Agenda Ransomware Deploys Linux Variant on Windows Systems Through Remote Management Tools and BYOVD Techniques

⋮ 10/22/2025

Ransomware

Trend™ Research identified a sophisticated Agenda ransomware attack that deployed a Linux variant on Windows systems. This cross-platform execution can make detection challenging for enterprises.

By: Jacob Santos, Junestherry Dela Cruz, Sarah Pearl Camiling, Sophia Nilette Robles, Maristel Policarpio, Raymart Yambot Oct 23, 2025 Read time: 11 min (2845 words)

Key takeaways:

- Trend™ Research identified Agenda ransomware group, known as Qilin, deploying a Linux-based ransomware binary on Windows hosts by abusing legitimate remote management and file transfer tools. The cross-platform execution sidesteps Windows-centric detections and security solutions, including conventional endpoint detection and response platforms.

- The technique enables low-noise operations that can disable recovery options through the targeted theft of backup credentials and neutralize endpoint defenses via BYOVD attack.
- • Agenda has affected more than 700 victims across 62 countries since January 2025, primarily targeting organizations in developed markets and high-value industries. Most victims were in the United States, France, Canada, and the United Kingdom, with manufacturing, technology, financial services, and healthcare among the hardest hit.
- Any environment that uses remote access platforms, centralized backup solutions, or hybrid Windows/Linux infrastructures could be at risk. Enterprises are encouraged to limit the use of remote access tools to authorized hosts and continuously monitor for unusual activity.
- Trend Vision One™ detects and blocks the specific IoCs mentioned in this blog, and offers customers access to hunting queries, threat insights, and intelligence reports related to Agenda ransomware. For more security best practices, see the guidance below.

Trend™ Research identified a sophisticated ransomware attack by the Agenda group that deployed their Linux ransomware variant on Windows systems. This follows a similar attack observed last June 2025, where MeshAgent and MeshCentral was used for deployment. In this recent incident, the threat actors utilized a novel deployment method combining WinSCP for secure file transfer and Splashtop Remote for executing the Linux ransomware binary on Windows machines.

The attack chain demonstrated advanced techniques including usage of Bring Your Own Vulnerable Driver (BYOVD) for defense evasion and deployment of multiple SOCKS proxy instances across various system directories to obfuscate command-and-control (C&C) traffic. The attackers abused legitimate tools, specifically installing AnyDesk through ATERA Networks' remote monitoring and management (RMM) platform and ScreenConnect for command execution. It abuses Splashtop for the final ransomware execution. They specifically targeted Veeam backup infrastructure using specialized credential extraction tools, systematically harvesting credentials from multiple backup databases to compromise the organization's disaster recovery capabilities before deploying the ransomware payload.

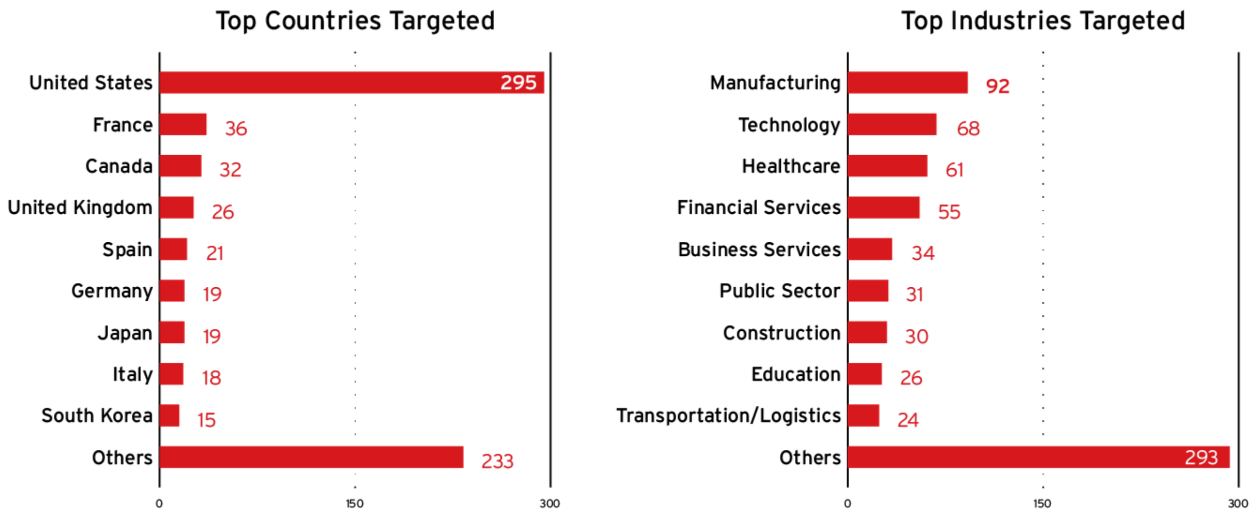
This attack challenges traditional Windows-focused security controls. The deployment of Linux ransomware on Windows systems demonstrates how threat actors are adapting to bypass endpoint detection systems not configured to detect or prevent Linux binaries executing through remote management channels.

The combination of BYOVD techniques, fake CAPTCHA social engineering, and the strategic targeting of backup infrastructure shows an approach of ensuring successful ransomware deployment while eliminating recovery options. The use of legitimate tools and cross-platform execution methods makes detection significantly more challenging. Organizations must urgently reassess their security posture to account for these unconventional attack vectors and implement enhanced monitoring of remote management tools and backup system access.

Impact and victimology

Agenda emerged as one of the top ransomware groups in 2025, demonstrating unprecedented operational tempo and global reach. Analysis of their data leak site since January reveals a ransomware-as-a-service

(RaaS) operation that systematically targeted organizations across economically developed nations, with a particular focus on the United States, Western Europe, and Japan. The victimology pattern shows opportunistic targeting across multiple high-value sectors, particularly manufacturing, technology, financial services, and healthcare — industries characterized by operational sensitivity, data criticality, and higher likelihood of ransom payment.



©2025 TREND MICRO

Figure 1. Geographic and sectoral distribution of victims since January 2025 based on the Agenda ransomware group’s leak
download

The group’s willingness to target critical infrastructure, including healthcare facilities and public sector entities, emphasizes their lack of ethical constraints and prioritization of financial gain over potential societal impact. Figure 1 illustrates the geographic and sectoral distribution of Agenda’s 2025 victims as documented on their data leak site, providing a visual representation of the threat actor's extensive global reach and multi-industry impact.

Attack chain

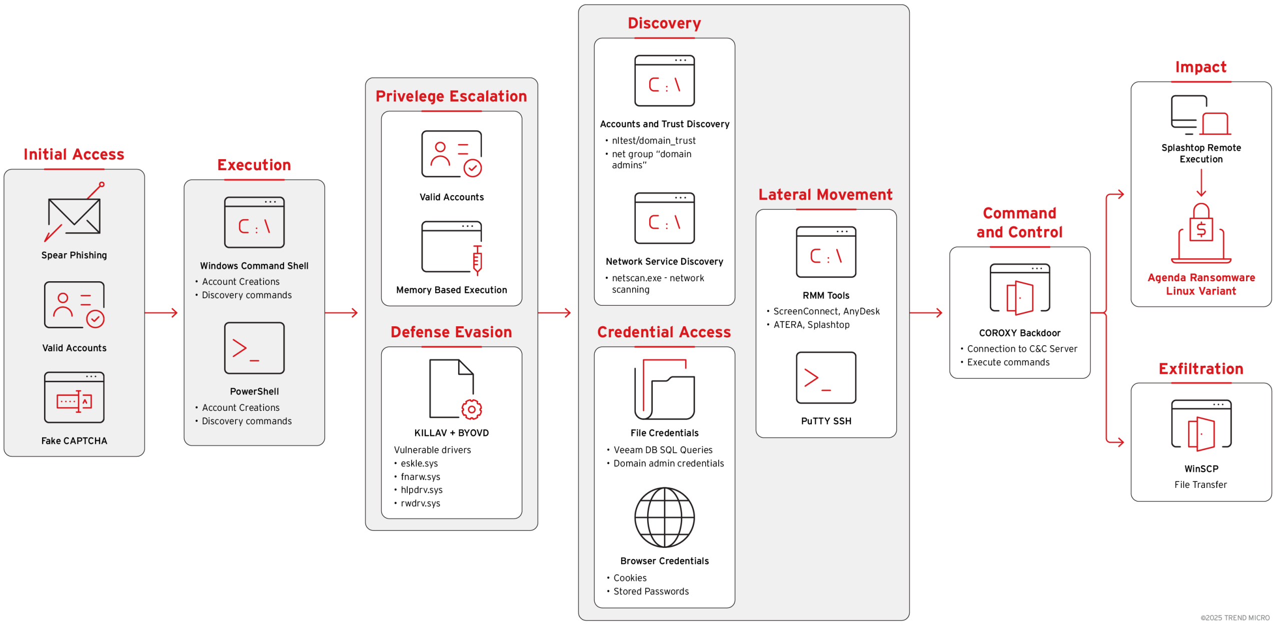


Figure 2. The Agenda ransomware infection chain

download

Initial Access

We identified that multiple endpoints within the compromised environment had connected to malicious fake CAPTCHA pages hosted on Cloudflare R2 storage infrastructure. These pages presented convincing replicas of legitimate Google CAPTCHA verification prompts:

- hxxps://pub-959ff112c2eb41ce8f7b24e38c9b4f94[.]r2[.]dev/Google-Captcha-Continue-Latest-J-KL-3[.]html
- hxxps://pub-2149a070e76f4ccabd67228f754768dc[.]r2[.]dev/I-Google-Captcha-Continue-Latest-27-L-1[.]html

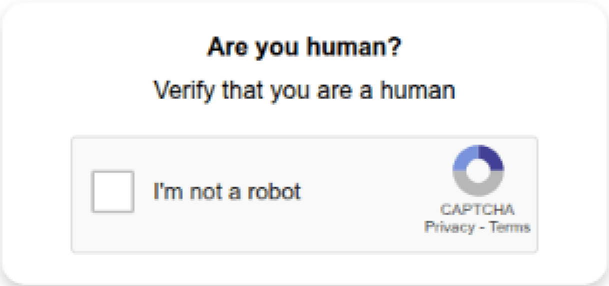


Figure 3. Screenshot of the webpage hosted on Cloudflare R2, displaying a fake Google CAPTCHA verification prompt designed to trick users into executing malicious commands

Analysis of the embedded obfuscated JavaScript within these fake CAPTCHA pages revealed a multistage payload delivery system that initiated downloads from secondary command-and-control servers:

- 45[.]221[.]64[.]245/mot/
- 104[.]164[.]55[.]7/231/means.d

We assess that the threat actors likely initiated their attack campaign through a sophisticated social engineering scheme involving these fake CAPTCHA pages. The pages appear to have delivered information stealers to the compromised endpoints, which subsequently harvested authentication tokens, browser cookies, and stored credentials from the infected systems. The presence of valid credentials used throughout the attack chain strongly suggests that these stolen credentials provided the Agenda threat actors with the valid accounts necessary for their initial access into the environment. This assessment is further supported by the attackers' ability to bypass multifactor authentication (MFA) and move laterally using legitimate user sessions, indicating they possessed harvested credentials rather than relying on traditional exploitation techniques.

Privilege Escalation

The attackers deployed a SOCKS proxy DLL to facilitate remote access and command execution. This proxy was loaded directly into memory using Windows' legitimate rundll32.exe process, making detection more difficult.

```
|── C:\Windows\System32\cmd.exe
```

└── *C:\Windows\System32\rundll32.exe*

└── *rundll32.exe socks64.dll,rundll*

└── *C:\ProgramData\Veeam\socks64.dll*

A backdoor administrative account named "*Supportt*" was created to ensure persistent elevated access. This account name was likely chosen to blend in with legitimate support accounts commonly found in enterprise environments.

- *net user Supportt ***** /add*
- *net localgroup Administrators Supportt /add*

The legitimate administrator account password was also reset to maintain control and prevent legitimate administrators from regaining access.

- *net user Administrator *****

Discovery

Extensive reconnaissance was conducted to map the network infrastructure. The attackers abused ScreenConnect's legitimate remote management capabilities to execute discovery commands through temporary command scripts, systematically enumerating domain trusts and identifying privileged accounts while appearing as normal administrative activity:

- nltest /domain_trusts
- net group "domain admins" /domain

Network scanning tools were deployed across multiple locations to discover additional systems, services, and potential lateral movement targets. The NetScan utility was executed from both the Desktop and Documents folders to perform comprehensive network enumeration.

- C:\Users\Administrator.<REDACTED>\Desktop\netscan.exe
- C:\Users\Administrator.<REDACTED>\Documents\netscan.exe

Remote management tools were strategically installed through legitimate RMM platforms to blend with normal IT operations. ATERA Networks' agent was leveraged to deploy AnyDesk version 9.0.5, while ScreenConnect provided an additional command execution vector. This dual-RMM approach provided the attackers with redundant remote access capabilities that appeared legitimate to security monitoring systems, allowing them to maintain persistent access even if one tool was discovered and removed.

Credential Access

The attackers specifically targeted Veeam backup infrastructure to harvest credentials, recognizing that backup systems often store credentials for accessing multiple systems across the enterprise. PowerShell scripts were executed with base64-encoded payloads to extract and decrypt stored credentials from Veeam databases, via powershell.exe -e [base64-encoded payload].

When decoded, these scripts revealed systematic targeting of multiple Veeam backup databases, each containing credentials for different segments of the infrastructure:

SQL Database Queries:

- SELECT [user_name], [password] FROM [VeeamBackup].[dbo].[Credentials]
- Targeted tables: Credentials, BackupRepositories, WinServers

Compromised Account Types:

- Domain administrator accounts: DOMAIN\admin-***, DOMAIN\da-backup-***
- Service accounts: svc-sql-***, DOMAIN\veeam-svc-***, svc-exchange-***
- Local administrators: SERVER01\Administrator, SERVER02\localadmin

Script Details:

- Decryption key found in script: 0jmz9Hrgy08rc0XrNpQ***[REDACTED]***
- Affected systems: Domain controllers, Exchange servers, SQL databases, file servers, backup repositories

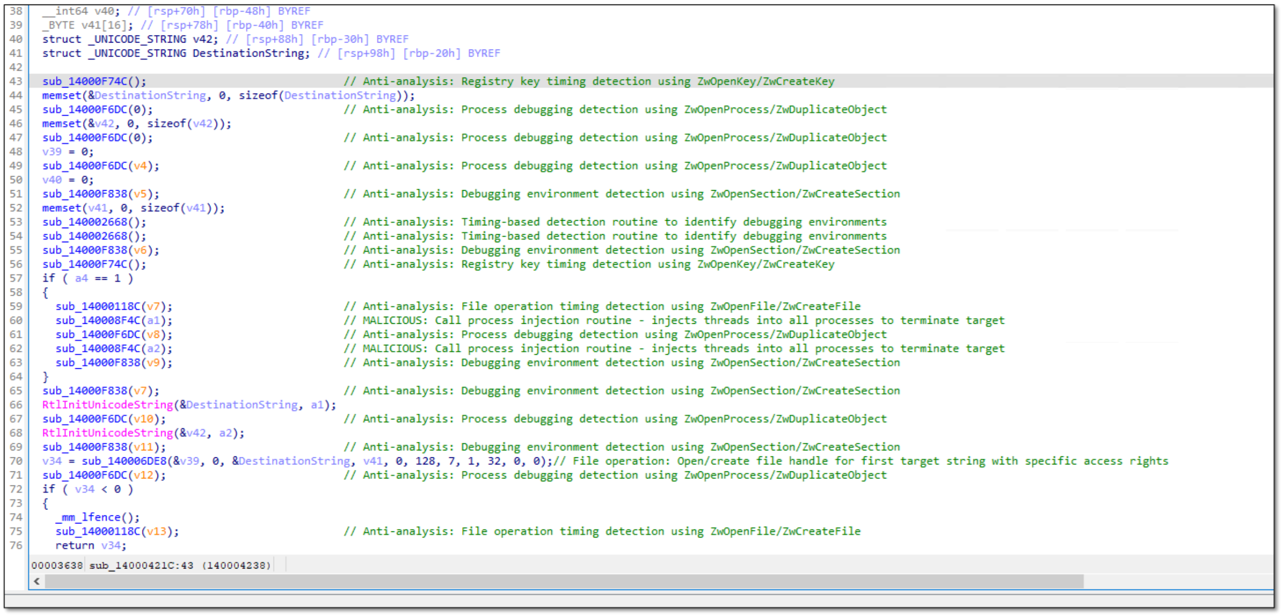
This approach provided the attackers with a comprehensive set of credentials for remote systems, domain controllers, and critical servers stored within the backup infrastructure.

Defense Evasion

The attackers deployed sophisticated anti-analysis tools to evade security solutions. Further probe confirmed that both 2stX.exe and Or2.exe utilize the eskle.sys driver for anti-AV capabilities through a BYOVD attack:

- C:\Users\Administrator.<REDACTED>\Downloads\2stX.exe
- C:\Users\Administrator.<REDACTED>\Downloads\Or2.exe
    - C:\Users\Administrator.<REDACTED>\Downloads\2stX\eskle.sys

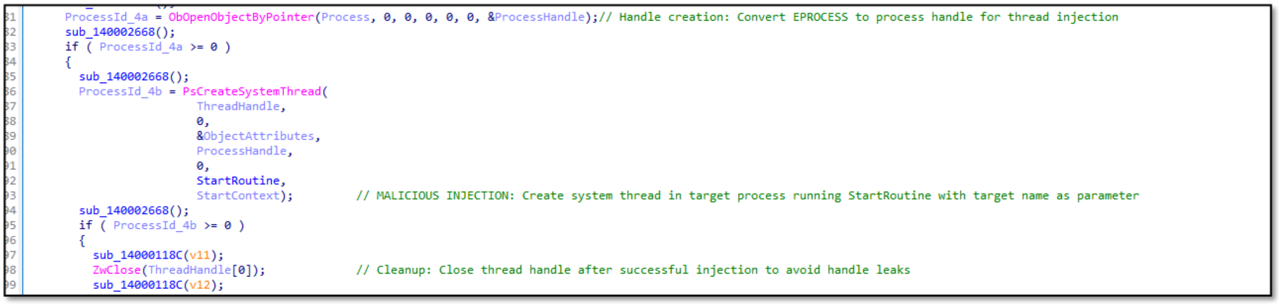
The eskle.sys driver was utilized to disable security solutions, terminate processes, and evade detection. Although these files could have been downloaded or copied onto the machine earlier, the origin of the eskle.sys driver is unclear. Its digital signature lists the vendor as “拇指世界（北京）网络科技有限公“ (translated: Thumb World (Beijing) Network Technology Co., Ltd.), which appears to be associated with the game.bb site. The driver likely belongs to a game-related package and is commonly used by cheat developers to evade anti-cheat systems; however, it could also be repurposed by advanced persistent threat actors.



```
__int64 v40; // [rsp+70h] [rbp-48h] BYREF
_BYTE v41[16]; // [rsp+78h] [rbp-40h] BYREF
struct _UNICODE_STRING v42; // [rsp+88h] [rbp-30h] BYREF
struct _UNICODE_STRING DestinationString; // [rsp+98h] [rbp-20h] BYREF

sub_14000F74C();                            // Anti-analysis: Registry key timing detection using ZwOpenKey/ZwCreateKey
memset(&DestinationString, 0, sizeof(DestinationString));
sub_14000F6DC(0);                           // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
memset(&v42, 0, sizeof(v42));
sub_14000F6DC(0);                           // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
v39 = 0;
sub_14000F6DC(v4);                          // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
v40 = 0;
sub_14000F838(v5);                          // Anti-analysis: Debugging environment detection using ZwOpenSection/ZwCreateSection
memset(v41, 0, sizeof(v41));
sub_140002668();                            // Anti-analysis: Timing-based detection routine to identify debugging environments
sub_140002668();                            // Anti-analysis: Timing-based detection routine to identify debugging environments
sub_14000F838(v6);                          // Anti-analysis: Debugging environment detection using ZwOpenSection/ZwCreateSection
sub_14000F74C();                            // Anti-analysis: Registry key timing detection using ZwOpenKey/ZwCreateKey
if ( a4 == 1 )
{
  sub_14000118C(v7);                        // Anti-analysis: File operation timing detection using ZwOpenFile/ZwCreateFile
  sub_140008F4C(a1);                        // MALICIOUS: Call process injection routine - injects threads into all processes to terminate target
  sub_14000F6DC(v8);                        // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
  sub_140008F4C(a2);                        // MALICIOUS: Call process injection routine - injects threads into all processes to terminate target
  sub_14000F838(v9);                        // Anti-analysis: Debugging environment detection using ZwOpenSection/ZwCreateSection
}
sub_14000F838(v7);                          // Anti-analysis: Debugging environment detection using ZwOpenSection/ZwCreateSection
RtlInitUnicodeString(&DestinationString, a1);
sub_14000F6DC(v10);                         // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
RtlInitUnicodeString(&v42, a2);
sub_14000F838(v11);                         // Anti-analysis: Debugging environment detection using ZwOpenSection/ZwCreateSection
v34 = sub_140006DE8(&v39, 0, &DestinationString, v41, 0, 128, 7, 1, 32, 0, 0);// File operation: Open/create file handle for first target string with specific access rights
sub_14000F6DC(v12);                         // Anti-analysis: Process debugging detection using ZwOpenProcess/ZwDuplicateObject
if ( v34 < 0 )
{
  _mm_lfence();
  sub_14000118C(v13);                       // Anti-analysis: File operation timing detection using ZwOpenFile/ZwCreateFile
  return v34;
00003638 sub_14000421C:43 (140004238)
```

Figure 4. Disassembly showing eskle.sys' anti-analysis capabilities, including virtual machine (VM) detection and debugging countermeasures

download



```
    ProcessId_4a = ObOpenObjectByPointer(Process, 0, 0, 0, 0, 0, &ProcessHandle);// Handle creation: Convert EPROCESS to process handle for thread injection
    sub_140002668();
    if ( ProcessId_4a >= 0 )
    {
      sub_140002668();
      ProcessId_4b = PsCreateSystemThread(
                       ThreadHandle,
                       0,
                       &ObjectAttributes,
                       ProcessHandle,
                       0,
                       StartRoutine,
                       StartContext);         // MALICIOUS INJECTION: Create system thread in target process running StartRoutine with target name as parameter
      sub_140002668();
      if ( ProcessId_4b >= 0 )
      {
        sub_14000118C(v11);
        ZwClose(ThreadHandle[0]);             // Cleanup: Close thread handle after successful injection to avoid handle leaks
        sub_14000118C(v12);
```

Figure 5. Disassembly showing eskle.sys' process termination capabilities

download

The eskle.sys driver forcibly stops programs by creating a handle to the target process, starting a new thread to run a termination routine, and cleaning up the handle. This enables it to disable security software, disrupt system operations, and maintain persistence.

An additional component named *msimg32.dll* was identified in our internal telemetry alongside relations to *ThrottleStop.sys*. Further analysis through controlled testing revealed that msimg32.dll functions as a dropper that deploys two driver files when executed:

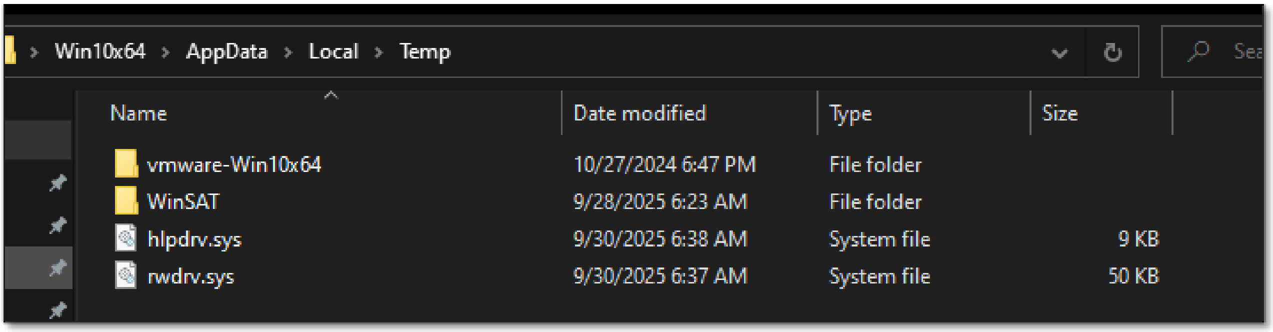
- C:\Users\Administrator.<REDACTED>\Downloads\msimg32.dll

Upon successful execution, the following drivers were dropped:

- %TEMP%\rwdrv.sys
- %TEMP%\hlpdrv.sys

This connection is significant, as both rwdrv.sys and hlpdrv.sys have been previously documented in Akira campaigns for gaining kernel-level access and potentially terminating traditional endpoint detection and response (EDR) solutions. Analysis revealed that msimg32.dll employs a DLL sideloading technique, requiring a legitimate host executable for proper execution. The DLL failed to load through standard methods like regsvr32 or rundll32.

However, testing confirmed successful loading when placed alongside compatible binaries, such as FoxitPDFReader.exe, which imports msimg32.dll as a dependency. Upon execution, the application loads the malicious DLL, which then drops both driver files to the system's temporary directory, as shown in Figure 5.



Name	Date modified	Type	Size
vmware-Win10x64	10/27/2024 6:47 PM	File folder	
WinSAT	9/28/2025 6:23 AM	File folder	
hlpdrv.sys	9/30/2025 6:38 AM	System file	9 KB
rwdrv.sys	9/30/2025 6:37 AM	System file	50 KB

Figure 6. File system view showing hlpdrv.sys and rwdrv.sys dropped by msimg32.dll in the Windows Temp directory after DLL sideloading execution

Additionally, we observed and analyzed three other executables (cg6.exe, 44a.exe, aa.exe) that were identified as potential anti-AV tools based on their behavioral patterns and code similarities. Analysis revealed that these executables contain driver-loading routines and process manipulation capabilities consistent with BYOVD techniques. These tools are suspected to utilize a different vulnerable driver (fnarw.sys), though definitive confirmation remains pending as the driver was unavailable for complete analysis:

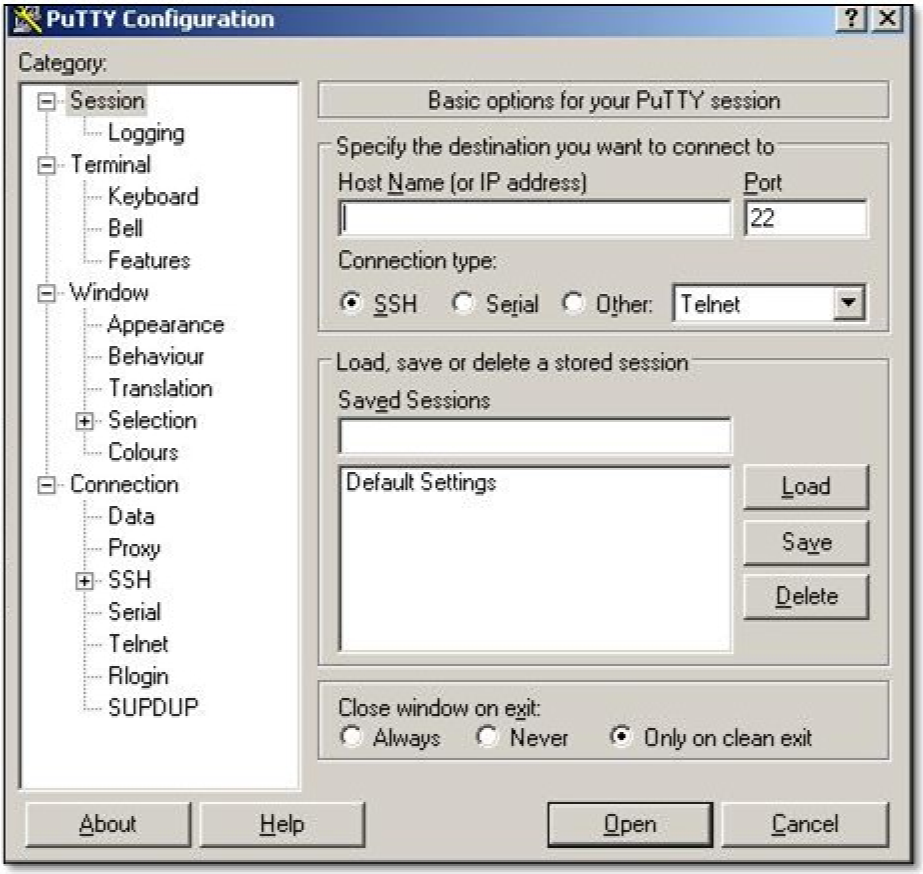
- C:\Users\<REDACTED>\Desktop\cg6.exe
- C:\Users\<REDACTED>\Desktop\44a.exe
- C:\Users\<REDACTED>\Desktop\aa.exe

Lateral Movement

Multiple PuTTY SSH clients were systematically deployed on compromised systems to facilitate lateral movement to Linux systems within the environment. The attackers staged these tools with different filenames but identical functionality:

- C:\Users\<REDACTED>\Desktop\test.exe
- C:\Users\<REDACTED>\Desktop\1.exe
- C:\Users\<REDACTED>\Desktop\2.exe
- C:\Users\<REDACTED>\Desktop\3.exe

These renamed PuTTY executables enabled the attackers to establish SSH connections to Linux infrastructure, expanding their reach beyond Windows systems and demonstrating the cross-platform nature of the attack.



Specify the destination you want to connect to

Host Name (or IP address) Port 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 7. PuTTY SSH client interface, deployed under various filenames (e.g., test.exe, 1.exe, 2.exe, 3.exe) on compromised systems

Command and Control

The threat actors established a C&C infrastructure through the deployment of multiple SOCKS proxy instances, identified as COROXY backdoor. These proxies were systematically placed across various system directories to create a distributed network of communication channels that obfuscated malicious traffic patterns and evaded network monitoring solutions.

The attackers positioned these SOCKS proxies in directories associated with legitimate enterprise software, including Veeam backup solutions, VMware virtualization infrastructure, and Adobe applications. This placement strategy served a dual purpose: blending malicious C&C traffic with normal application communications while exploiting the trust typically afforded to these well-known software vendors in enterprise environments.

- C:\ProgramData\Veeam\socks64.dll
- C:\ProgramData\USOShared\socks64.dll
- C:\ProgramData\VMware\logs\socks64.dll

- C:\ProgramData\Adobe\socks64.dll
- C:\ProgramData\Veeam\Backup\OracleLogBackup\socks64.dll

The distributed nature of this SOCKS proxy deployment provided the attackers with redundant communication channels, ensuring persistent C&C capabilities even if individual proxies were discovered and removed. Each proxy instance functioned as an independent tunnel for encrypted communications, allowing the threat actors to maintain remote access, exfiltrate data, and orchestrate subsequent attack stages while remaining concealed within legitimate network traffic flows.

Impact

The final ransomware deployment showcased cross-platform execution. WinSCP was utilized for secure file transfer of the Linux ransomware binary to the Windows system:

C:\Users\<REDACTED>\AppData\Local\Programs\WinSCP\WinSCP.exe

└── C:\Users\<REDACTED>\Desktop\mmh_linux_x86-64.filepart

└── C:\Users\<REDACTED>\Desktop\mmh_linux_x86-64

Unique to the technique was the use of Splashtop Remote's management service (SRManager.exe) to execute the Linux ransomware binary directly on Windows systems:

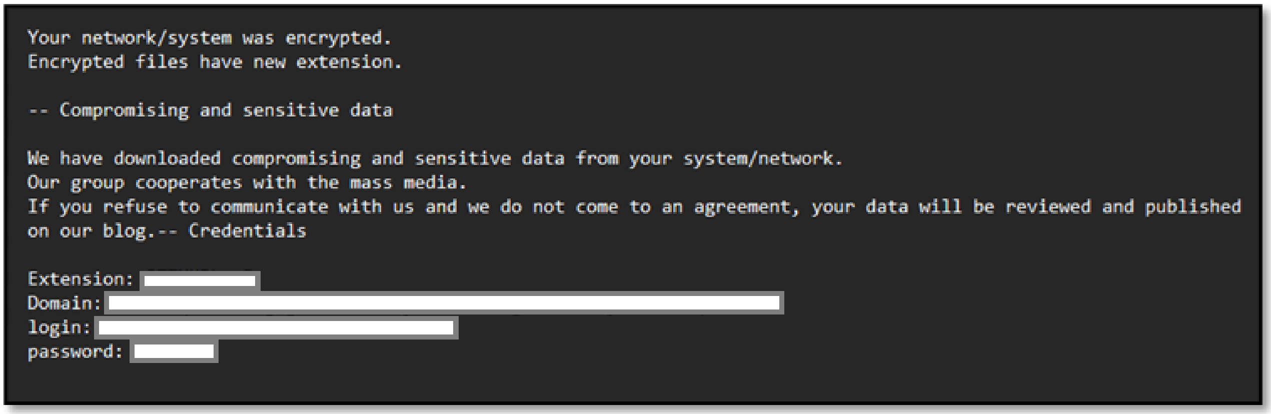
C:\Program Files (x86)\Splashtop\Splashtop Remote\Server\SRManager.exe

└── C:\Users\<REDACTED>\Desktop\mmh_linux_x86-64

This unconventional approach used Splashtop's remote execution capabilities to run the cross-platform payload, bypassing traditional Windows-focused security controls. The execution method is significant, as most endpoint detection systems are not configured to monitor or prevent Linux binaries being executed through legitimate remote management tools on Windows platforms. The Linux ransomware binary provided cross-platform capability, allowing the attackers to impact both Windows and Linux systems within the environment using a single payload.

Linux variant ransomware analysis

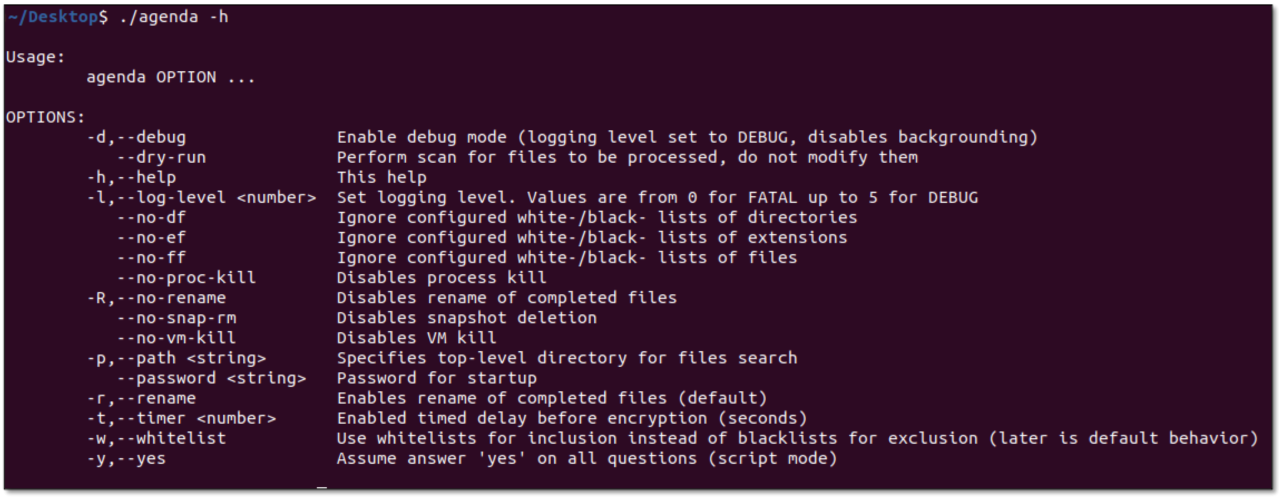
Analysis of the Linux ransomware binary revealed an advanced cross-platform payload with extensive configuration capabilities and platform-specific targeting.



```
Your network/system was encrypted.
Encrypted files have new extension.

-- Compromising and sensitive data

We have downloaded compromising and sensitive data from your system/network.
Our group cooperates with the mass media.
If you refuse to communicate with us and we do not come to an agreement, your data will be reviewed and published
on our blog.-- Credentials

Extension: [redacted]
Domain: [redacted]
login: [redacted]
password: [redacted]
```

Figure 8. Ransom note extracted from the binary

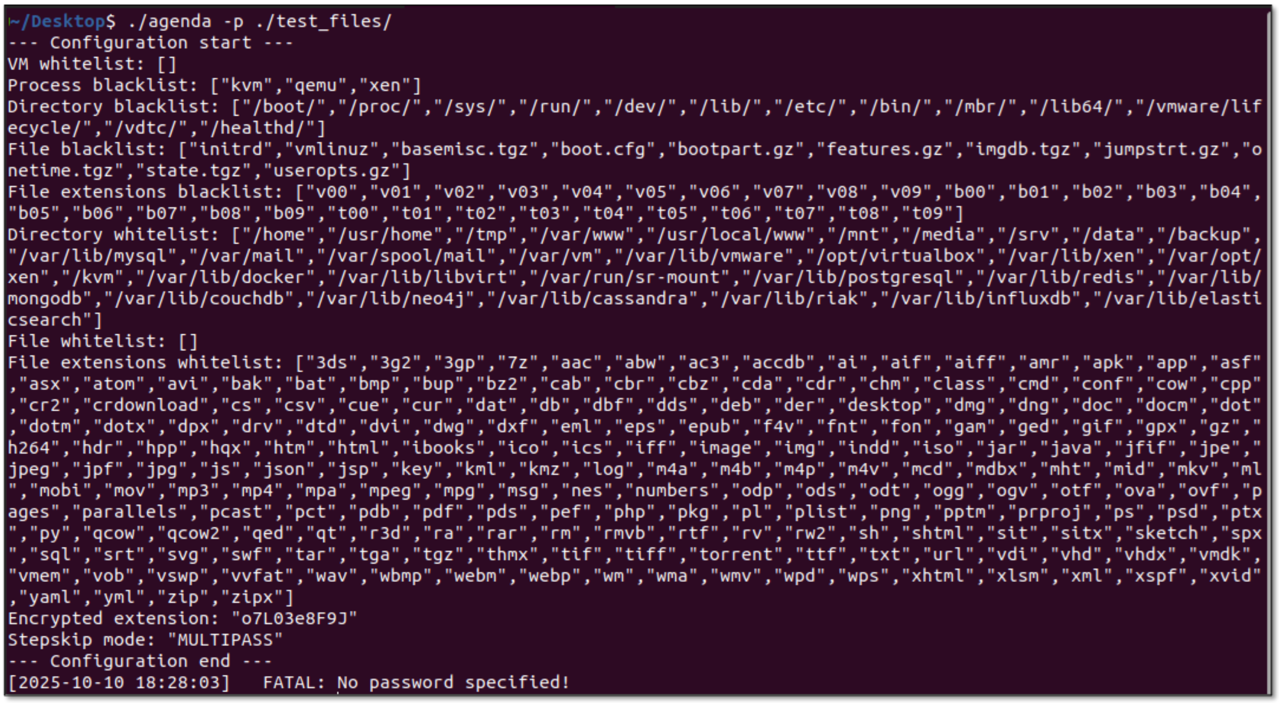
The ransomware deployed a standard Agenda ransom note threatening data publication and providing victim-specific credentials for negotiation. The note included file extension, and domain/login/password fields for accessing the threat actors' communication portal.



```
~/Desktop$ ./agenda -h

Usage:
        agenda OPTION ...

OPTIONS:
        -d,--debug                Enable debug mode (logging level set to DEBUG, disables backgrounding)
           --dry-run              Perform scan for files to be processed, do not modify them
        -h,--help                 This help
        -l,--log-level <number>   Set logging level. Values are from 0 for FATAL up to 5 for DEBUG
           --no-df                Ignore configured white-/black- lists of directories
           --no-ef                Ignore configured white-/black- lists of extensions
           --no-ff                Ignore configured white-/black- lists of files
           --no-proc-kill         Disables process kill
        -R,--no-rename            Disables rename of completed files
           --no-snap-rm           Disables snapshot deletion
           --no-vm-kill           Disables VM kill
        -p,--path <string>        Specifies top-level directory for files search
           --password <string>    Password for startup
        -r,--rename               Enables rename of completed files (default)
        -t,--timer <number>       Enabled timed delay before encryption (seconds)
        -w,--whitelist            Use whitelists for inclusion instead of blacklists for exclusion (later is default behavior)
        -y,--yes                  Assume answer 'yes' on all questions (script mode)
```

Figure 9. Command-line parameters

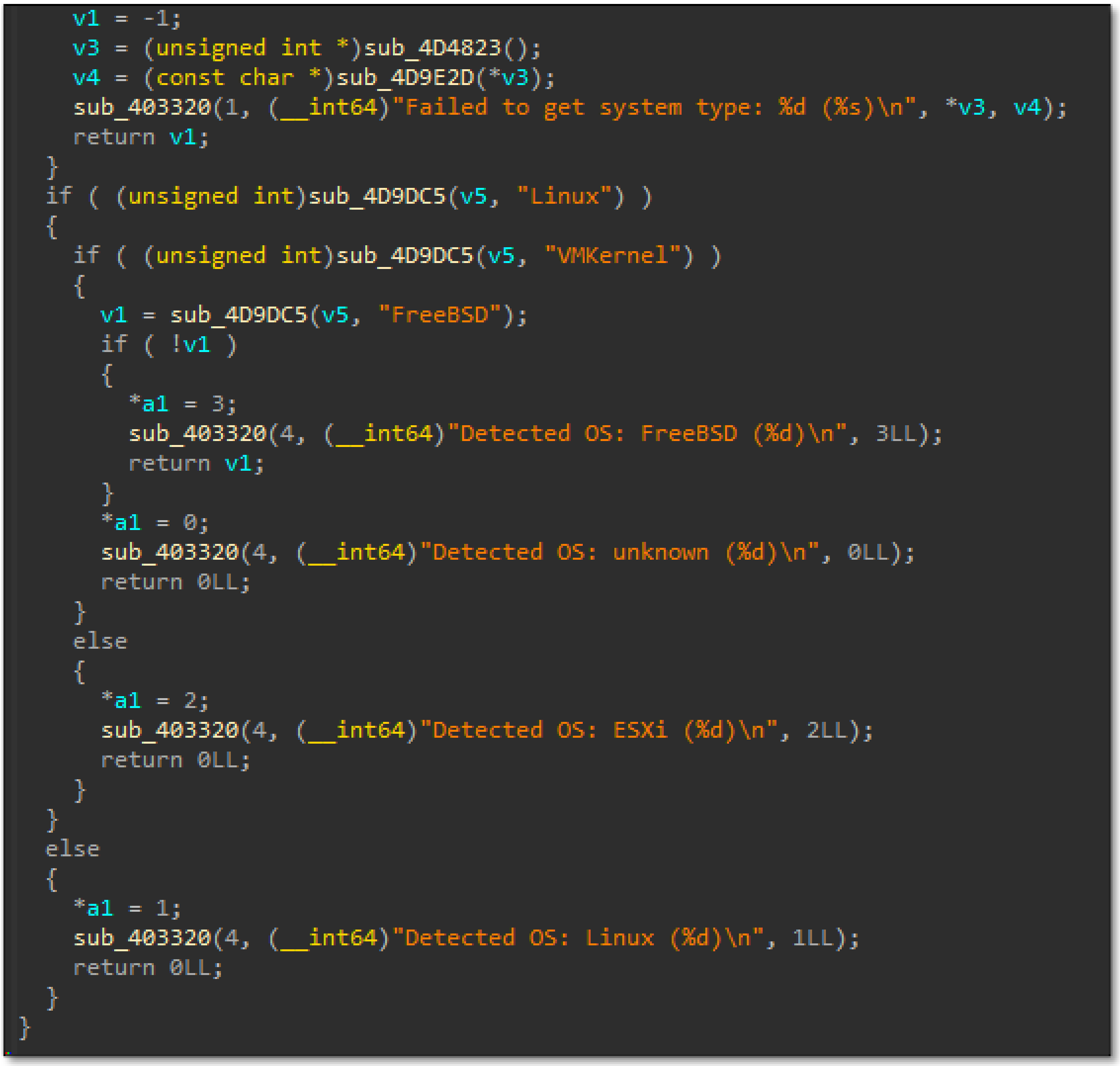
The binary implemented comprehensive command-line options including debug mode (-d), logging levels (-l), path specifications (-p), whitelist configurations, and encryption control parameters. Notable features included timer delays (-t) for delayed execution and a "yes" mode (-y) for automated operation without user prompts, indicating operational maturity.



```
~/Desktop$ ./agenda -p ./test_files/
--- Configuration start ---
VM whitelist: []
Process blacklist: ["kvm","qemu","xen"]
Directory blacklist: ["/boot/","/proc/","/sys/","/run/","/dev/","/lib/","/etc/","/bin/","/mbr/","/lib64/","/vmware/lif
ecycle/","/vdtc/","/healthd/"]
File blacklist: ["initrd","vmlinuz","basemisc.tgz","boot.cfg","bootpart.gz","features.gz","imgdb.tgz","jumpstrt.gz","o
netime.tgz","state.tgz","useropts.gz"]
File extensions blacklist: ["v00","v01","v02","v03","v04","v05","v06","v07","v08","v09","b00","b01","b02","b03","b04",
"b05","b06","b07","b08","b09","t00","t01","t02","t03","t04","t05","t06","t07","t08","t09"]
Directory whitelist: ["/home","/usr/home","/tmp","/var/www","/usr/local/www","/mnt","/media","/srv","/data","/backup",
"/var/lib/mysql","/var/mail","/var/spool/mail","/var/vm","/var/lib/vmware","/opt/virtualbox","/var/lib/xen","/var/opt/
xen","/kvm","/var/lib/docker","/var/lib/libvirt","/var/run/sr-mount","/var/lib/postgresql","/var/lib/redis","/var/lib/
mongodb","/var/lib/couchdb","/var/lib/neo4j","/var/lib/cassandra","/var/lib/riak","/var/lib/influxdb","/var/lib/elasti
csearch"]
File whitelist: []
File extensions whitelist: ["3ds","3g2","3gp","7z","aac","abw","ac3","accdb","ai","aif","aiff","amr","apk","app","asf"
,"asx","atom","avi","bak","bat","bmp","bup","bz2","cab","cbr","cbz","cda","cdr","chm","class","cmd","conf","cow","cpp"
,"cr2","crdownload","cs","csv","cue","cur","dat","db","dbf","dds","deb","der","desktop","dmg","dng","doc","docm","dot"
,"dotm","dotx","dpx","drv","dtd","dvi","dwg","dxf","eml","eps","epub","f4v","fnt","fon","gam","ged","gif","gpx","gz","
h264","hdr","hpp","hqx","htm","html","ibooks","ico","ics","iff","image","img","indd","iso","jar","java","jfif","jpe","
jpeg","jpf","jpg","js","json","jsp","key","kml","kmz","log","m4a","m4b","m4p","m4v","mcd","mdbx","mht","mid","mkv","ml
","mobi","mov","mp3","mp4","mpa","mpeg","mpg","msg","nes","numbers","odp","ods","odt","ogg","ogv","otf","ova","ovf","p
ages","parallels","pcast","pct","pdb","pdf","pds","pef","php","pkg","pl","plist","png","pptm","prproj","ps","psd","ptx
","py","qcow","qcow2","qed","qt","r3d","ra","rar","rm","rmvb","rtf","rv","rw2","sh","shtml","sit","sitx","sketch","spx
","sql","srt","svg","swf","tar","tga","tgz","thmx","tif","tiff","torrent","ttf","txt","url","vdi","vhd","vhdx","vmdk",
"vmem","vob","vswp","vvfat","wav","wbmp","webm","webp","wm","wma","wmv","wpd","wps","xhtml","xlsm","xml","xspf","xvid"
,"yaml","yml","zip","zipx"]
Encrypted extension: "o7L03e8F9J"
Stepskip mode: "MULTIPASS"
--- Configuration end ---
[2025-10-10 18:28:03]   FATAL: No password specified!
```

Figure 10. Screenshot showing how the application outputs its configuration to the console upon launch; a password must also be provided

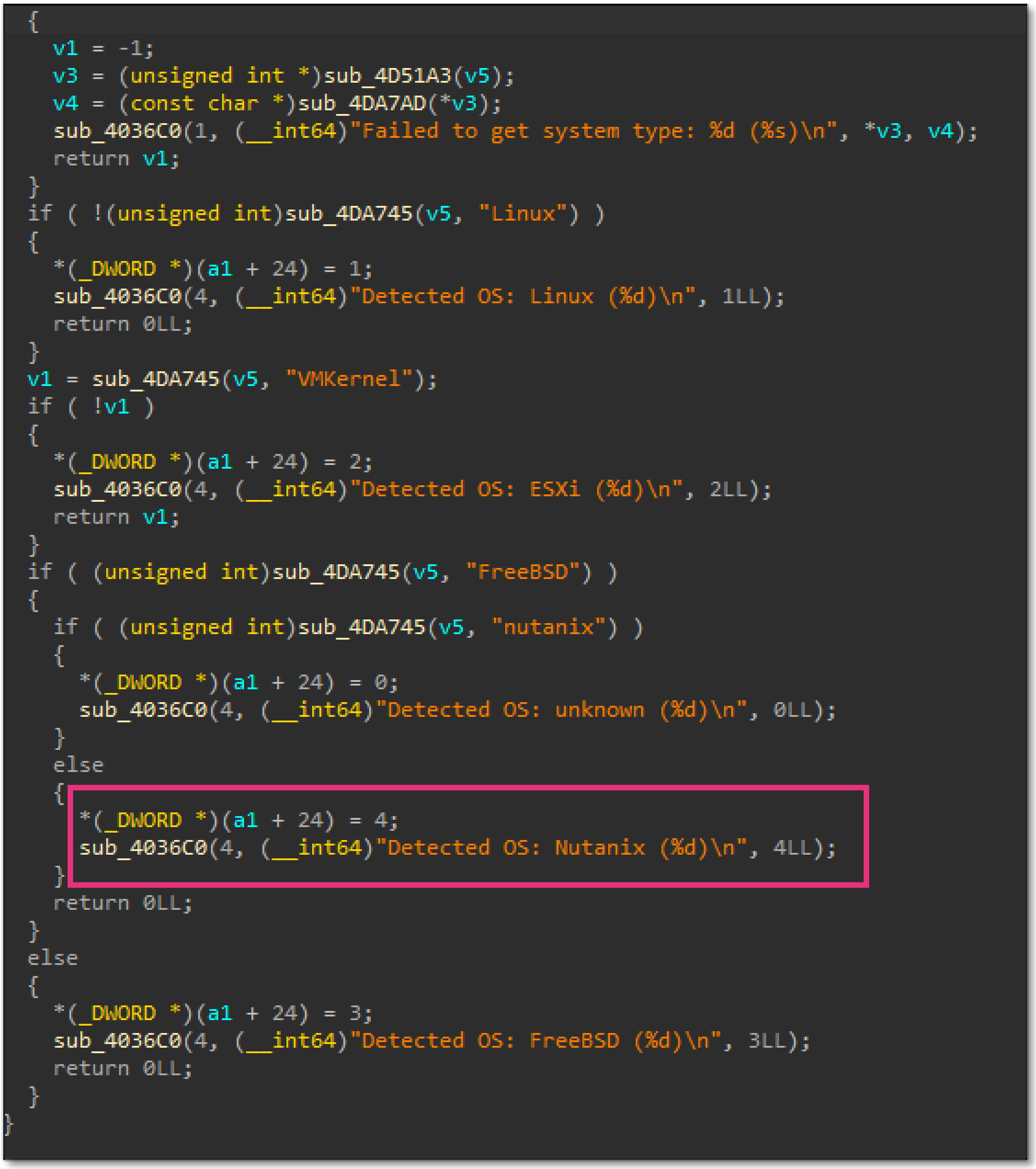
Execution required password authentication and displayed verbose configuration output including whitelisted processes, file extension blacklists, and path exclusions. The configuration showed extensive targeting of VMware ESXi paths (/vmfs/, /dev/, /lib64/) while excluding critical system directories, demonstrating hypervisor-focused deployment strategies.



```
    v1 = -1;
    v3 = (unsigned int *)sub_4D4823();
    v4 = (const char *)sub_4D9E2D(*v3);
    sub_403320(1, (__int64)"Failed to get system type: %d (%s)\n", *v3, v4);
    return v1;
  }
  if ( (unsigned int)sub_4D9DC5(v5, "Linux") )
  {
    if ( (unsigned int)sub_4D9DC5(v5, "VMKernel") )
    {
      v1 = sub_4D9DC5(v5, "FreeBSD");
      if ( !v1 )
      {
        *a1 = 3;
        sub_403320(4, (__int64)"Detected OS: FreeBSD (%d)\n", 3LL);
        return v1;
      }
      *a1 = 0;
      sub_403320(4, (__int64)"Detected OS: unknown (%d)\n", 0LL);
      return 0LL;
    }
    else
    {
      *a1 = 2;
      sub_403320(4, (__int64)"Detected OS: ESXi (%d)\n", 2LL);
      return 0LL;
    }
  }
  else
  {
    *a1 = 1;
    sub_403320(4, (__int64)"Detected OS: Linux (%d)\n", 1LL);
    return 0LL;
  }
}
```

Figure 11. OS detection from previous variants

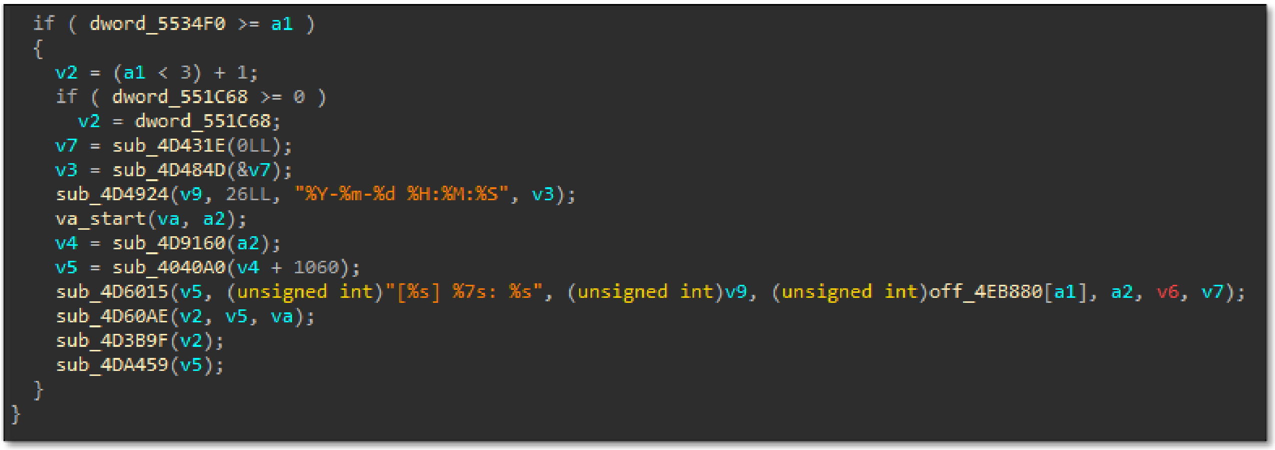
Earlier variants implemented OS detection for FreeBSD, VMkernel (ESXi), and standard Linux distributions, enabling platform-specific encryption behavior, as shown in Figure 13.



```
  {
    v1 = -1;
    v3 = (unsigned int *)sub_4D51A3(v5);
    v4 = (const char *)sub_4DA7AD(*v3);
    sub_4036C0(1, (__int64)"Failed to get system type: %d (%s)\n", *v3, v4);
    return v1;
  }
  if ( !(unsigned int)sub_4DA745(v5, "Linux") )
  {
    *(_DWORD *)(a1 + 24) = 1;
    sub_4036C0(4, (__int64)"Detected OS: Linux (%d)\n", 1LL);
    return 0LL;
  }
  v1 = sub_4DA745(v5, "VMKernel");
  if ( !v1 )
  {
    *(_DWORD *)(a1 + 24) = 2;
    sub_4036C0(4, (__int64)"Detected OS: ESXi (%d)\n", 2LL);
    return v1;
  }
  if ( (unsigned int)sub_4DA745(v5, "FreeBSD") )
  {
    if ( (unsigned int)sub_4DA745(v5, "nutanix") )
    {
      *(_DWORD *)(a1 + 24) = 0;
      sub_4036C0(4, (__int64)"Detected OS: unknown (%d)\n", 0LL);
    }
    else
    {
      *(_DWORD *)(a1 + 24) = 4;
      sub_4036C0(4, (__int64)"Detected OS: Nutanix (%d)\n", 4LL);
    }
    return 0LL;
  }
  else
  {
    *(_DWORD *)(a1 + 24) = 3;
    sub_4036C0(4, (__int64)"Detected OS: FreeBSD (%d)\n", 3LL);
    return 0LL;
  }
}
```

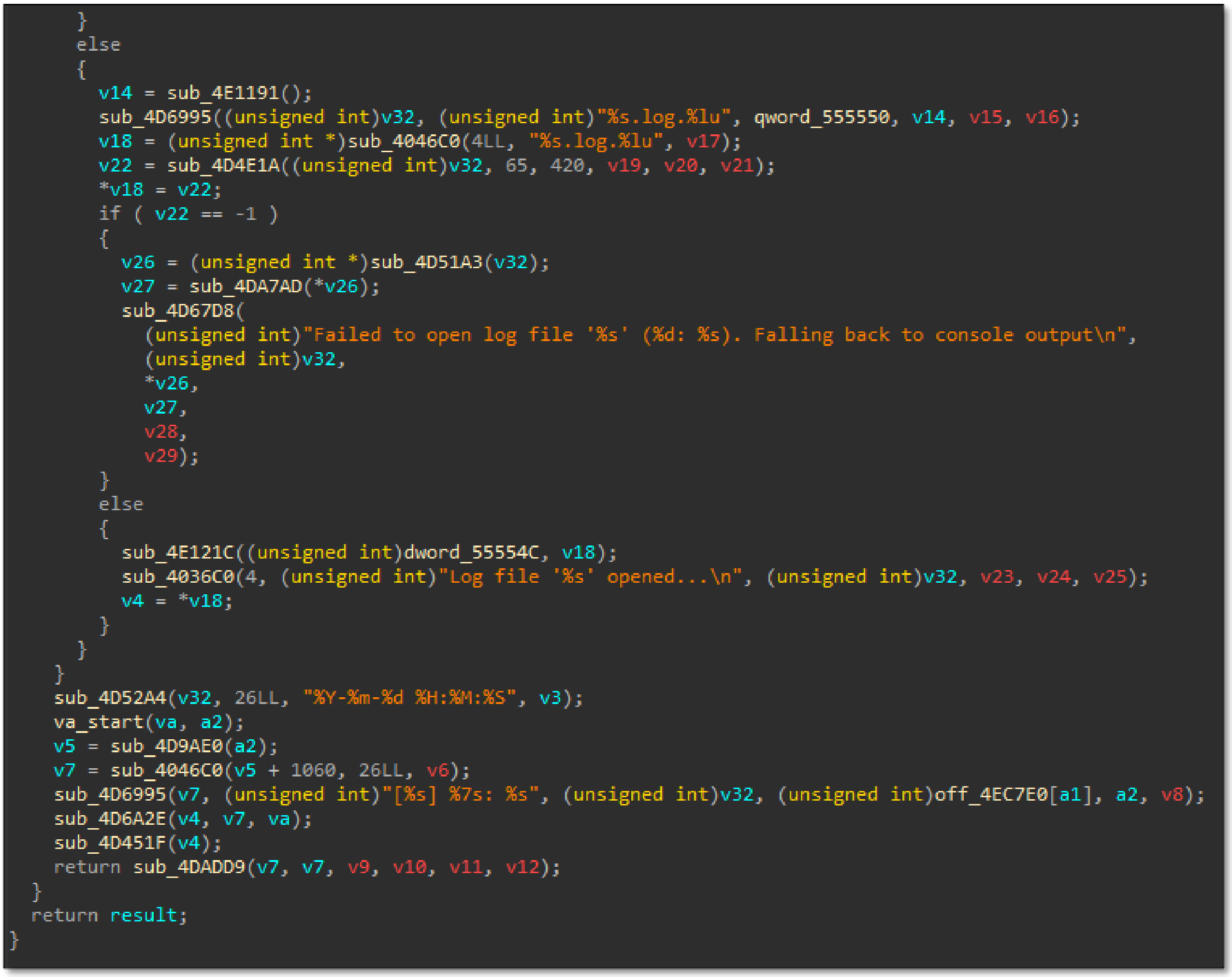
Figure 12. Recent sample with added OS checking for Nutanix

Updated samples incorporated Nutanix AHV detection, expanding targeting to include hyperconverged infrastructure platforms. This demonstrated the threat actors’ adaptation to modern enterprise virtualization environments beyond traditional VMware deployments.



```
  if ( dword_5534F0 >= a1 )
  {
    v2 = (a1 < 3) + 1;
    if ( dword_551C68 >= 0 )
      v2 = dword_551C68;
    v7 = sub_4D431E(0LL);
    v3 = sub_4D484D(&v7);
    sub_4D4924(v9, 26LL, "%Y-%m-%d %H:%M:%S", v3);
    va_start(va, a2);
    v4 = sub_4D9160(a2);
    v5 = sub_4040A0(v4 + 1060);
    sub_4D6015(v5, (unsigned int)"[%s] %7s: %s", (unsigned int)v9, (unsigned int)off_4EB880[a1], a2, v6, v7);
    sub_4D60AE(v2, v5, va);
    sub_4D3B9F(v2);
    sub_4DA459(v5);
  }
}
```

Figure 13. Old sample with its logging routine



```
      }
      else
      {
        v14 = sub_4E1191();
        sub_4D6995((unsigned int)v32, (unsigned int)"%s.log.%lu", qword_555550, v14, v15, v16);
        v18 = (unsigned int *)sub_4046C0(4LL, "%s.log.%lu", v17);
        v22 = sub_4D4E1A((unsigned int)v32, 65, 420, v19, v20, v21);
        *v18 = v22;
        if ( v22 == -1 )
        {
          v26 = (unsigned int *)sub_4D51A3(v32);
          v27 = sub_4DA7AD(*v26);
          sub_4D67D8(
            (unsigned int)"Failed to open log file '%s' (%d: %s). Falling back to console output\n",
            (unsigned int)v32,
            *v26,
            v27,
            v28,
            v29);
        }
        else
        {
          sub_4E121C((unsigned int)dword_55554C, v18);
          sub_4036C0(4, (unsigned int)"Log file '%s' opened...\n", (unsigned int)v32, v23, v24, v25);
          v4 = *v18;
        }
      }
    }
    sub_4D52A4(v32, 26LL, "%Y-%m-%d %H:%M:%S", v3);
    va_start(va, a2);
    v5 = sub_4D9AE0(a2);
    v7 = sub_4046C0(v5 + 1060, 26LL, v6);
    sub_4D6995(v7, (unsigned int)"[%s] %7s: %s", (unsigned int)v32, (unsigned int)off_4EC7E0[a1], a2, v8);
    sub_4D6A2E(v4, v7, va);
    sub_4D451F(v4);
    return sub_4DADD9(v7, v7, v9, v10, v11, v12);
  }
  return result;
}
```

Figure 14. New sample with minor modification in its logging routine

Comparison of logging routines revealed incremental improvements in error handling, with newer variants implementing enhanced file operation logging and fallback mechanisms for failed log file creation. The modifications included additional error messages and improved diagnostic output for troubleshooting deployment issues. The Linux variant's advanced capability, combined with cross-platform deployment via Splashtop Remote, represented significant tactical evolution targeting hybrid infrastructure environments.

Security best practices

This Agenda attack shows how ransomware operators are further weaponizing legitimate IT tools and hybrid environments to quietly bypass conventional security. Defenses must address operational blind spots and strengthen visibility and control over critical assets. Here are some best practices:

- **Secure remote access and RMM tools.** Limit RMM platforms to authorized management hosts and enforce MFA. Monitor for abnormal activity, such as logins outside business hours or lateral movement between unexpected endpoints. Consider restricting what applications or scripts are allowed to run on specific systems to reduce the risk of abusing legitimate binaries for malicious activity.
- **Harden the backup infrastructure.** Targeted backup systems to steal credentials and disable recovery. Segment backup networks, enforce the principle of least privilege, rotate admin credentials as needed, and monitor for the suspicious use of administrative tools such as PowerShell or SQL queries interacting with backup credentials. Consider token or account revocation mechanisms to contain compromise if credentials are exposed.
- **Account for detecting BYOVD and cross-platform threats.** BYOVD attacks involve threat actors installing legitimate but vulnerable drivers to escalate privileges, disable security controls, or hide activity. These are risky because they exploit trusted components, often bypassing traditional AV/EDR. Monitor for unsigned or unexpected driver loads, DLL sideloading, and Linux binary execution on Windows through remote tools. Expand detection rules to cover payloads not native to the system.
- **Extend visibility across hybrid environments.** Ensure that the organization's EDR and SOC playbooks include both Windows and Linux telemetry and actively monitor internal lateral movement to detect early stages of hybrid ransomware attacks.
- **Protect credentials and access tokens.** Apply phishing-resistant MFA, strengthen conditional access policies, and monitor for abnormal use of privileged accounts or tokens.

Proactive Security with Trend Vision One™

Trend Vision One™ is the only AI-powered enterprise cybersecurity platform that centralizes cyber risk exposure management and security operations, delivering robust layered protection across on-premises, hybrid, and multi-cloud environments.

Trend Vision One Threat Intelligence

To stay ahead of evolving threats, Trend customers can access Trend Vision One™ Threat Insights which provides the latest insights from Trend™ Research on emerging threats and threat actors.

Trend Vision One Threat Insights

Threat Actors: Water Galura

Emerging Threats: Agenda Ransomware Deploys Linux Variant on Windows Systems through Remote Management Tools and BYOVD Techniques

Trend Vision One Intelligence Reports (IOC Sweeping)

Agenda Ransomware Deploys Linux Variant on Windows Systems through Remote Management Tools and BYOVD Techniques

Hunting Queries

Trend Vision One Search App

Trend Vision One customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post with data in their environment.

Outbound Connection to Suspicious Remote Host on Port 4396 - Agenda Ransomware

eventSubId: 204 AND dst: 146.70.104.163 AND dpt: 4396 AND LogType: detection

More hunting queries are available for Trend Vision One customers with Threat Insights entitlement enabled.

Indicators of Compromise

Indictors of compromise can be found here.

Tags

Latest News | Ransomware | Research | Articles, News, Reports