

VerdantBamboo: Just Another BRICKSTORM in the Firewall

 volexity.com/blog/2026/06/04/verdantbamboo-just-another-brickstorm-in-the-firewall

Steven Adair

June 4, 2026



In September 2025, Volexity conducted an incident response engagement that began after suspicious network traffic was observed from a Linux-based virtual machine appliance on a customer's network. The virtual machine was an [Egnyte Storage Sync](#) system, which is designed to facilitate syncing local on-premise files with the cloud. Volexity discovered that instead of connecting to a domain affiliated with Egnyte, the appliance was connecting to a threat-actor-controlled domain behind Cloudflare IP addresses. The appliance was also making TLS connections to one of Google's public DNS servers (8.8.8.8). It appeared to be using Google to perform queries via DNS over HTTPS, as there was no DNS activity for the domain observed in the connections. Later in the investigation, this was confirmed to be the case after Volexity obtained snapshots of the Storage Sync system for analysis.

Ultimately, Volexity's analysis revealed that a Chinese threat actor, which Volexity tracks as VerdantBamboo (WARP PANDA, UNC5221), had compromised the Storage Sync system. The suspicious connections observed were the result of a malware implant known as **BRICKSTORM** (described in public reporting, including posts by [CISA](#), [Google Cloud](#) and [NVISO](#)).

Volatility's analysis also revealed that the appliance had periodically been accessed by VerdantBamboo via IP addresses assigned through the victim organization's web SSL VPN. The initial findings determined that the threat actor used the malware's proxying capabilities deployed on the Storage Sync system, along with compromised credentials, to access the victim's Microsoft 365 (M365) environment. Volatility assesses with high confidence that this was done to blend in with legitimate network traffic and evade [Conditional Access policies](#) that would have otherwise prevented access. Further investigation uncovered a much longer timeline: the initial compromise had actually occurred at least 18 months earlier.

After Volatility completed the initial remediation effort, VerdantBamboo returned to once again breach the victim organization by connecting with stolen administrative credentials to the victim's firewall. VerdantBamboo used that access to enable and configure web SSL VPN access to that device. That access was then used to further connect to systems internally and deploy additional custom malware to a Synology NAS appliance.

During the course of the investigation, Volatility determined that VerdantBamboo had compromised the victim organization's Managed Services Provider (MSP). Working closely with the MSP, Volatility found suspect traffic emanating from the MSP's network as well. After a brief investigation of the MSP's network, Volatility found that the MSP's [pfSense](#) firewall had been compromised by VerdantBamboo with a BSD variant of BRICKSTORM. Volatility concluded that this firewall, like the victim organization's Storage Sync system, had also been compromised at least 18 months earlier. Volatility assesses with medium confidence that the victim organization had been compromised by way of VerdantBamboo's breach of the MSP.

Forensic investigation of the Storage Sync system and the pfSense firewall resulted in the discovery of multiple samples of the BRICKSTORM backdoor. Volatility would later find a BRICKSTORM sample on the victim organization's network on a virtual machine that had been retired. This sample was installed on a legacy Linux-based GroupWise server used for archived email messages, which had since been fully shutdown but was still accessible for the investigation.

While BRICKSTORM was the primary implant used by VerdantBamboo, Volatility identified two previously undocumented (at the time of discovery) malware families:

- **PLENET**, a malware family written in .NET Core and compiled to native code using the [Native AOT](#) features added in .NET 7. The analyzed sample was written for Linux target systems. This malware was referred to as "GRIMBOLT" by Google Cloud.

- **AGENTPSD**, a malware family written in Python that was deployed to Linux systems. This malware was compiled to a binary using [PyInstaller](#). It has limited functionality, and Volexity assesses with high confidence that it served as a fallback should VerdantBamboo's primary backdoor no longer function.

This blog post describes the attack chain and an overview of the capabilities of AGENTPSD, BRICKSTORM, and PLENET.

Incident Details

The initial incident was discovered when a Linux virtual machine running Egnyte Storage Sync system was found to have been compromised by VerdantBamboo. Volexity's investigation determined that VerdantBamboo was able to access the Storage Sync system using valid credentials via secure shell (SSH) with an unprivileged account named egnyteservice. Volexity found that the threat actor periodically connected to the system and used its access to deploy two backdoors:

- **AGENTPSD**, a basic reverse shell written in Python and packaged with PyInstaller.
- **BRICKSTORM**, a Golang-based remote access trojan (RAT).

During its investigation, Volexity identified that the source IP address of the SSH connections were those assigned by the organization's web-based SSL VPN. The victim organization's MSP managed the Storage Sync system, but it did so through the web interface and not via SSH. The egnyteservice account is the default account on the Storage Sync virtual machine. While it comes with a default password, Volexity confirmed the password had been changed by the MSP, indicating that the threat actor must have gained access to these credentials from the victim organization's MSP.

BRICKSTORM was the initial malware implant that Volexity discovered on the Storage Sync system. It was found in the `/usr/sbin/` directory which is not a path the egnyteservice account should be able write to, as it requires root-level permission. However, Volexity uncovered evidence that VerdantBamboo had discovered the settings for the account's sudo configuration, which included an **inadvertent local privilege escalation**. The following lines are from sudo configuration on the analyzed Storage Sync system:

```
Cmnd_Alias EGNYTEAPPS = /usr/local/bin/egnyte/rsync_data_migration.sh,  
/usr/bin/tee, /usr/bin/config_network,  
/usr/local/bin/egnyte/config_network, /sbin/reboot, /sbin/poweroff ,  
/usr/bin/systemctl restart networking  
egnyteservice ALL = PASSWD: EGNYTEAPPS
```

This configuration allows the egnyteservice account to run the tee command as root via sudo, allowing the threat actor to arbitrarily write files to anywhere on the file system. According to the system logs, VerdantBamboo created a file in /etc/cron.d/ named ssync that would execute /home/egnyteservice/ssync.sh. The attacker would then remove their file from /etc/cron.d/ once executed; the contents of ssync.sh simply executed the backdoor that VerdantBamboo had written into the /usr/sbin/ directory. After this operation was successful, the threat actor removed the file from /etc/cron.d, meaning there was no long-term persistence method for this implant. Instead, VerdantBamboo would login and manually launch BRICKSTORM via this method, when required.

Volexity discovered that VerdantBamboo had set up a fallback command-and-control (C2) channel on the Storage Sync system in the event that BRICKSTORM was no longer running and the threat actor could not access it via SSH. To do this, VerdantBamboo modified the file /etc/crontab and added the following entry:

```
20      14      15      *      *      root
/usr/local/bin/egnyte/egnyte_host_monitor_client >> /dev/null 2>&1
```

This entry would result in the file egnyte_host_monitor_client being executed as root at 14:20 on the 15th day of every month. This file is a simple Python reverse shell that Volexity tracks as **AGENTPSD**. It was configured to attempt to connect back to a domain that was not the one found in the BRICKSTORM malware implant. Both malware implants had been on the Storage Sync system for at least 18 months prior to Volexity's investigation. Based on Volexity's analysis, AGENTPSD was never used, which makes sense since BRICKSTORM continued to be available to the threat actor.

While the Storage Sync server was isolated and taken offline, Volexity determined that VerdantBamboo had been accessing the victim organization via their web-based SSL VPN, however, logs did not go back far enough to determine how that had been accessed. Volexity surmises that the threat actor used compromised local accounts that did not require multi-factor authentication (MFA) to obtain access to the network. The MSP retired the SSL VPN device at the same time the Storage Sync server was taken offline.

Volexity reported the local privilege escalation vulnerability to Egnyte and it was subsequently fixed in [Storage Sync v13.13](#).

All Eyes on the MSP

During the investigation, Volexity suspected that the victim organization's MSP may have been compromised. Volexity worked with the MSP to look into systems on the their network,

and Volexity quickly zeroed in on the MSP's pfSense firewall. pfSense is a popular open-source firewall that runs on the FreeBSD operating system.

Analysis of the pfSense firewall revealed that it had been compromised by **multiple threat actors**. Volexity found evidence of web shells, cryptocurrency miners, and alternate VPN configurations. Most notably, however, Volexity discovered that VerdantBamboo had breached the firewall and deployed a FreeBSD-compatible BRICKSTORM implant. This implant beacons back to a different domain than those identified in Volexity's investigation of the victim organization's compromise. Volexity found that VerdantBamboo had set up persistence for the BRICKSTORM implant by modifying the file `/etc/rc.d/cron` to include a single line to execute the implant. The attackers had placed their backdoor in the `/usr/local/libexec/ipsec/` directory and named it `blacklist`. The firewall also showed signs of compromise by VerdantBamboo going back at least 18 months. The modified cron file is shown below.

```
#!/bin/sh
#
# $FreeBSD$
#

# PROVIDE: cron
# REQUIRE: LOGIN FILESYSTEMS
# BEFORE: securelevel
# KEYWORD: shutdown

. /etc/rc.subr

name="cron"
desc="Daemon to execute scheduled commands"
rcvar="cron_enable"
command="/usr/sbin/${name}"
pidfile="/var/run/${name}.pid"
/usr/local/libexec/ipsec/blacklist
load_rc_config $name
if checkyesno cron_dst
then
    cron_flags="$cron_flags -s"
fi
run_rc_command "$1"
```

Volexity had limited involvement in analyzing additional systems at the MSP. However, it was clear that VerdantBamboo had a root-level compromise on one of the MSP's critical systems. Volexity assesses with high confidence that the threat actor was able to gain access to key

systems leveraged by the MSP and obtained credentials and infrastructure details that could be used to breach the victim organization.

VerdantBamboo Returns

A few days after the victim organization's Storage Sync server and web-based SSL VPN were taken offline, Volexity discovered that VerdantBamboo had somehow regained access to the victim organization's network. A Synology NAS device was actively beaconing back to the same C2 domain found on the Storage Sync server. Volexity obtained access to the Synology system and used [Volexity Surge Collect Pro](#) to collect memory and selective files for analysis. The data collected was then processed with [Volexity Volcano Server](#). Volexity's analysis revealed that VerdantBamboo had connected to the Synology NAS web interface with administrative credentials and used that access to enable SSH to the system. The threat actor then connected over SSH to deploy a previously undocumented backdoor, which Volexity tracks under the name **PLENET**.

Through further investigation, Volexity found the source of the SSH was an IP address that was assigned by the victim organization's firewall. Volexity also discovered that when the MSP took the web-based SSL VPN offline, the victim organization's firewall was then accessible over the internet via a public IP address, and the firewall's administrative interface was exposed.

Since the device's administrative interface was exposed to the Internet, VerdantBamboo was able to connect to it and use stolen administrative credentials that were not protected by MFA. VerdantBamboo then set up a web-based SSL VPN network and connected to it. This access was used to pivot into the victim organization's network again, and deploy PLENET. Interestingly, Volexity found VerdantBamboo had administrative-level access to the victim organization's VMware infrastructure and validated those credentials via web-based logins. However, the threat actor was not observed SSH'ing into, or otherwise installing malware on, any ESXi or vCenter systems, [despite public reporting indicating this is something the threat actor frequently does](#).

Based on the evidence Volexity uncovered, VerdantBamboo's modus operandi was to target devices that lacked EDR coverage. By deploying malware on these devices, the threat actor was able to establish proxy connections to access the victim organization's M365 environment. This allowed VerdantBamboo to blend in with legitimate traffic and bypass Conditional Access policies put in place by the victim organization that would have otherwise prevented access. In large part, this exploitation was made possible by the lack of security system coverage on these devices.

Malware Analysis

This section details the various malware families identified by Volexity that were used by VerdantBamboo during the investigation.

AGENTPSD

AGENTPSD is a basic reverse shell utility written in Python and packaged into a native binary using PyInstaller. Volexity is not aware of any public description of this malware, or any samples of AGENTPSD uploaded to VirusTotal. File details for the sample analyzed are shown below:

Name	egnyte_host_monitor_client
Size	6.4MB (6692728 Bytes)
File Type	ELF Executable
MD5	98ee964edeb5a988c3bba8ea1e57fe0e
SHA1	e952c18272efa1c3d73d0a5381bcf443c02743fe
SHA256	ee41e06ed96182ce80cd4544a6abd5d7719c4a5c0e5ddb266a83842d39b99b0a

The malware first decodes an embedded C2 server address. The encoding uses standard Base64 but ignores the first character of the buffer. Volexity notes that the same encoding style of Base64 on a substring'd variable was also observed in webshells attributed to VerdantBamboo, such as [this one on VirusTotal](#). Once the malware decodes the C2 address, the sample generates a client identification string using the following format:

```
| root_<hostname>_<hex machine UUID>
```

The malware then enters its command loop. The command request consists of an HTTPS POST request with the following parameters:

Headers

```
| Content-Type:"application/octet-stream"  
| sec-fetch-tag:"1<Base64 encoded sysinfo>"
```

POST data

```
s=web&t=aft&atyp=csi&ei=-qv0YeW-  
J8TEkPIP8Nmb6A8&rt=wsrt.38,aft.4354,afti.4354,prt.710,sct.403&imn=18&ima=  
8&imad=8&aftp=907&bb=1&bl=b2ug
```

The command string is parsed out of the server response by searching for the hardcoded pattern PoRaSGw3jzQ8YSaz. Once the pattern is located, the remainder of the response buffer is Base64 decoded, and the command string is split into individual commands using the newline (\n) character. After each command is executed, the malware sleeps for an interval defined by a global parameter. Likewise, the number of commands executed in a batch is limited by a maximum sleep duration also defined by a global parameter.

There are three supported command types:

Command	Description
interval	Updates the sleep interval
during	Updates the maximum duration
builtin	Executes command text on the native shell

Shell output of the builtin command is sent to the server via an HTTPS POST request with the following parameters:

Headers

Content-Type:"application/octet-stream"
sec-fetch-tag = '2<Base64 encoded sysinfo>'

POST data

'<command text>\r\n<command output>'

Passive DNS data for the C2 domain indicates that it never resolved to an IP address despite being registered at the time of the compromise. Given this, and the relative simplicity of the malware, Volexity assesses with medium confidence that AGENTPSD is intended as a fallback to maintain access to the system should the threat actor's other malware be removed or otherwise cease to function.

BRICKSTORM

BRICKSTORM is an advanced malware implant, with variants written initially in Golang and later in Rust. BRICKSTORM is VerdantBamboo's primary malware for directly interfacing with

compromised devices. This malware has been described or referenced in the public domain several times, [including in a post by Google Cloud in September 2025](#). BRICKSTORM was also referenced in [a guide shared to F5 customers](#) following F5’s October 2025 disclosure of a breach.

Unlike many malware families built with Golang, BRICKSTORM’s malicious functionality is not housed in the main package. Rather, the malware developers created a package named `wssoft`, or `wsshell` in some older variants, containing a variety of protocol and command-handling implementations. As such, the main package is relatively small and serves to initiate and link the required functionality using an asynchronous model based around “tasks,” as they are called by the malware. This construction allows BRICKSTORM developers to customize instances of the malware for specific targets, using only the task extensions needed for the specific operation. This design explains why prior reports from [Mandiant](#) and [NVISO](#) describe differing capability sets for samples of BRICKSTORM.

The `wssoft` package namespace is split internally into `core` and `libs`. The `wssoft.libs` sub-namespace consists of functions and utilities for local operations, such as file system and process activities, as well as a client for shell command execution. The `wssoft.core` sub-namespace, in turn, contains protocol handlers for interacting with C2, the task dispatcher implementation, and any extension tasks needed for the sample’s malicious behavior.

BRICKSTORM was also discovered by Volexity on the initially analyzed Egnyte Storage Sync system. The active sample was located in the `/usr/sbin/` directory. Details for this file are provided below:

Name	<code>luserput (sbin)</code>
Size	5.6MB (5844992 Bytes)
File Type	ELF Executable
MD5	58d4eccc982c9e9b1b98aa62c514e53a
SHA1	f4d77958a12a0778283d3e679b24b18f82e332c4
SHA256	40d264cf9c73923932c3dfd52d20f46ff602be3fea8dc6ecc71aca46e6067bf5

These BRICKSTORM instances use the websocket protocol handler for connecting to the C2. They contain three core task extensions:

Internal Name	Description
command	A client for remote shell command execution
socks	A Socks5 proxy server implementation
web	A web server for surfacing filesystem information

The BRICKSTORM sample found on the MSP's pfSense firewall appliance is notable as it is a FreeBSD compatible variant of the backdoor and protected with the [gobfuscate](#) binary obfuscator. Details for this file are shown below:

Name	blacklist
Size	5.6MB (5874528 Bytes)
File Type	ELF Executable
MD5	84ad78b2bab946c3677fdc28ebd8a774
SHA1	681075027553546c119ec447eb8df84633dcffce
SHA256	f70abe93112637d3ec2f6c5e058ccac0307ebf63e496f38588cbfc17a8f8a264

Using program analysis tooling, Volatility was able to deobfuscate the majority of the wssoft namespace, as well as namespaces of various public packages from GitHub. Once deobfuscated, Volatility was able to verify that the same three task extensions from the previously discussed samples were present, and these extensions were initiated in the same manner.

PLENET

PLENET is a cross-platform backdoor developed in .NET Core and compiled into a standalone binary using [Native AOT](#). The use of this uncommon runtime significantly complicated sample analysis. Native AOT is relatively new, only being added to the .NET specification in November 2022. As such, tooling to assist in malware analysis for files created using this runtime is immature. The malware developer is likely aware of this and likely selected .NET Core to provide additional obfuscation. Details for the PLENET sample analyzed are shown below:

Name	ovs-dbctl
-------------	-----------

Size	2.5MB (2635736 Bytes)
File Type	ELF Executable
MD5	95dc2289427ed29b8b996d0e3d1b78cb
SHA1	f8d93c1769e877aae7e7d5c289a467b5ae371c7a
SHA256	eb141a43958802727a6c813452450c10b92704bea4474ee5fd87c0a1be326e2e

This sample is packed with UPX. Once unpacked, the first obstacle was Native AOT's data "hydration" feature. To save space in the compiled binary, static data, such as object metadata and hardcoded strings, are "dehydrated" into a compressed block. This block is then decompressed at runtime into the "hydrated" program segment. The process of mapping the hydrated data was achieved using the [ghidra-nativeaot](#) plugin.

PLENET demonstrates similar design patterns to BRICKSTORM. Like BRICKSTORM, PLENET C2 traffic uses the WebSocket protocol and makes use of a multiplexing library ([Nerdbank.Streams](#)) to manage simultaneous streams to the server. Likewise, PLENET makes use of the asynchronous functionality of the .NET runtime via the native [Task](#) object. Despite these similarities, PLENET uses a more conventional command processing protocol than BRICKSTORM, consisting of a single command dispatcher.

Volexity has verified the following capabilities for PLENET:

- Interactive shell
- Remote command execution
- File manipulation
- C2 server switching

VerdantBamboo Infrastructure Goes Dark

During the incident response investigation, Volexity sought ways to proactively identify infrastructure related to VerdantBamboo. Although the domains used for BRICKSTORM C2 use Cloudflare, their resolving IP addresses could be discovered through certificates presented in Censys data. On September 6, 2025, analyzing common features of these resolving IP addresses led to the development of the following Censys Platform query to fingerprint BRICKSTORM C2 servers:

```
host.service_count<= 4 AND host.services:  
(banner_hash_sha256:"e28a96f983b8605decd2ac1db16ebad5fa741a6aa4e585a38ade
```

```
0e5ad7d6cec0" AND port=443) AND  
host.services.cert.parsed.issuer.organization = "CloudFlare, Inc." AND  
host.services: (port=22 AND software.vendor:openbsd)
```

This search keys on the following:

- A minimalistic zero-length HTTP response (likely a Golang HTTP server)
- An OpenBSD-based SSH client
- A Cloudflare certificate
- A minimal number of services (four or fewer)

While this query did not yield a 100-percent true-positive ratio, it did lead to the discovery of a number of C2 IP addresses and related domains. Between the September 18 and September 23, all of the servers previously matching this pattern turned off their services on port 443. On September 24, 2025, Google Cloud published [a blog post on BRICKSTORM activity](#). While it is possible this timing is coincidental, Volexity assesses with low confidence that VerdantBamboo may have been aware their activities were being investigated, and thus sought to change their TTPs to avoid ongoing detection.

Conclusion

VerdantBamboo is a highly sophisticated threat actor that seeks to leverage a combination of living-off-the-land techniques and malware deployment on systems that traditionally do not or cannot run EDR software. This threat actor appears to have good knowledge of proprietary appliances, allowing them to deploy malware with customized persistence mechanisms. They also appear to have operational security discipline aimed at leveraging a limited number of domains and IP addresses per victim and setting up customized implant naming and persistence on a per-device basis.

The BRICKSTORM malware family is publicly documented, with variants dedicated to various appliances, including vSphere infrastructure. In this blog post, Volexity documented this malware as well as other previously undocumented malware, PLENET and AGENTPSD, that were all deployed to various Linux-based and BSD-based appliances. Notably, PLENET is compiled with .NET Native AOT, resulting in a large static binary with the entire .NET Core and all its dependencies embedded in one file. Malware written in uncommon programming languages can be challenging from a reverse-engineering perspective. Volexity assesses with medium confidence that VerdantBamboo chose this framework to hinder malware analysis.

Finally, VerdantBamboo's intrusion was notable for the amount of time the threat actor was able to maintain persistence on compromised devices on the network edge. All malware used by VerdantBamboo was deployed to proprietary appliances, such as firewalls and other Linux-

based VM appliances (Synology, Egnyte). These appliances generally do not allow the owner to have root access, and monitoring these appliances to enable the detection of a breach is complicated (and sometimes impossible, depending on the device). Volexity assesses with high confidence that the threat actor chose to deploy malware these types of devices specifically to remain undetected as long as possible.

Indicators associated with these campaigns are available [here](#).

Acknowledgments

Volexity would like to thank its customer and partners for allowing us to publish the details of their incident response efforts. Further, Volexity would like to recognize Egnyte for their rapid response to Volexity's reporting of the privilege escalation issue. Their response was a model for how vendors should handle security disclosure from initial contact to resolution.

Investigative Assistance

If any organization or individual believes they may have been targeted by a similar attack, please feel free to reach out to Volexity via our [contact form](#). We would be glad to assess any potential targeting and assist in determining if such an attack may have succeeded.

Volexity's Threat Intelligence research, such as the content from this blog, is published to customers via its [Threat Intelligence Service](#). The activity described in this blog post was shared with Volexity Threat Intelligence customers in TIB-20251104, ANU-20260127 and MAR-20260224.

If you are interested in learning more about Volexity's services, including [Network Security Monitoring](#) and [Incident Response](#), or our leading memory forensics solutions, [Volexity Surge Collect Pro](#) for memory acquisition and [Volexity Volcano](#) for memory analysis, please do not hesitate to [contact us](#).

KEY TAKEAWAYS

- VerdantBamboo compromised firewalls, cloud storage sync systems, and NAS devices with customized malware implants to maintain persistence to victim networks while avoiding systems that would be subject to EDR controls.
- During its incident response engagement, Volexity discovered that VerdantBamboo had also breached the victim organization's Managed Services Provider (MSP), which may have served as the source of the victim organization's compromise.

- Once evicted from the victim organization's network, VerdantBamboo was able to regain administrative access to the victim organization's firewall, and then configured a custom VPN network and connected into it.
- VerdantBamboo was observed deploying PLENET (aka GRIMBOLT) on a Linux-based NAS device; this malware is an advanced backdoor built using the Native AOT features of .NET.
- A new basic Python backdoor, which Volexity calls AGENTPSD, was also deployed by VerdantBamboo as a fallback mechanism to maintain persistence on a compromised system in the event BRICKSTORM ceased to function.