Another BRICKSTORM: Stealthy Backdoor Enabling Espionage into Tech and Legal Sectors

By Mandiant, Google Threat Intelligence Group

Published: 2025-09-24 · Archived: 2026-04-05 17:16:46 UTC

Written by: Sarah Yoder, John Wolfram, Ashley Pearson, Doug Bienstock, Josh Madeley, Josh Murchie, Brad Slaybaugh, Matt Lin, Geoff Carstairs, Austin Larsen

---

Introduction

Google Threat Intelligence Group (GTIG) is tracking BRICKSTORM malware activity, which is being used to maintain persistent access to victim organizations in the United States. Since March 2025, Mandiant Consulting has responded to intrusions across a range of industry verticals, most notably legal services, Software as a Service (SaaS) providers, Business Process Outsourcers (BPOs), and Technology. The value of these targets extends beyond typical espionage missions, potentially providing data to feed development of zero-days and establishing pivot points for broader access to downstream victims.

We attribute this activity to UNC5221 and closely related, suspected China-nexus threat clusters that employ sophisticated capabilities, including the exploitation of zero-day vulnerabilities targeting network appliances. While UNC5221 has been used synonymously with the actor publicly reported as Silk Typhoon, GTIG does not currently consider the two clusters to be the same.

These intrusions are conducted with a particular focus on maintaining long-term stealthy access by deploying backdoors on appliances that do not support traditional endpoint detection and response (EDR) tools. The actor employs methods for lateral movement and data theft that generate minimal to no security telemetry. This, coupled with modifications to the BRICKSTORM backdoor, has enabled them to remain undetected in victim environments for 393 days, on average. Mandiant strongly encourages organizations to reevaluate their threat model for appliances and conduct hunt exercises for this highly evasive actor. We are sharing an updated threat actor lifecycle for BRICKSTORM associated intrusions, along with specific and actionable steps organizations should take to hunt for and protect themselves from this activity.

Figure 1: BRICKSTORM targeting

Threat Actor Lifecycle

The actor behind BRICKSTORM employs sophisticated techniques to maintain persistence and minimize the visibility traditional security tools have into their activities. The section is a review of techniques observed from multiple Mandiant investigations, with customer details sanitized.

Initial Access

A consistent challenge across Mandiant investigations into BRICKSTORM intrusions has been determining the initial intrusion vector. In many cases, the average dwell time of 393 days exceeded log retention periods and the artifacts of the initial intrusion were no longer available. Despite these challenges, a pattern in the available evidence points to the actor's focus on compromising perimeter and remote access infrastructure.

In at least one case, the actor gained access by exploiting a zero-day vulnerability. Mandiant has identified evidence of this actor operating from several other edge appliances early in the lifecycle, but could not find definitive evidence of vulnerability exploitation. As noted in our previous blog post from April 2025, Mandiant has identified the use of post-exploitation scripts that have included a wide range of anti-forensics functions designed to obscure entry.

Establish Foothold

The primary backdoor used by this actor is BRICKSTORM, as previously discussed by Mandiant and others. BRICKSTORM includes SOCKS proxy functionality and is written in Go, which has wide cross-platform support. This is essential to support the actor’s preference to deploy backdoors on appliance platforms that do not support traditional EDR tools. Mandiant has found evidence of BRICKSTORM on Linux and BSD-based appliances from multiple manufacturers. Although there is evidence of a BRICKSTORM variant for Windows,

Mandiant has not observed it in any investigation. Appliances are often poorly inventoried, not monitored by security teams, and excluded from centralized security logging solutions. While BRICKSTORM has been found on many appliance types, UNC5221 consistently targets VMware vCenter and ESXi hosts. In multiple cases, the threat actor deployed BRICKSTORM to a network appliance prior to pivoting to VMware systems. The actor moved laterally to a vCenter server in the environment using valid credentials, which were likely captured by the malware running on the network appliances.

Our analysis of samples recovered from different victim organizations has found evidence of active development of BRICKSTORM. While the core functionality has remained, some samples are obfuscated using Garble and some carry a new version of the custom `wssoft` library. Mandiant recovered one sample of BRICKSTORM with a “delay” timer built-in that waited for a hard-coded date months in the future before beginning to beacon to the configured command and control domain. Notably, this backdoor was deployed on an internal vCenter server after the victim organization had begun their incident response investigation, demonstrating that the threat actor was actively monitoring and capable of rapidly adapting their tactics to maintain persistence.

As previously reported, BRICKSTORM deployments are often designed to blend in with the target appliance, with the naming convention and even the functionality of the sample being designed to masquerade as legitimate activity. Mandiant has identified samples using Cloudflare Workers and Heroku applications for C2, as well as sslip.io or nip.io to resolve directly to C2 IP addresses. From the set of samples we’ve recovered, there has been no reuse of C2 domains across victims.

Escalate Privileges

At one investigation, Mandiant analyzed a vCenter server and found the threat actor installed a malicious Java Servlet filter for the Apache Tomcat server that runs the web interface for vCenter. A Servlet Filter is code that runs every time the web server receives an HTTP request. Normally, installing a filter requires modifying a configuration file and restarting or reloading the application; however, the actor used a custom dropper that made the modifications entirely in memory, making it very stealthy and negating the need for a restart. The malicious filter, tracked by Mandiant as BRICKSTEAL, runs on HTTP requests to the vCenter web login Uniform Resource Indicators (URIs) `/web/saml2/sso/*`. If present, it decodes the `HTTP Basic` authentication header, which may contain a username and password. Many organizations use Active Directory authentication for vCenter, which means BRICKSTEAL could capture those credentials. Often, users who log in to vCenter have a high level of privilege in the rest of the enterprise. Previously shared hardening guidance for vSphere includes steps that can mitigate the ability of BRICKSTEAL to capture usable credentials in this scenario, such as enforcement of multi-factor authentication (MFA).

VMware vCenter is an attractive target for threat actors because it acts as the management layer for the vSphere virtualization platform and can take actions on VMs such as creating, snapshotting, and cloning. In at least two cases, the threat actor used their access to vCenter to clone Windows Server VMs for key systems such as Domain Controllers, SSO Identity Providers, and secret vaults. This is a technique that other threat actors have used. With a clone of the virtual machine, the threat actor can mount the filesystem and extract files of interest, such as the Active Directory Domain Services database (`ntds.dit`). Although these Windows Servers likely have security tools installed on them, the threat actor never powers on the clone so the tools are not executed. The following

example shows vCenter VPXD logs of the threat actor using the local vSphere Administrator account to clone a VM.

```
 2025-04-01 03:37:40 [vim.event.TaskEvent] [info] [VSPHERE.LOCAL\Administrator] [<vCenter inventory object>] [<u
2025-04-01 03:37:49 [vim.event.VmBeingClonedEvent] [info] [VSPHERE.LOCAL\Administrator] [<vCenter inventory obje
2025-04-01 03:42:07 [vim.event.VmClonedEvent] [info] [VSPHERE.LOCAL\Administrator] [<vCenter inventory object>]
2025-04-01 04:05:40 [vim.event.TaskEvent] [info] [VSPHERE.LOCAL\Administrator] [<vCenter inventory object>] [<ur
2025-04-01 04:05:47 [vim.event.VmRemovedEvent] [info] [VSPHERE.LOCAL\Administrator] [<vCenter inventory object>]
```

In one instance the threat actor used legitimate server administrator credentials to repeatedly move laterally to a system running Delinea (formerly Thycotic) Secret Server. The forensic artifacts recovered from the system were consistent with the execution of a tool, such as secret stealer, to automatically extract and decrypt all credentials stored by the Secret Server application.

Move Laterally

Typically, at least one instance of BRICKSTORM would be the primary source of hands-on keyboard activity, with two or more compromised appliances serving as backups. To install BRICKSTORM, the actor used legitimate credentials to connect to the appliance, often with SSH. In one instance the actor used credentials known to be stored in a password vault they previously accessed. In another instance they used credentials known to be stored in a PowerShell script the threat actor previously viewed. In multiple cases the actor logged in to either the ESXi web-based UI or the vCenter Appliance Management Interface (VAMI) to enable the SSH service so they could connect and install BRICKSTORM. The following are example VAMI access events that show the threat actor connecting to VAMI and making changes to the SSH settings for vCenter.

```
 ::ffff:<Source IP> <vCenter IP>:5480 - [<timestamp>] "GET / HTTP/1.1" 200 1153 "-" "<User Agent>"
::ffff:<Source IP> <vCenter IP>:5480 - [<timestamp>] "POST /rest/com/vmware/cis/session HTTP/1.1" 200 60 "https
::ffff:<Source IP> <vCenter IP>:5480 - [<timestamp>] "PUT /rest/appliance/access/ssh HTTP/1.1" 200 0 "https://1
```

Establish Persistence

To maintain access to victim environments, the threat actor modified the `init.d`, `rc.local`, or `systemd` files to ensure BRICKSTORM started on appliance reboot. In multiple cases, the actor used the `sed` command line utility to modify legitimate startup scripts to launch BRICKSTORM. The following are a few example `sed` commands executed by the actor on vCenter.

```
 sed -i s/export TEXTDOMAIN=vami-lighttp/export TEXTDOMAIN=vami-lighttp\n\/path/to/brickstorm/g /opt/vmware/etc/

sed -i $a\SETCOLOR_WARNING="echo -en `/path/to/brickstorm`\\033[0;33m" /etc/sysconfig/init
```

The threat actor has also created a web shell tracked by Mandiant as SLAYSTYLE on vCenter servers. SLAYSTYLE, tracked by MITRE as BEEFLUSH, is a JavaServer Pages (JSP) web shell that functions as a

backdoor. It is designed to receive and execute arbitrary operating system commands passed through an HTTP request. The output from these commands is returned in the body of the HTTP response.

Complete Mission

A common theme across investigations is the threat actor's interest in the emails of key individuals within the victim organization. To access the email mailboxes of target accounts, the threat actor made use of Microsoft Entra ID Enterprise Applications with `mail.read` or `full_access_as_app` scopes. Both scopes allow the application to access mail in any mailbox. In some cases, the threat actor targeted the mailboxes of developers and system administrators while in other cases, they targeted the mailboxes of individuals involved in matters that align with PRC economic and espionage interests.

When the threat actor exfiltrated files from the victim environment, they used the SOCKS proxy feature of BRICKSTORM to tunnel their workstation and directly access systems and web applications of interest. In multiple cases the threat actor used legitimate credentials to log in to the web interface for internal code stores and download repositories as ZIP archives. In other cases the threat actor browsed to specific directories and files on remote machines by specifying Windows Universal Naming Convention (UNC) paths.

In several cases the BRICKSTORM samples deployed by the threat actor were removed from compromised systems. In these cases, the presence of BRICKSTORM was observed by conducting forensic analysis of backup images that identified the BRICKSTORM malware in place.

Hunting Guidance

Mandiant has previously discussed the diminishing usefulness of atomic IOCs and the need to adopt TTP-based hunting. Across BRICKSTORM investigations we have not observed the reuse of C2 domains or malware samples, which, coupled with high operational security, means these indicators quickly expire or are never observed at all. Therefore, a TTP-based hunting approach is not only an ideal practice, but a necessity to detect patterns of attack that are unlikely to be detected by traditional signature-based defenses. The following is a checklist of the minimal set of hunts Mandiant recommends organizations conduct to search for BRICKSTORM and related activities.

Step	Hunt	Data Sources
0	Create or update asset inventory that includes edge devices and other appliances	N/A
1	File and backup scan for BRICKSTORM	Appliance file system, backups

2	Internet traffic from edge devices and appliances	Firewall connection logs, DNS logs, IDS/IPS, netflow
3	Access to Windows servers and desktops from appliances	EDR telemetry, Security Event Logs, Terminal Service Logs, Windows UAL
4	Access to credentials and secrets	Windows Shellbags, EDR telemetry
5	Access to M365 mailboxes using Enterprise Application	M365 UAL
6	Cloning of sensitive virtual machines	vSphere VPXD logs
7	Creation of local vCenter and ESXi accounts	VMware audit events
8	SSH enablement on vSphere platform	VMware audit events, VAMI logs
9	Rogue VMs	VMware audit events, VM inventory reports

Create or Update Asset Inventory

Foundational to the success of any threat hunt is an asset inventory that includes devices not covered by the standard security tool stack, such as edge devices and other appliances. Because these appliances lack support for traditional security tools an inventory is critical for developing effective compensating controls and detections. Especially important is to track the management interface addresses of these appliances, as they act as the default gateway that malware and threat actor commands will egress out of.

Mandiant recommends organizations take a multi-step approach to building or updating this inventory:

1. **Known knowns:** Begin with the appliance classes that all organizations use: firewalls, VPN concentrators, virtualization platforms, conferencing systems, badging, and file storage.
2. **Known unknowns:** Work across teams to brainstorm appliance classes that may be more specialized to your organization, but the security organization likely lacks visibility into.

3. **Unknown unknowns:** These are the appliances that were supposed to be decommissioned but weren't, sales POVs, and others. Consider using network visibility tools or your existing EDR to scan for "live" IP addresses that do not show in your EDR reports. This has the added benefit of identifying unmanaged devices that should have EDR but don't.

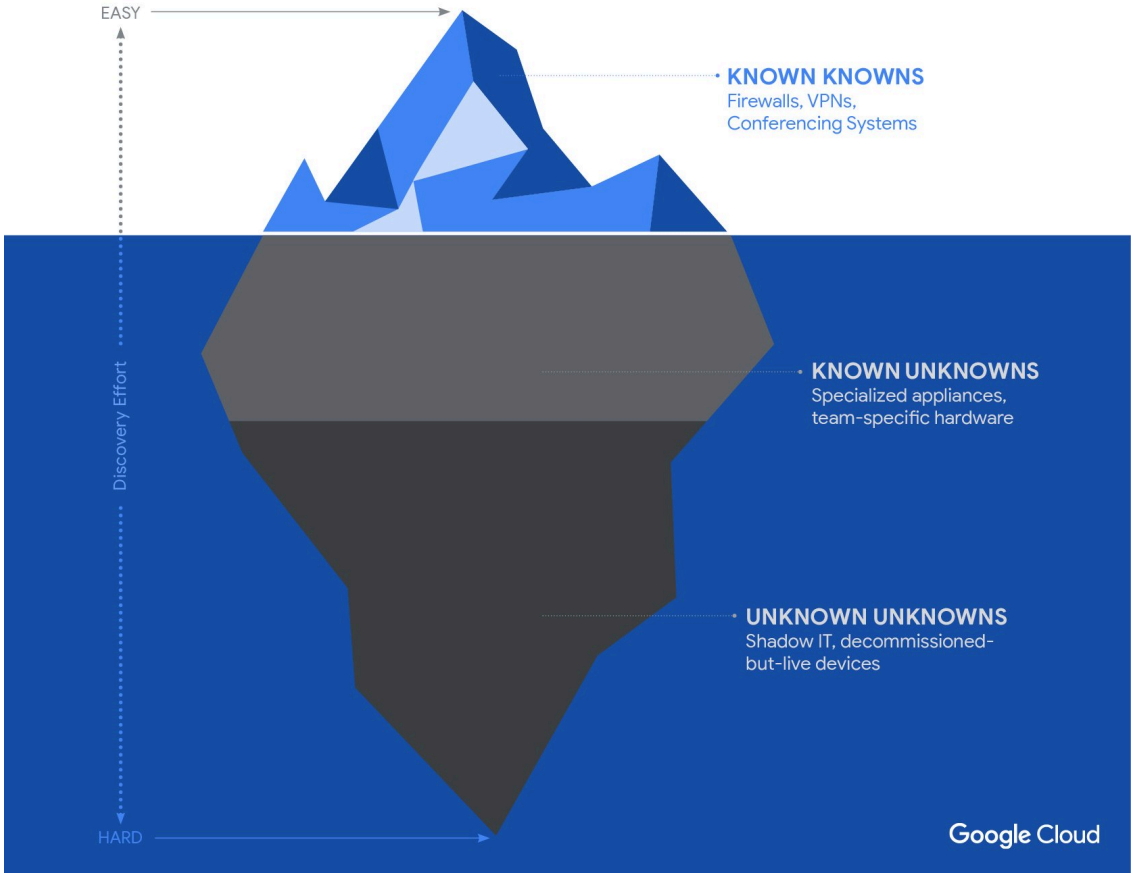

Figure 2: Asset inventory

File and Backup Scan for BRICKSTORM

YARA rules have proven to be the most effective method for detecting BRICKSTORM binaries on appliances. We are sharing relevant YARA rules in the appendix section of this post. Yara can be difficult to run at scale, but some backup solutions provide the ability to run YARA across the backup data store. Mandiant is aware of multiple customers who have identified BRICKSTORM through this method.

To aid organizations in hunting for BRICKSTORM activity in their environments, Mandiant released a scanner script, which can run on appliances and other Linux or BSD-based systems.

Internet Traffic from Edge Devices and Appliances

Use the inventory of appliance management IP addresses to hunt for evidence of malware beaconing in network logs. In general, appliances should not communicate with the public Internet from management IP addresses except to download updates and send crash analytics to the manufacturer.

Established outbound traffic to domains or IP addresses not controlled by the appliance manufacturer should be regarded as very suspicious and warranting forensic review of the appliance. BRICKSTORM can use DNS over HTTP (DoH), which should be similarly rare when sourced from appliance management IP addresses.

Access to Windows Systems from Appliances

The threat actor primarily accessed Windows machines (both desktops and servers) using type 3 (network) logins, although in some cases the actor also established RDP sessions. Appliances should rarely log in to Windows desktops or servers and any connections should be treated as suspicious. Some examples of false positives could include VPN appliances using a known service account to connect to a domain controller in order to perform LDAP lookups and authenticated vulnerability scanners using a well-known service account.

In addition to EDR telemetry, Terminal Services logs and Security event logs, defenders should obtain and parse the Windows User Access Log (UAL). The UAL is stored on Windows Servers inside the directory `Windows\System32\LogFiles\Sum` and can be parsed using open-source tools such as `SumECmd`. This log source records attempted authenticated connections to Windows systems and often retains artifacts going back much longer than typical Windows event logs. Note that this log source includes successful and unsuccessful logins, but is still useful to identify suspicious activity sourced from appliances.

Access to Credentials and Secrets

Use the forensic capabilities of EDR tools to acquire Windows Shellbags artifacts from Windows workstations and servers. Shellbags records folder paths that are browsed by a user with the Windows Explorer application. Use an open-source parser to extract the relevant data and look for patterns of activity that are suspicious:

- Access to folder paths where the initiating user is a service account, especially service accounts that are unfamiliar or rarely used
- File browsing activity sourced from servers that include a Windows Universal Naming Convention (UNC) path that points to a workstation (e.g., `\\bobwin7.corp.local\browsing\path)`
- File browsing activity to folder paths that contain credential data, such as:
    - Browser profile paths (e.g., `%appdata%\Mozilla\Firefox\Profiles`)
    - Appdata locations used to store session tokens (e.g., `Users\<username>\.azure\`)
    - Windows credential vault (`%appdatalocal%\Microsoft\Credentials`)
    - Data Protection API (DPAPI) keys (`%appdata%\Microsoft\Protect\<SID>\`)

Access to M365 Mailboxes using Enterprise Application

Mandiant has observed this actor use common techniques to conduct bulk email access and exfiltration from Microsoft 365 Exchange Online. Organizations should follow our guidance outlined in our APT29 whitepaper to hunt for these techniques. Although the white paper specifically references APT29, these techniques have become

widely used by many groups. In multiple investigations the threat actor used a Microsoft Entra ID Enterprise Application with `mail.read` or `full_access_as_app` scopes to access mailboxes of key individuals in the victim organization.

To hunt for this activity, we recommend a phased approach:

1. Enumerate the Enterprise Applications and Application Registrations with graph permissions that can read all mail.
2. For each application, validate that there is at least one secret or certificate configured for it. Record the Application (client) ID
3. Conduct a free text search against the Unified Audit Log or the `OfficeActivity` table in Sentinel for the client IDs from step 2. This will return the `mailitemsaccessed` events that recorded the application accessing mail.
4. For each application analyze the source IP addresses and user-agent strings for discrepancies. Legitimate usage of the applications should occur from well-defined IP addresses. Additionally, look for focused interest in key personnel mailboxes across multiple days.

When accessing M365 and other internet-facing services the actor has used multiple commercial VPN and proxy providers. Mandiant has found evidence of the threat actor using PIA, NordVPN, Surfshark, VPN Unlimited, and PrivadoVPN, although there is no reason for these to be the only solutions used. There is also evidence to support that this actor has access to a purpose-built obfuscation network built from compromised small office/home office routers. Mandiant has no knowledge of how these routers are being compromised. The exit nodes for commercial VPNs and obfuscation networks change rapidly and sharing atomic indicators for hunting purposes is unlikely to yield results. Instead, identify the key individuals in the organization, with respect to the organization vertical and likely goals of the threat actor. Fetch `mailitemsaccessed` logs for those mailboxes for the last year or as long as retention allows. Analyze the `SessionID` values of the log events and look for IDs that span multiple IP addresses where the IP addresses are not in the user's typical geographic location.

Cloning of Sensitive Virtual Machines

On VMware vCenter servers, VPXD logs contain valuable information for VM management related tasks such as clone events, powering on and off a VM, and creating snapshots. The threat actor often used the `VSPHERE.LOCAL\Administrator` account when cloning VMs and targeted VMs that would contain credentials such as password vaults and domain controllers. The threat actor would delete the cloned VM shortly after cloning, and primarily operated between the hours of 01:00 and 10:00 UTC. Investigators should search vCenter VPXD logs for activity that matches the aforementioned criteria and confirm if the cloning activity was intended or not.

Creation of Local vCenter and ESXi Accounts

Mandiant identified evidence the threat actor created a new local account to install BRICKSTORM and then removed the account after they were done. The following logs show the threat actor using the local Administrator

account to create a new local account and add it to the `BashShellAdministrators` group. The threat actor established an SSH connection from a compromised appliance to vCenter using the newly created account and installed the BRICKSTORM backdoor on vCenter. Shortly after, the threat actor deleted the account. Investigators should review audit logs in `/var/log/audit/sso-events/audit_events.log` for the creation and deletion of local accounts and validate their purpose. In one instance, the threat actor named the account with a similar naming convention as a local service account used for backups on vCenter.

```
 2025-04-01T06:45:32 {"user":"Administrator@VSPHERE.LOCAL","client":"","timestamp":"04/01/2025 06:45:32 GMT","de

2025-04-01T06:45:55 {"user":"Administrator@VSPHERE.LOCAL","client":"","timestamp":"04/01/2025 06:45:55 GMT","des

2025-04-01T06:46:23 {"user":"Administrator@VSPHERE.LOCAL","client":"","timestamp":"04/01/2025 06:46:23 GMT","des

2025-04-01T06:52:03 <vcenter_hostname> sshd[36952]: Postponed keyboard-interactive/pam for <account_name>@vspher

2025-04-01T06:52:30 <vcenter_hostname> sudo: pam_unix(sudo:session): session opened for user root

2025-04-01T06:53:39 Creation of BRICKSTORM on vCenter

2025-04-01T06:56:18 <vcenter_hostname> sudo: pam_unix(sudo:session): session closed for user root

2025-04-01T06:56:25 <vcenter_hostname> sshd[36952]: pam_unix(sshd:session): session closed for user <account_nar

2025-04-01T06:56:57 {"user":"Administrator@VSPHERE.LOCAL","client":"","timestamp":"04/01/2025 06:56:57 GMT","des

2025-04-01T06:58:12 {"user":"Administrator@VSPHERE.LOCAL","client":"","timestamp":"04/01/2025 06:58:12 GMT","des
```

SSH Enablement on ESXi and vCenter

For ESXi servers, monitoring should be set up for SSH logins using local accounts. In most organizations it is relatively rare for legitimate direct access to the ESXi hosts over SSH. In many cases the SSH server is disabled by default. Write rules to alert on log events when SSH is enabled for a vSphere platform appliance.

Rogue VMs

Organizations should review VMWare audit events that track the creation and deletion of new VMs, particularly using non-standard ISO images and Operating Systems. Audit events may also record the threat actor downloading archived ISO images to the datastore volumes used by vSphere.

Hardening Guidance

It is crucial to maintain an up-to-date inventory of appliances and other devices in the network that do not support the standard security tool stack. Any device in that inventory, whether internal or internet-facing, should be configured to follow a principle of least access.

- Internet access: Appliances should not have unrestricted access to the internet. Work with your vendors or monitor your firewall logs to lock down internet access to only those domains or IP addresses that the appliance requires to function properly.
- Internal network access: Appliances exposed to the internet should not have unrestricted access to internal IP address space. The management interface of most appliances does not need to establish connections to internal IP addresses. Work with the vendor to understand specific needsLDAP queries to verify user attributes for VPN logins.

Mandiant has previously published guidance to secure the vSphere platform from threat actors. We recommend you follow the guidance, especially the forwarding of logs to a central SIEM, enabling vSphere lockdown mode, enforcing MFA for web logins, and enforcing the `execInstalledOnly` policy.

Organizations should assess and improve the isolation of any credential vaulting systems. In many cases if a threat actor is able to gain access to the underlying Operating System, any protected secrets can be exposed. Servers hosting credential vaulting applications should be considered Tier 0 systems and have strict access controls applied to them. Mandiant recommends organizations work with their vendors to adopt secure software practices such as storing encryption keys in the Trusted Platform Module (TPM) of the server.

Outlook and Implications

Recent intrusion operations tied to BRICKSTORM likely represent an array of objectives ranging from geopolitical espionage, access operations, and intellectual property (IP) theft to enable exploit development. Based on evidence from recent investigations the targeting of the US legal space is primarily to gather information related to US national security and international trade. Additionally, GTIG assesses with high confidence that the objective of BRICKSTORM targeting SaaS providers is to gain access to downstream customer environments or the data SaaS providers host on their customers' behalf. The targeting of technology companies presents an opportunity to conduct theft of valuable IP to further the development of zero-day exploits.

Acknowledgements

This analysis would not have been possible without the assistance from across Google Threat Intelligence Group, Mandiant Consulting and FLARE. We would like to specifically thank Nick Simonian from GTIG Research and Discovery (RAD). We would also like to thank Ryan Tomcik from Mandiant Threat Defense (MTD) for contributing network detection content.

Indicators of Compromise

The following indicators of compromise are available in a Google Threat Intelligence (GTI) collection. Note that Mandiant has not observed instances where the threat actor reused a malware sample and hunting for the exact indicators is unlikely to yield results.

SHA-256 Hash	File Name	Description

`90b760ed1d0dcb3ef0f2b6d6195c9d852bcb65eca293578982a8c4b64f51b035`	pg_update	BRICKSTORM
`2388ed7aee0b6b392778e8f9e98871c06499f476c9e7eae6ca0916f827fe65df`	spclisten	BRICKSTORM
`aa688682d44f0c6b0ed7f30b981a609100107f2d414a3a6e5808671b112d1878`	vmp	BRICKSTORM

YARA Detections

G_APT_Backdoor_BRICKSTORM_3

```
 rule G_APT_Backdoor_BRICKSTORM_3 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
strings:
                $str1 = { 48 8B 05 ?? ?? ?? ?? 48 89 04 24 E8 ?? ?? ?? ?? 48 B8 ?? ?? ?? ?? ?? ?? ?? ?? 48 89
                $str2 = "regex" ascii wide nocase
                $str3 = "mime" ascii wide nocase
                $str4 = "decompress" ascii wide nocase
                $str5 = "MIMEHeader" ascii wide nocase
                $str6 = "ResolveReference" ascii wide nocase
                $str7 = "1157920892103562487626974469494075735299969552241357603424222590610685120443691157920
        condition:
                uint16(0) == 0x457F and all of them
}
```

G_Backdoor_BRICKSTORM_2

```
 rule G_Backdoor_BRICKSTORM_2 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $obf_func = /[a-z]{20}\/[a-z]{20}\/[a-z]{20}\/[a-z]{20}.go/
                $decr1 = { 0F B6 4C 04 ?? 0F B6 54 04 ?? 31 D1 88 4C 04 ?? 48 FF C0 [0-4] 48 83 F8 ?? 7C }
                $decr2 = { 40 88 7C 34 34 48 FF C3 48 FF C6 48 39 D6 7D 18 0F B6 3B 48 39 CE 73 63 44 0F B6 04
                $decr3 = { 0F B6 54 0C ?? 0F B6 5C 0C ?? 31 DA 88 14 08 48 FF C1 48 83 F9 ?? 7C E8 }
                $str1 = "main.selfWatcher"
                $str2 = "main.copyFile"
                $str3 = "main.startNew"
                $str4 = "WRITE_LOG=true"
                $str5 = "WRITE_LOGWednesday"
                $str6 = "vami-httpdvideo/webm"
```

```
                $str7 = "/opt/vmware/sbin/"
                $str8 = "/home/vsphere-ui/"
                $str9 = "/opt/vmware/sbin/vami-http"
                $str10 = "main.getVFromEnv"
        condition:
                uint32(0) == 0x464c457f and ((any of ($decr*) and $obf_func) or (any of ($decr*) and any of ($
}
```

G_APT_Backdoor_BRICKSTORM_1

```
rule G_APT_Backdoor_BRICKSTORM_1 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $ = "WRITE_LOGWednesday"
                $ = "/home/vsphere-ui/"
                $ = "WRITE_LOG=true"
                $ = "dns rcode: %v"
                $ = "dns query not specified or too small"
                $ = "/dev/pts: bad file descriptor"
                $ = "/libs/doh.Query"
                $ = "/libs/doh.createDnsMessage"
                $ = "/libs/doh.unpackDnsMessage"
                $ = "/core/protocol/websocket.(*WebSocketNetConfig).Dial"
                $ = "/core/protocol/websocket.(*connection).Read"
                $ = "/core/protocol/websocket.(*connection).getReader"
                $ = "/core/protocol/websocket.(*connection).Write"
                $ = "/core/protocol/websocket.(*connection).Close"
                $ = "/core/protocol/websocket.(*connection).LocalAddr"
                $ = "/core/protocol/websocket.(*connection).RemoteAddr"
                $ = "/core/protocol/websocket.(*connection).SetDeadline"
                $ = "/core/protocol/websocket.(*connection).SetReadDeadline"
                $ = "/core/protocol/websocket.(*connection).SetWriteDeadline"
                $ = "/core/protocol.UnPackHeaderData"
                $ = "/core/protocol.NewWebSocketClient"
                $ = "/libs/func1.(*Client).BackgroundRun"
                $ = "/libs/func1.CreateClient"
                $ = "/libs/func1.NewService"
                $ = "/libs/func1.(*Service).Get"
                $ = "/libs/func1.(*Service).DoTask"
                $ = "/libs/func1.(*Service).Put"
                $ = "/core/extends/command.Command"
                $ = "/core/extends/command.CommandNoContext"
                $ = "/core/extends/command.ExecuteCmd"
                $ = "/core/extends/command.RunShell"
                $ = "/core/extends/socks.UnPackHeaderData"
```

```
                $ = "/core/extends/socks.handleRelay"
                $ = "/libs/fs.(*RemoteDriver).realPath"
                $ = "/libs/fs.(*RemoteDriver).ChangeDir"
                $ = "/libs/fs.(*RemoteDriver).Stat"
                $ = "/libs/fs.(*SimplePerm).GetMode"
                $ = "/libs/fs.(*SimplePerm).GetOwner"
                $ = "/libs/fs.(*SimplePerm).GetGroup"
                $ = "/libs/fs.(*RemoteDriver).ListDir"
                $ = "/libs/fs.(*RemoteDriver).DeleteDir"
                $ = "/libs/fs.(*RemoteDriver).DeleteFile"
                $ = "/libs/fs.(*RemoteDriver).Rename"
                $ = "/libs/fs.(*RemoteDriver).MakeDir"
                $ = "/libs/fs.(*RemoteDriver).GetFile"
                $ = "/libs/fs.(*RemoteDriver).PutFile"
                $ = "/libs/fs.(*RemoteDriver).UpFile"
                $ = "/libs/fs.(*RemoteDriver).MD5"
                $ = "/libs/doh/doh.go"
                $ = "/core/protocol/websocket/config.go"
                $ = "/core/extends/command/command.go"
                $ = "/libs/fs/driver_unix.go"
                $ = "/libs/fs/perm_linux.go"
        condition:
                uint32(0) == 0x464c457f and 8 of them
}
```

G_APT_Backdoor_BRICKSTORM_2

```
 rule G_APT_Backdoor_BRICKSTORM_2 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $str1 = { 0F 57 C0 0F 11 84 ?? ?? ?? ?? ?? C6 44 ?? ?? 00 4? C7 84 ?? ?? ?? ?? ?? 00 00 00 00
                $str2 = { 4? C7 84 ?? ?? ?? ?? ?? 00 00 00 00 4? C7 84 ?? ?? ?? ?? ?? 00 00 00 00 4? C7 84 ??
        condition:
                uint32be(0) == 0x7F454C46 and any of them
}
```

G_APT_BackdoorWebshell_SLAYSTYLE_1

```
 rule G_APT_BackdoorWebshell_SLAYSTYLE_1 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $str1 = /String \w{1,10}=request\.getParameter\(\"\w{1,15}\"\);/ ascii wide nocase
                $str2 = "=new String(java.util.Base64.getDecoder().decode(" ascii wide nocase
```

```
                $str21 = /String\[\]\s\w{1,10}=\{\"\/bin\/sh\",\"-c\",\w{1,10}\+\"\s2>&1\"\};/ ascii wide noca
                $str3 = "= Runtime.getRuntime().exec(" ascii wide nocase
                $str4 = "java.io.InputStream" ascii wide nocase
                $str5 = "java.util.Base64.getEncoder().encodeToString(org.apache.commons.io.IOUtils.toByteArra
        condition:
                filesize < 5MB and all of them
}
```

G_APT_BackdoorWebshell_SLAYSTYLE_2

```
rule G_APT_BackdoorWebshell_SLAYSTYLE_2 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $str1 = "request.getParameter" nocase
                $str2 = "/bin/sh"
                $str3 = "java.io.InputStream" nocase
                $str4 = "Runtime.getRuntime().exec(" nocase
                $str5 = "2>&1"
        condition:
                (uint16(0) != 0x5A4D and uint32(0) != 0x464C457F) and filesize < 7KB and all of them and @str4
}
```

G_Backdoor_BRICKSTEAL_1

```
rule G_Backdoor_BRICKSTEAL_1 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $str1 = "comvmware"
                $str2 = "abcdABCD1234!@#$"
                $str3 = "ads.png"
                $str4 = "User-Agent"
                $str5 = "com/vmware/"
        condition:
                all of them and filesize < 10KB
}
```

G_Dropper_BRICKSTEAL_1

```
rule G_Dropper_BRICKSTEAL_1 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
```

```
                $str1 = "Base64.getDecoder().decode"
                $str2 = "Thread.currentThread().getContextClassLoader()"
                $str3 = ".class.getDeclaredMethod"
                $str4 = "byte[].class"
                $str5 = "method.invoke"
                $str6 = "filterClass.newInstance()"
                $str7 = "/websso/SAML2/SSO/*"
        condition:
                all of them
}
```

G_Dropper_BRICKSTEAL_2

```
 rule G_Dropper_BRICKSTEAL_2 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $str1 = /\(Class<\?>\)\smethod\.invoke\(\w{1,20},\s\w{1,20},\s0,\s\w{1,20}\.length\);/i ascii
                $str2 = "(\"yv66vg" ascii wide
                $str3 = "request.getSession().getServletContext" ascii wide
                $str4 = ".getClass().getDeclaredField(" ascii wide
                $str5 = "new FilterDef();" ascii wide
                $str6 = "new FilterMap();" ascii wide
        condition:
                all of them
}
```

Network Detections

Google SecOps customers have access to these broad category rules and more under the Mandiant Front-Line Threats rule pack. The following are YARA-L 2.0 rules for use in Google Security Operations; however, their logic can be replicated into other formats for use in other security products.

Multiple DNS-over-HTTPS Services Queried

```
rule hunting_t1071_001_multiple_dns_over_https_services_queried {
 meta:
   rule_name = "Multiple DNS-over-HTTPS Services Queried"
   severity = "Low"
   tactic = "TA0011" // Command and Control
   technique = "T1071.001" // Application Layer Protocol: Web Protocols
   reference = "https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign"
   description = "Detects on requests by a source IP address to DNS-over-HTTPS (DoH) resolver IP addresses assc

 events:
```

```
    $e.metadata.event_type = "NETWORK_CONNECTION"
    $e.target.ip = /^(8\.8\.8\.8|8\.8\.4\.4|9\.9\.9\.9|9\.9\.9\.11|1\.1\.1\.1|1\.0\.0\.1|45\.90\.28\.160|45\.90\
    (
      $e.target.port = 443 or
      $e.target.url = /dns-query|:443\/$|\d\.\d\.\d\.\d\/$/ nocase
    )
    $source_entity = strings.coalesce($e.principal.asset_id,$e.principal.ip)

  match:
    $source_entity over 2h

  outcome:
    $risk_score = max(35)
    $unique_doh_ips_count = count_distinct($e.target.ip)

  condition:
    $e and $unique_doh_ips_count >= 5

  options:
    allow_zero_values = true
}
```

Unknown Endpoint Generating DNS-over-HTTPS and Web Application Development Services Communication

```
rule hunting_t1071_001_unknown_endpoint_generating_doh_and_web_development_services_communication {
 meta:
   rule_name = "Unknown Endpoint Generating DNS-over-HTTPS and Web Application Development Services Communicati
   severity = "Medium"
   tactic = "TA0011" // Command and Control
   technique = "T1071.001" // Application Layer Protocol: Web Protocols
   reference = "https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign"
   description = "Detects on requests by an unknown source IP address to multiple DNS-over-HTTPS (DoH) resolver

 events:
   $c1.metadata.event_type = "NETWORK_CONNECTION"
   $c1.target.ip = /^(8\.8\.8\.8|8\.8\.4\.4|9\.9\.9\.9|9\.9\.9\.11|1\.1\.1\.1|1\.0\.0\.1|45\.90\.28\.160|45\.90
   $c1.principal.hostname = ""
   $c1.principal.asset_id = ""
   (
     $c1.target.port = 443 or
     $c1.target.url = /dns-query|:443\/$|\d\.\d\.\d\.\d\/$/ nocase
   )
   $c2.metadata.event_type = "NETWORK_CONNECTION"
   $c2.target.hostname = /\.workers\.dev$|\.herokuapp\.com$/ nocase
   $c2.principal.hostname = ""
```

```
    $c2.principal.asset_id = ""
    $c2.target.port = 443
    $source_entity = $c1.principal.ip
    $source_entity = $c2.principal.ip

  match:
    $source_entity over 24h

  outcome:
    $risk_score = max(65)
    $unique_doh_ips_count = count_distinct($c1.target.ip)

  condition:
    $c1 and $c2 and $unique_doh_ips_count >= 3

  options:
    allow_zero_values = true
}
```

Unknown Endpoint Generating Google DNS-over-HTTPS and Cloudflare Hosted IP Communication

```
 rule hunting_t1071_001_unknown_endpoint_generating_google_doh_and_cloudflare_communication {
  meta:
    rule_name = "Unknown Endpoint Generating Google DNS-over-HTTPS and Cloudflare Hosted IP Communication"
    severity = "Medium"
    tactic = "TA0011" // Command and Control
    technique = "T1071.001" // Application Layer Protocol: Web Protocols
    reference = "https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign"
    description = "Detects on requests by an unknown source IP address to Google DNS-over-HTTPS (DoH) resolver s

  events:
    $c1.metadata.event_type = "NETWORK_CONNECTION"
    $c1.target.ip = /^(8\.8\.8\.8|8\.8\.4\.4)$/ nocase
    $c1.principal.hostname = ""
    $c1.principal.asset_id = ""
    (
      $c1.target.port = 443 or
      $c1.target.url = /dns-query|:443\/$|\d\.\d\.\d\.\d\/$/ nocase
    )
    $c2.metadata.event_type = "NETWORK_CONNECTION"
    $c2.principal.hostname = ""
    $c2.principal.asset_id = ""
    $c2.target.ip_geo_artifact.network.carrier_name = /cloudflare/ nocase
    $c2.target.port = 443
    $source_entity = $c1.principal.ip
    $source_entity = $c2.principal.ip
```

```
  match:
    $source_entity over 1h

  outcome:
    $risk_score = max(65)
    $time_diff = math.abs(min($c1.metadata.event_timestamp.seconds) - min($c2.metadata.event_timestamp.seconds))

  condition:
    $c1 and $c2 and $time_diff <= 2

  options:
    allow_zero_values = true
}
```

Unknown Endpoint Generating Google DNS-over-HTTPS and Amazon Hosted IP Communication

```
 rule hunting_t1071_001_unknown_endpoint_generating_google_doh_and_amazon_communication {
  meta:
    rule_name = "Unknown Endpoint Generating Google DNS-over-HTTPS and Amazon Hosted IP Communication"
    severity = "Medium"
    tactic = "TA0011" // Command and Control
    technique = "T1071.001" // Application Layer Protocol: Web Protocols
    reference = "https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign"
    description = "Detects on requests by an unknown source IP address to Google DNS-over-HTTPS (DoH) resolver s

  events:
    $c1.metadata.event_type = "NETWORK_CONNECTION"
    $c1.target.ip = /^(8\.8\.8\.8|8\.8\.4\.4)$/ nocase
    $c1.principal.hostname = ""
    $c1.principal.asset_id = ""
    (
      $c1.target.port = 443 or
      $c1.target.url = /dns-query|:443\/$|\d\.\d\.\d\.\d\/$/ nocase
    )
    $c2.metadata.event_type = "NETWORK_CONNECTION"
    $c2.principal.hostname = ""
    $c2.principal.asset_id = ""
    $c2.target.ip_geo_artifact.network.carrier_name = /amazon/ nocase
    $c2.target.port = 443
    $source_entity = $c1.principal.ip
    $source_entity = $c2.principal.ip

  match:
    $source_entity over 24h
```

```
  outcome:
    $risk_score = max(65)
    $time_diff = math.abs(min($c1.metadata.event_timestamp.seconds) - min($c2.metadata.event_timestamp.seconds))

  condition:
    // As observed by Mandiant IR, the two connection events to DoH and Amazon occurred nearly simultaneously
    $c1 and $c2 and $time_diff <= 2

  options:
    allow_zero_values = true
}
```

Posted in

- Threat Intelligence

Source: https://cloud.google.com/blog/topics/threat-intelligence/brickstorm-espionage-campaign