.dll** via **GetModuleHandleW**.

lsasrv.dll is a key **LSASS** component that manages authentication and security policies, and provides access to credential data.

This shows that the malware attempts **credential dumping**.



```

4C:8D05 DAC81600 | lea r8,qword ptr ds:[7FFBF5CBBD50] | 00007FFBF5CBBD50:""
48:8D4D 50 | lea rcx,qword ptr ss:[rbp+50] | rcx:ZwQueryInformation
BA 04010000 | mov edx,104
48:FF15 FA540C00 | call qword ptr ds:[<swprintf_s>]
0F1F4400 00 | nop dword ptr ds:[rax+rax],eax
33D2 | xor edx,edx
44:8D43 67 | lea r8d,qword ptr ds:[rbx+67] | rbx+67:"us 0x%08lx\n"
48:8D4D C8 | lea rcx,qword ptr ss:[rbp-38] | rcx:ZwQueryInformation
E8 C573F9FF | call <JMP.&memset>

```

nginodynamioC+S/EA
nThread+14

```

nThread+14
///?\\%s\\system32\\conhost.exe --headless %s--width %hu --height %hu --signal 0x%x --server 0x%x"
nThread+14

```

Images of X64dbg

Arguments are being passed to conhost.exe as follows:

--headless %s --width %hu --height %hu --signal 0x%x --server 0x%x

Here, a console window appears to be used for executing a command, but with the **-headless** option, it runs **without being visible** to the user.

One of the most **critical** aspects of this command structure is the **--signal and --server** arguments. This structure allows for command exchange with a **remote server**, and in the case of malicious software, a **reverse shell** mechanism has been observed.

Images Firewall Operations

During the analysis process, some firewall operations associated with Windows Filtering Platform (WFP) were found. It was observed that the malware can access Windows Firewall structures and perform operations such as **adding rules, listing existing rules and deleting rules**.



```
00007FFBF4CC9D60 <iphlpapi.GetNetworkParams>
mov qword ptr ss:[rsp+8],rbx
mov qword ptr ss:[rsp+18],rbp
mov qword ptr ss:[rsp+20],rsi
push rdi
sub rsp,20
mov rdi,rdx
mov rbp,rcx
test rdx,rdx
jne iphlpapi.7FFBF4CC9D87

00007FFBF5A84240 <kernelbase.GetSystemTimeAsFileTime>
mov eax,7FFBF50014
mov rax,qword ptr ds:[rcx]
mov dword ptr ds:[rcx],eax
shr rax,20
mov dword ptr ds:[rcx+4],eax
ret

00007FFBF5AB4D90 <kernelbase.GetVersion>
push rbx
sub rsp,20
mov rax,qword ptr ds:[7FFBF5CE95A8]
xor ebx,ebx
test rax,rax
jne kernelbase.7FFBF5AB4DCE

00007FFBF5A9F4A0 <kernelbase.GetVersionExw>
mov qword ptr ss:[rsp+8],rbx
push rdi
sub rsp,20
mov eax,dword ptr ds:[rcx]
xor edi,edi
sub eax,114
mov rbx,rcx
test eax,FFFFFFFF
jne kernelbase.7FFBF5B03350

00007FFBF72BA610 <kernel32.GetComputerName>
mov r11,rsi
mov qword ptr ds:[r11+18],rbx
mov qword ptr ds:[r11+20],rsi
push rdi
sub rsp,50
mov rax,qword ptr ds:[7FFBF7352220]
xor rax,rsi
mov qword ptr ss:[rsp+48],rax
mov rsi,rdx
mov dword ptr ss:[rsp+20],10
mov rdi,rcx
lea rdx,qword ptr ds:[r11-38]
lea rcx,qword ptr ds:[r11-30]
lea rbx,qword ptr ds:[r11-30]
call <kernel32.GetComputerNameW>
test eax,eax
je kernel32.7FFBF72D37A4

00007FFBF4CCD220 <iphlpapi.GetIpForwardTable>
mov qword ptr ss:[rsp+8],rbx
push rbp
push rsi
push rdi
push r12
push r13
push r14
push r15
lea rbp,qword ptr ss:[rsp-27]
sub rsp,C0
xor r13d,r13d
mov rbx,rdx
mov qword ptr ss:[rbp-21],r13
mov r14,rcx
mov qword ptr ss:[rbp+7],r13
mov qword ptr ss:[rbp-1],r13
mov qword ptr ss:[rbp-9],r13
mov qword ptr ss:[rbp-11],r13
mov dword ptr ss:[rbp-29],r13d
mov qword ptr ss:[rbp+17],r13
mov qword ptr ss:[rbp+8],r13
test rdx,rdx
jne iphlpapi.7FFBF4CCD233

00007FFBF5A5DE50 <kernelbase.GetSystemInfo>
push rbx
sub rsp,80
mov rax,qword ptr ds:[7FFBF5CE95A8]
xor rax,rsi
mov qword ptr ss:[rsp+70],rax
xor r9d,r9d
lea rdx,qword ptr ss:[rsp+30]
mov rbx,rcx
xor ecx,ecx
xor ecx,ecx
mov rax,qword ptr ds:[rcx+40]
```

Images Multiple Discovery & Reconnaissance Operations

Also, the following findings related to discovery and reconnaissance activities have been identified:

- Obtaining hardware information from the infected device
- Retrieving extended user information
- Gathering information about system architecture and processor type
- Obtaining operating system details
- Retrieving the machine name
- Obtaining the IP address assigned to the machine
- Resolving network addresses
- Obtaining ARP table information



● Accessing the service control manager
● Listing active TCP/UDP connections
● Enumerating account privileges of active users
● Gathering information about active user accounts
● Retrieving the system time of the infected device
● Obtaining detailed information about running processes
● Determining the locations of system folders

Categorization & IOC List

Threat Group	Identified Threat Categories
Ransomhub	Backdoor Trojan
	Keylogger

You can download the IOC List from [ThreatMon Github](#)



IOC List	
DOMAIN	504e1c95[.]host[.]njalla[.]net
IPv4	80[.]78[.]28[.]149
SHA256	ae7c31d4547dd293ba3fd3982b715c65d731ee07a9c1cc402234d8705c01dfca
SHA256	b058c128c801e2ee03874e183239ff369c599f3a2324905ff73f99d16d3b1a16



Mitre & Attack Table

Tactics	ID	Name	Description
Initial Access	T5566	Phishing	Betruger backdoor can be delivered via a phishing attack
Privilege Escalation	T1134	Access Token Manipulation	Uses CreateProcessWithTokenW API from advapi32.dll for privilege escalation
	T1134.002	Access Token Manipulation: Create Process with Token	Evidence of the SeShutdownPrivilege structure used for token manipulation
	T1543.003	Create or Modify System Process: Windows Service	Able to modify and manipulate a windows service.
Defense Evasion	T1027	Obfuscated Files or Information	Malicious code is encrypted to evade detection.
Credential Access	T1003.001	OS Credential Dumping: LSASS Memory	Betruger Backdoor targets the LSASS process" and "loads lsasrv.dll via GetModuleHandleW
Lateral Movement	T1021	Remote Services	Possible to spread within the network via remote connections.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Communicates with C2 server using WSS protocol with TLS/SSL encryption
	T1573.002	Encrypted Channel: Asymmetric Cryptography	Uses encrypted communication channels for stealthy data exfiltration
	T1105	Ingress Tool Transfer	Capabilities for downloading additional payloads



Impact	T1041	Data Exfiltration	Sensitive data is stolen from the target system.
Collection	T1056.001	Input Capture: Keylogging	Contains keylogging structure using SetWindowsHookEx with WH_KEYBOARD hook type
Discovery	T1082	System Information Discovery	Obtaining hardware information, system architecture, processor type, and OS details
	T1083	File and Directory Discovery	Determining locations of system folders and disk discovery
	T1124	System Time Discovery	Retrieving system time of the infected device
	T1016	System Network Configuration Discovery	Retrieving IP addresses, MAC addresses, ARP tables, and routing tables
	T1033	System Owner/User Discovery	Gathering information about user accounts and privileges
	T1057	Process Discovery	Obtaining detailed information about running processes
	T1012	Query Registry	Accesses authentication configuration using RegOpenKeyExW API
	T1018	Remote System Discovery	Network discovery capabilities to map potential attack vectors
	Exfiltration	T1041	Exfiltration Over C2 Channel
Impact	T1486	Data Encrypted for Impact	Prepares environment for ransomware by generating RSA keys for encryption



Yara Rule

You can download the yara rule from [ThreatMon Github](#)

```
rule Betruger_Backdoor_YaraRule {
    meta:
        description = "Enhanced YARA rule for detecting Betruger Backdoor
used by Ransomhub"
        author = "Aziz Kaplan"
        email = "aziz.kaplan@threatmonit.io"
        reference = "https://threatmonit.io/"
        threat_level = 10
        severity = "critical"
        family = "Ransomhub.Betruger"
        tlp = "GREEN"
        mitre_att = "T1486, T1490, T1083, T1057, T1082"

    strings:
        $str1 = "Avast Antivirus" wide ascii
        $str2 = "avast-av" wide ascii
        $str3 = "IDI_ASWAVBOOTTIMESCANSHMIN" wide ascii
        $str4 = "AV Boot-time Scanner" wide ascii

        $str5 = "Windows Registry Editor" wide ascii
        $str6 = "lifetime_creation_monitor_holder" wide ascii
        $str7 = "/runassvc /winre" wide ascii nocase

        $op1 = {4? 8d 4c ?4 40 e8 26 18 00 00}
        $op2 = {80 7d c8 00 75 09 4? 8b cf 4? 89 4d c0}
        $op3 = {ff 15 7f b1 0b 00 85 c0 74 08 8b c8 ff 15 db b5 0b 00}
        $op4 = {ff 15 af 54 0b 00 85 c0 74 2d 4? 8d 4c ?4 60}

        $op5 = {4? 8d 84 ?4 c8 00 00 00 4? 89 44 ?4 28}
        $op6 = {ff 15 6d 2f 06 00 85 c0 0f 84 dd 00 00 00}
```



```
$op7 = {4? 8d 4d 70 4? c7 45 70 00 00 00 00 ff 15 3c 99 05 00}
$op8 = {4? 8d 55 78 ?? ?? ?? ?? ?? ff 15 26 90 05 00}

$op9 = {4? 8d 44 ?4 30 ba 08 00 00 00}
$op10 = {4? c7 44 ?4 30 00 00 00 00 ff 15 67 7e 05 00}

$op11 = {ba ff ff ff ff 4? 8b ca 4? 8b c2 4? 8b cf ff 15 cf b9 06 00}
$op12 = {ff 15 67 bb 06 00 4? 8b d8 4? 89 45 0f}

$op13 = {ff 15 74 5a 05 00 4? 8d 0d 0d ee 08 00 4? 89 4c ?4 20}

$op14 = {4? 8d 57 40 4? 8d 4d e0 ff 15 6b a8 05 00}
$op15 = {4? 8d 15 58 20 09 00 4? 8d 4d e0 e8 43 89 03 00}

$op16 = {4? 8b d0 4? 8b ce ff 15 4d 17 08 00}

    condition:
uint16(0) == 0x5A4D and
filesize > 5MB and filesize < 10MB and
(
    (all of ($op*) and all of ($str*))
    or
    (all of ($op*) and 4 of ($str*))
    or
    all of ($op*)
)
}
```



Sigma Rules

You can download the Sigma Rules from [ThreatMon Github](#)

```
title: Trojanized Avast Binary C2 Communication
id: f8a12b2e-d45f-4a84-9b4b-1e0e3e8f5a7d
status: test
description: Detects suspicious C2 communication from trojanized avast.exe binary to
malicious servers.
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
date: 2025-03-23
references:
- ThreatMon Malware R&D
logsource:
product: windows
category: network_connection
detection:
selection:
  Image|endswith: '\avast.exe'
  DestinationHostname|contains: 'njalla.net'
  Initiated: 'true'
condition: selection
falsepositives:
- Legitimate Avast security software updates
- Downloading virus definitions
level: high
```

```
title: Fake Avast Application Detected
id: 762f3a9c-5d2b-42e3-bc4f-16c7f8d94856
status: test
description: Detects execution of a malicious avast.exe application that mimics the
legitimate Avast security software.
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
date: 2025-03-23
references:
- ThreatMon Malware R&D
logsource:
product: windows
category: process_creation
detection:
```



selection:

Image|endswith: '\\avast.exe'

CurrentDirectory|contains:

- '\\Desktop\\'
- '\\Downloads\\'
- '\\Documents\\'

filter:

IntegrityLevel: 'System'

ParentImage|endswith:

- '\\Program Files\\Avast Software\\Avast\\AvastSvc.exe'
- '\\Program Files (x86)\\Avast Software\\Avast\\AvastSvc.exe'

condition: selection and not filter

falsepositives:

- User executing normal Avast installer from desktop

level: high

title: Suspicious Avast Repetitive C2 Connection Attempts

id: e02a4c10-7193-4a79-af9b-75d580b1d68c

status: test

description: Detects fake avast.exe making repetitive connection attempts to the same domain within short time intervals

author: Aziz Kaplan <aziz.kaplan@threatmonit.io>

date: 2025-03-23

references:

- ThreatMon Malware R&D

logsource:

product: windows

category: network_connection

detection:

selection:

Image|endswith: '\\avast.exe'

DestinationHostname|contains:

- 'njalla.net'

Initiated: 'true'

timeframe: 5m

condition: selection | count() > 5

falsepositives:

- Heavy update activity

level: high



title: Suspicious Avast Application Hosts File Access
id: d16b0f5e-6c2f-4d3f-ab43-76cfc7510e2c
status: test
description: Detects malicious avast.exe application reading Windows hosts file
author: Aziz Kaplan <aziz.kaplan@threatmonit.io>
date: 2025-03-23
references:
- ThreatMon Malware R&D
logsource:
product: windows
category: file_event
detection:
selection:
 Image|endswith: '\\avast.exe'
 TargetFilename: 'C:\\Windows\\System32\\drivers\\etc\\hosts'
 Access: 'ReadData'
filter:
 CurrentDirectory|contains:
 - '\\Program Files\\Avast Software\\Avast\\'
 - '\\Program Files (x86)\\Avast Software\\Avast\\'
condition: selection and not filter
falsepositives:
- Legitimate Avast software reading network configuration
level: medium



Windows Defender Custom Detection Rule

You can download the detection rule from [ThreatMon Github](#)

```
<Detections>
  <Detection>
    <Name>Betruger Backdoor Detection</Name>
    <Description>Detects Betruger backdoor malware associated with RansomHub
group</Description>
    <Severity>High</Severity>
    <FileSha256
Detect="Equals">ae7c31d4547dd293ba3fd3982b715c65d731ee07a9c1cc402234d8705c01dfca</File
Sha256>
    <FileSha256
Detect="Equals">b058c128c801e2ee03874e183239ff369c599f3a2324905ff73f99d16d3b1a16</File
Sha256>
    <FileName Detect="Contains">aswAvBootTimeScanShMin.exe</FileName>
    <FileName Detect="Contains">mailer.exe</FileName>
    <FileName Detect="Contains">turbomailer.exe</FileName>
    <NetworkConnection>
    <RemoteIP Detect="Equals">80.78.28.149</RemoteIP>
    <RemoteDomain Detect="Contains">504e1c95.host.njalla.net</RemoteDomain>
    </NetworkConnection>
  </Detection>
</Detections>
```



Mitigations - General

- Implement a rigorous vulnerability management program focusing on internet-facing systems
 - Prioritize patching of critical vulnerabilities mentioned in the report (CVE-2023-46604, CVE-2023-22515, CVE-2023-3519, etc.)
 - Block known malicious IPs and domains provided within the report
 - Implement network segmentation to limit lateral movement
 - Maintain offline backups that are disconnected from the network
 - Implement the 3-2-1 backup strategy (3 copies, 2 different media, 1 offsite)
 - Implement email scanning for malicious attachments and links
 - Train users to identify suspicious emails, especially those with attachments
 - Apply the principle of least privilege for all accounts
 - Implement multi-factor authentication across all systems
 - Deploy EDR solutions that can detect the IOCs mentioned in the report
 - Implement the provided Yara rules and Sigma rules for detection
 - Add the Windows Defender Custom Detection Rule to security solutions
 - Block execution of binaries from unusual locations (Desktop, Downloads, etc.)
 - Develop and test an incident response plan specific to ransomware attacks
 - Establish an offline communication channel for emergencies
 - Disable unnecessary remote management tools (especially those mentioned in the report: AnyDesk, Atera, etc.)
 - Implement strict monitoring of remote access tools
 - Implement application whitelisting to prevent unauthorized executables
 - Verify digital signatures on software installations
 - Be particularly vigilant with software that masquerades as legitimate security tools like Avast
 - Set up alerts for suspicious command-line parameters (especially those containing "--headless" flags)
- Monitor for suspicious registry modifications, especially to authentication providers



Mitigations - CVE

Apache ActiveMQ (CVE-2023-46604)

- Upgrade to Apache ActiveMQ versions 5.15.16, 5.16.7, 5.17.6, or 5.18.3 (or newer)
- If immediate patching isn't possible, implement network filtering to block remote access to the ActiveMQ admin console
- Monitor for unusual connection attempts to ActiveMQ instances
- Restrict access using firewall rules to allow connections only from trusted IP addresses

Atlassian Confluence (CVE-2023-22515)

- Immediately update to fixed versions (7.19.16, 8.3.3, 8.4.3, 8.5.2, or later)
- Implement web application firewalls to filter malicious requests
- Monitor authentication logs for unexpected admin account creation
- Consider temporarily disabling public access until patching is complete

Citrix NetScaler ADC & Gateway (CVE-2023-3519)

- Update to recommended firmware versions that fix this vulnerability
- Implement IP restrictions for management interfaces
- Deploy proper TLS inspection for encrypted traffic
- Consider implementing additional authentication mechanisms

Fortinet FortiOS SSL-VPN & FortiProxy (CVE-2023-27997)

- Update to the latest FortiOS versions with the security patches
- Enforce multi-factor authentication for VPN access
- Implement geolocation-based access restrictions
- Monitor for unusual VPN connection patterns



Fortinet FortiClientEMS (CVE-2023-48788)

- Apply the latest security patches from Fortinet
- Segment networks where FortiClient management servers are located
- Restrict administrative access to these systems

F5 BIG-IP (CVE-2023-46747)

- Apply the hotfixes provided by F5
- Implement control plane access restrictions
- Consider implementing proper management network isolation
- Use the latest BIG-IP iApp templates with security settings enabled

Windows NetLogon (CVE-2020-1472, aka Zerologon)

- Apply Microsoft security updates
- Enable enforcement mode for secure RPC
- Monitor for unauthorized attempts to reset machine account passwords
- Implement Windows Event log monitoring to detect exploitation attempts

Windows BITS (CVE-2020-0787)

- Apply appropriate Microsoft security patches
- Consider disabling the BITS service if not essential
- Implement application control to prevent unauthorized use of BITSAdmin
- Monitor for suspicious BITS transfer jobs using event logs



Windows SMBv1 (CVE-2017-0144, aka EternalBlue)

- Disable SMBv1 completely across the enterprise
- Block SMB ports (TCP 445) at network boundaries
- Implement strict network segmentation for legacy systems that require SMBv1
- Use Microsoft's SMB security features like SMB signing and encryption

Additional technical mitigations:

- Implement specific detection and alerting for lateral movement attempts using these vulnerabilities
- Deploy network traffic analysis to detect exploitation attempts
- Consider micro-segmentation to isolate critical assets that might be targeted via these vulnerabilities
- Conduct regular vulnerability scanning focused on these specific CVEs
- Develop incident response playbooks specific to exploitation of these vulnerabilities



ThreatMon

Under Cyber Wings

More Information About ThreatMon



One Platform for all intelligence needs.

ThreatMon End-to-end intelligence is a cutting-edge, cloud-based SaaS platform that continuously monitors the dark and surface web, providing early warnings and actionable insights into emerging threats.

We are a SaaS platform designed to help businesses proactively detect and address threats before a cyber attack occurs. Unlike traditional cyber threat intelligence, we provide comprehensive and holistic cyber intelligence.

- Attack Surface Intelligence
- Fraud Intelligence
- Dark and Surface Web Intelligence
- Threat Intelligence



Contact Us:



Email Address
team@threatmonit.io



<https://x.com/MonThreat>



<https://www.linkedin.com/company/threatmon>