UNC3524: Eye Spy on Your Email

Doug Bienstock, Melissa Derr, Josh Madeley, Tyler McLellan, Chris Gardner May 02, 2022 14 mins read Uncategorized Groups (UNC Groups) Threat Research Threat Intelligence Malware email backdoor

Since December 2019, Mandiant has observed advanced threat actors increase their investment in tools to facilitate bulk email collection from victim environments, especially as it relates to their support of suspected espionage objectives. Email messages and their attachments offer a rich source of information about an organization, stored in a centralized location for threat actors to collect. Most email systems, whether on-premises or in the cloud, offer programmatic methods to search and access email data across an entire organization, such as eDiscovery and the Graph API. Mandiant has observed threat actors use these same tools to support their own collection requirements and to target the mailboxes of individuals in victim organizations.

In this blog post, we introduce UNC3524, a newly discovered suspected espionage threat actor that, to date, heavily targets the emails of employees that focus on corporate development, mergers and acquisitions, and large corporate transactions. On the surface, their targeting of individuals involved in corporate transactions suggests a financial motivation; however, their ability to remain undetected for an order of magnitude longer than the average dwell time of 21 days in 2021, as reported in M-Trends 2022, suggests an espionage mandate. Part of the group’s success at achieving such a long dwell time can be credited to their choice to install backdoors on appliances within victim environments that do not support security tools, such as anti-virus or endpoint protection. The high level of operational security, low malware footprint, adept evasive skills, and a large Internet of Things (IoT) device botnet set this group apart and emphasize the “advanced” in Advanced Persistent Threat. UNC3524 also takes persistence seriously. Each time a victim environment removed their access, the group wasted no time re-compromising the environment with a variety of mechanisms, immediately restarting their data theft campaign. We are sharing the tools, tactics, and procedures used by UNC3524 to help organizations hunt for and protect against their operations.

Attack Lifecycle

Initial Compromise and Maintain Presence

After gaining initial access by unknown means, UNC3524 deployed a novel backdoor tracked by Mandiant as QUIETEXIT, which is based on the open-source Dropbear SSH client-server software. For their long-haul remote access, UNC3524 opted to deploy QUIETEXIT on opaque network appliances within the victim environment; think backdoors on SAN arrays, load balancers, and wireless access point controllers. These kinds of devices don’t support antivirus or endpoint detection and response tools (EDRs), subsequently leaving the underlying operating systems to vendors to manage. These appliances are often running older versions of BSD or CentOS and would require considerable planning to compile functional malware for them. By targeting trusted systems within victim environments that do not support any type of security tooling, UNC3524 was able to remain undetected in victim environments for at least 18 months.

QUIETEXIT works as if the traditional client-server roles in an SSH connection were reversed. Once the client, running on a compromised system, establishes a TCP connection to a server, it performs the SSH server role. The QUIETEXIT component running on the threat actor’s infrastructure initiates the SSH connection and sends a password. Once the backdoor establishes a connection, the threat actor can use any of the options available to an SSH client, including proxying traffic via SOCKS. QUIETEXIT has no persistence mechanism; however, we have observed UNC3524 install a run command (rc) as well as hijack legitimate application-specific startup scripts to enable the backdoor to execute on system startup.

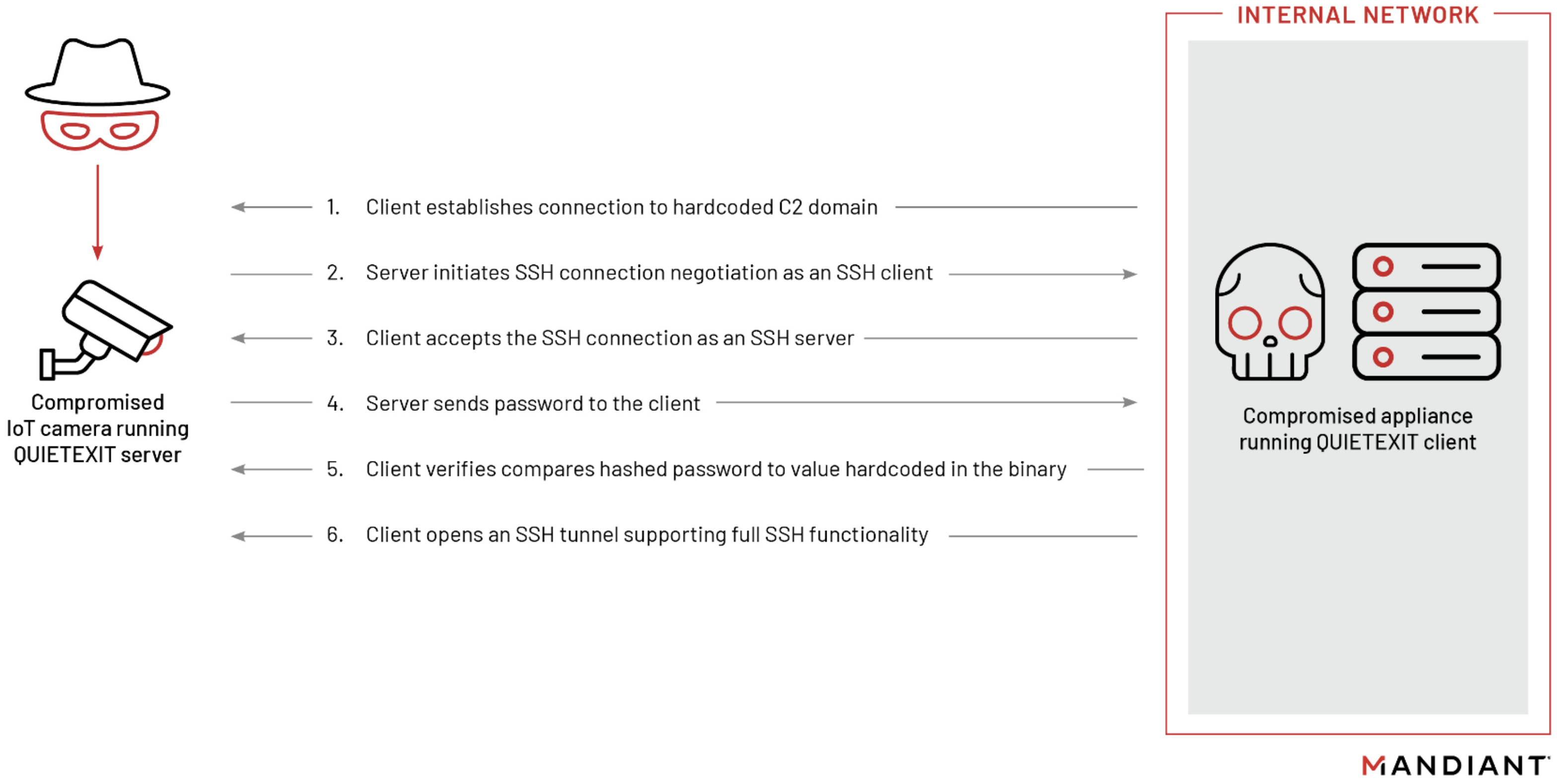
INTERNAL NETWORK
1. Client establishes connection to hardcoded C2 domain
2. Server initiates SSH connection negotiation as an SSH client
3. Client accepts the SSH connection as an SSH server
4. Server sends password to the client
5. Client verifies compares hashed password to value hardcoded in the binary
6. Client opens an SSH tunnel supporting full SSH functionality
Compromised IoT camera running QUIETEXIT server
Compromised appliance running QUIETEXIT client
MANDIANT

Figure 1: How QUIETEXIT works with IoT devices

On startup, QUIETEXIT attempts to change its name to cron, but the malware author did not implement this correctly, so it fails. During our incident response investigations, we recovered QUIETEXIT samples that were renamed to blend in with other legitimate files on the file system. In one case with an infected node of a NAS array, UNC3524 named the binary to blend in with a suite of scripts used to mount various filesystems to the NAS.

When run with command line arguments `-X -p <port>` the malware connects to a hard-coded command and control (C2) address on the specific port. If this fails, it will attempt to connect to a second hard coded C2 if one is configured. The user can also specify a hostname or IP address on the command line in the -p argument as well, e.g. `-X -p <host>:<port>`.The -X command line argument is case sensitive. If the lower-case x option is used, then the malware will only attempt to connect to the C2 server once. If the upper-case X option is used, then the malware will sleep for a random number of minutes between a hard-coded time range and fork to reattempt the connection. It re-attempts the connection regardless of whether a connection has already been established. In our investigations we observed UNC3524 use C2 domains that intended to blend in with legitimate traffic originating from the infected appliances. Using the example of an infected load balancer, the C2 domains contained strings that could plausibly relate to the device vendor and branded operating system name. This level of planning demonstrates that UNC3524 understands incident response processes and tried to make their C2 traffic appear as legitimate to anyone that might scroll through DNS or session logs.

All QUIETEXIT C2 domains that Mandiant observed used Dynamic DNS providers. Dynamic DNS allows for threat actors to update the DNS records for domains in a near seamless fashion. When the C2s where inactive, the threat actor had the domains resolve to 127.0.0.1. However, occasionally the port numbers would change or VPS infrastructure would be used rather than compromised camera botnet. We suspected that when the threat actor experienced issues accessing a victim, they would troubleshoot using new infrastructure or different ports.

In some cases, the threat actor deployed a secondary backdoor as a means of alternate access into victim environments. This alternate access was a REGEORG web shell previously placed on a DMZ web server. REGEORG is a web shell that creates a SOCKS proxy, keeping with UNC3452's preference for tunneling malware. Once inside the victim environment, the threat actor spent time to identify web servers in the victim environment and ensure they found one that was Internet accessible before copying REGEORG to it. They also took care to name the file so that it blended in with the application running on the compromised server. Mandiant also observed instances where UNC3452 used timestomping to alter the Standard Information timestamps of the REGEORG web shell to match other files in the same directory.

UNC3452 only used these web shells when their QUIETEXIT backdoors stopped functioning and only to re-establish QUIETEXIT on another system in the network. Rather than use the public version of REGEORG published by Sensepost, UNC3452 used a still public but little-known version of the web shell that is heavily obfuscated. This allowed them to bypass common signature-based detections for REGEORG.

Move Laterally

Once UNC3524 established a foothold in the network they demonstrated a very low malware footprint and instead relied on built-in Windows protocols. During our incident response investigations, we traced most accesses to a victim appliance infected with QUIETEXIT. QUIETEXIT supports the full functionality of SSH, and our observation is consistent with UNC3524 using it to establish a SOCKS tunnel into the victim environments. By standing up a SOCKS tunnel, the threat actor effectively plugs in their machine to an ethernet jack within the victim's network. By tunneling over SOCKS, the threat actor can execute tools to steal data from their own computer, leaving no traces of the tooling itself on victim computers.

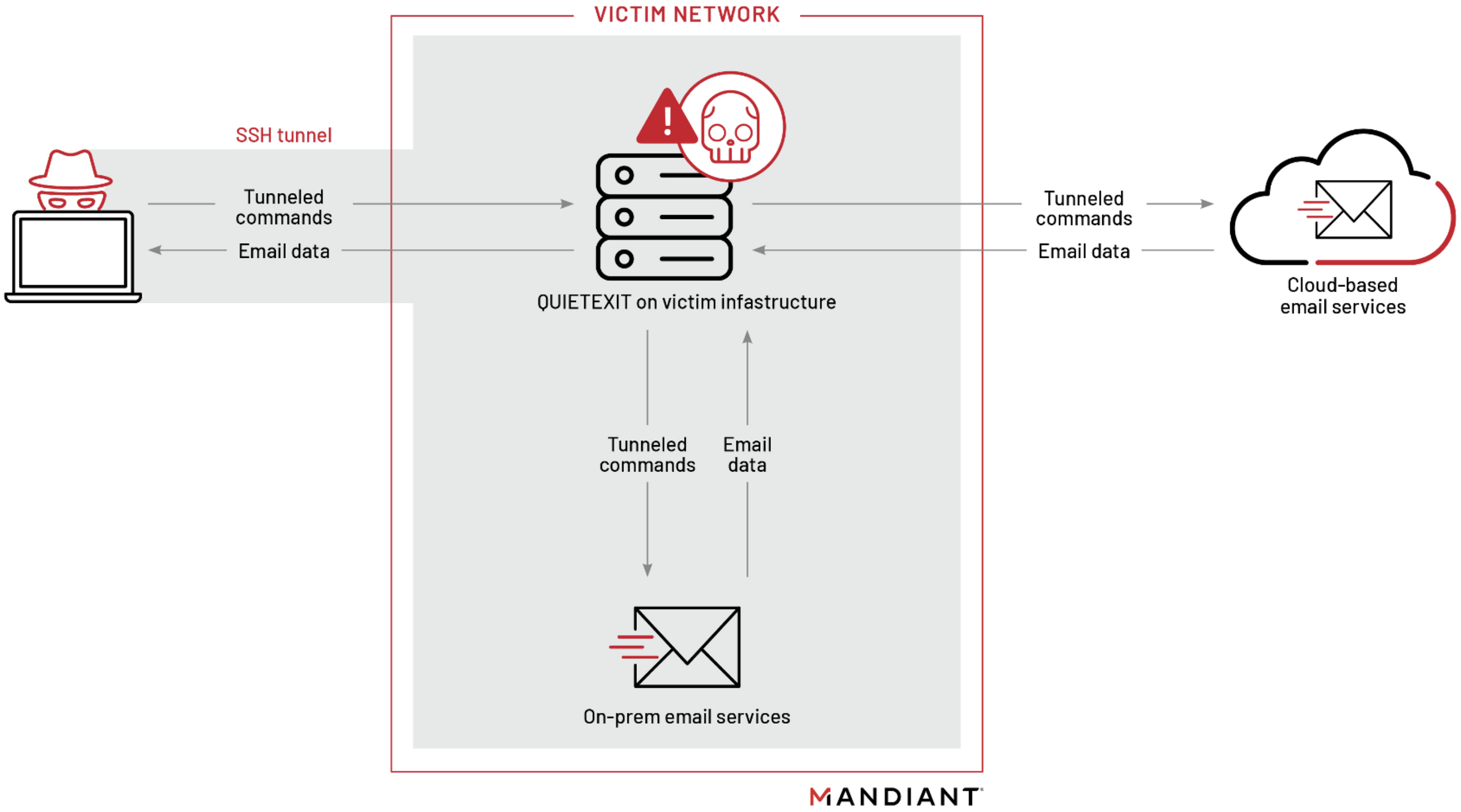

Figure 2: Tunneling through QUIETEXIT

To perform lateral movement to systems of interest, UNC3524 used a customized version of Impacket's WMIEXEC. WMIEXEC uses Windows Management Instrumentation to establish a semi-interactive shell on a remote host. The utility provides a semi-interactive shell by writing command outputs to a file on the remote host and then printing the output to the terminal. The default Impacket version uses a hardcoded file path and filename structure for these output files, providing a detection opportunity. Mandiant has observed UNC3524 modifying the hardcoded file path (\\127.0.0.1\ADMIN$\debug\DEBUG.LOG) to evade basic detections for filenames such as Impacket's default double underscore files. We also observed the threat actor using the built-in `reg save` command to save registry hives and extract LSA secrets offline.

Complete Mission

Once UNC3524 successfully obtained privileged credentials to the victim's mail environment, they began making Exchange Web Services (EWS) API requests to either the on-premises Microsoft Exchange or Microsoft 365 Exchange Online environment. In each of the UNC3524 victim environments, the threat actor would target a subset of mailboxes, focusing their attention on executive teams and employees that work in corporate development, mergers and acquisitions, or IT security staff. It's likely that the threat actor was targeting the IT security team as a method to determine if their operation had been detected.

The methods that UNC3524 used to authenticate to the Exchange infrastructure evolved throughout the course of the intrusions; this may be a result of them periodically losing access due to the natural changes in corporate infrastructure or simply updating their tactics. They authenticated to Exchange using the username and password of targeted accounts, using accounts holding ApplicationImpersonation rights, or using Service Principal credentials. Each of these methods, their detections, and configuration recommendations can be found at Mandiant's UNC2452 Microsoft 365 Hardening Guide.

Once authenticated to the exchange infrastructure, UNC3524 made a series of EWS API requests to extract mail items from the target mailbox. For each mailbox, the threat actor made a series of `GetFolder` and `FindFolder` requests that returned data describing the mailbox, such as the number of unread messages and sub-folders within the specified folder.

Figure 3: Sample EWS GetFolder request

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types"> <soap:Header>

<t:RequestServerVersion Version="Exchange2013" />

<t:ExchangeImpersonation>

<t:ConnectingSID>

<t:PrimarySmtpAddress>target@victimorg.com</t:PrimarySmtpAddress>
```

```
</t:ConnectingSID>
</t:ExchangeImpersonation>
</soap:Header>
<soap:Body>
<GetFolder xmlns="https://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types">
<FolderShape>
<t:BaseShape>Default</t:BaseShape>
</FolderShape>
<FolderIds>
<t:DistinguishedFolderId Id="Root"/>
</FolderIds>
</GetFolder>
</soap:Body>
</soap:Envelope>
```

After the enumeration of the mailbox structure, the threat actor issued a `FindItem` request with a Query Filter that selected all messages from a specific folder with a `DateTimeCreated` greater than a specific date. The date in the filter corresponded to the last time the threat actor accessed the mailbox. This meant that the threat actor would acquire all newly created items in the mailbox since the last time they had extracted data. This follows an approach that Mandiant has previously observed with APT29. Rather than target a mailbox using specific keywords, the threat actor instead extracted the entire contents over a particular date range.

Figure 4: Sample EWS FindItem request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:m="https://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types"
xmlns:soap="https://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>
<t:RequestServerVersion Version="Exchange2013" />
<t:ExchangeImpersonation>
<t:ConnectingSID>
<t:PrimarySmtpAddress>target@victimorg.com</t:PrimarySmtpAddress>
</t:ConnectingSID>
</t:ExchangeImpersonation>
</soap:Header>
```

```
<soap:Body>
<m:FindItem Traversal="Shallow">
<m:ItemShape>
<t:BaseShape>IdOnly</t:BaseShape>
<t:AdditionalProperties>
<t:FieldURI FieldURI="item:Subject" />
<t:FieldURI FieldURI="item:DateTimeCreated " />
</t:AdditionalProperties>
</m:ItemShape>
<m:IndexedPageItemView MaxEntriesReturned="100" Offset="0" BasePoint="Beginning" />
<m:Restriction>
<t:IsGreaterThan>
<t:FieldURI FieldURI="item:DateTimeCreated " />
<t:FieldURIOrConstant>
<t:Constant Value="2022-01-01T00:00:00" />
</t:FieldURIOrConstant>
</t:IsGreaterThan>
</m:Restriction>
<m:SortOrder>
<t:FieldOrder Order="Descending">
<t:FieldURI FieldURI="item:DateTimeCreated" />
</t:FieldOrder>
</m:SortOrder>
<m:ParentFolderIds>
<t:DistinguishedFolderId Id="inbox" />
</m:ParentFolderIds>
</m:FindItem>
</soap:Body>
</soap:Envelope>
```

Finally, the threat actor iterated through each message identifier returned in the `FindItem` response and made a `GetItem` request. The threat actor set the `IncludeMimeContent` parameter to true for the request, which resulted in Exchange returning the message in MIME format. This is important because the MIME message includes both the message body and any attachments. It is worth noting that if the messages were encrypted using PGP, SMIME, Office 365 Message Encryption (OME), or other encryption technology, then the `GetItem` response will only contain the ciphertext or in the case of OME, a link to authenticate and view the real message.

Figure 5: Sample EWS GetItem request

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types">

<soap:Body>

<GetItem

xmlns="https://schemas.microsoft.com/exchange/services/2006/messages"

xmlns:t="https://schemas.microsoft.com/exchange/services/2006/types">

<ItemShape>

<t:BaseShape>Default</t:BaseShape>

<t:IncludeMimeContent>true</t:IncludeMimeContent>

</ItemShape>

<ItemIds>

<t:ItemId Id="<ID OF MESSAGE>" ChangeKey="CQAAAB" />

</ItemIds>

</GetItem>

</soap:Body>

</soap:Envelope>
```

Operational Security and Infrastructure

Throughout their operations, the threat actor demonstrated sophisticated operational security that we see only a small number of threat actors demonstrate. The threat actor evaded detection by operating from devices in the victim environment's blind spots, including servers running uncommon versions of Linux and network appliances running opaque OSes. These devices and appliances were running versions of operating systems that were unsupported by agent-based security tools, and often had an expected level of network traffic that allowed the attackers to blend in. The threat actor's use of the QUIETEXIT tunneler allowed them to largely live off the land, without the need to bring in additional tools, further reducing the opportunity for detection. This allowed UNC3524 to remain undetected in victim environments for, in some cases, upwards of 18 months.

The C2 systems that Mandiant identified were primarily legacy conference room camera systems sold by LifeSize, Inc. and in one instance, a D-Link IP camera. These camera systems appeared to be infected, likely with the server component of QUIETEXIT. These cameras were directly Internet exposed, possibly through an improper UPnP configuration, and may have been running older firmware. Mandiant suspects that default credentials, rather than an exploit, were the likely mechanism used to compromise these devices and form the IoT botnet used by UNC3524. Similar to the use of embedded network devices, UNC3524 can avoid detection by operating from compromised infrastructure connected directly to the public Internet such as IP cameras where typical antivirus and security monitoring may be absent.

Detection

UNC3524's use of compromised appliances makes host-based hunting and detection extremely difficult. The best opportunity for detection remains in network-based logging, specifically monitoring traffic at the layer 7 level. Mandiant recommends hunting for traffic tagged as the "SSH" application egressing environments over ports other than 22. This traffic should be relatively small, and any findings should be investigated. Organizations can also look for outbound SSH traffic originating from IP addresses that are unknown or not in asset management systems. These source systems are more likely to be appliances that aren't centrally managed. Finally, large volumes of network traffic originating from the "management" interfaces of appliances such as NAS arrays and load balancers should be investigated as suspicious as well.

UNC3524 targets opaque network appliances because they are often the most unsecure and unmonitored systems in a victim environment. Organizations should take steps to inventory their devices that are on the network and do not support monitoring tools. Each device likely has vendor-specific hardening actions to take to ensure that the proper logging is enabled, and logs are forwarded to a central repository. Organizations can also take steps to use network access controls to limit or completely restrict egress traffic from these devices.

For host-based hunting, Mandiant recommends hunting for QUIETEXIT on devices using the provided grep commands. Most appliances that provide shell access should have the grep binary available.

Find QUIETEXIT hard-coded byte string using grep:

```
grep "\x48\x8b\x3c\xd3\x4c\x89\xe1\xf2\xae" -rs /
```

Find QUIETEXIT by looking for the hard-coded password value:

```
grep '\xDD\xE5\xD5\x97\x20\x53\x27\xBF\xF0\xA2\xBA\xCD\x96\x35\x9A\xAD\x1C\x75\xEB\x47' -rs /
```

Find QUIETEXIT persistence mechanisms in the appliance's rc.local directory by looking for the command line arguments:

```
grep -e " -[Xx] -p [[:digit:]{2,6}]" -rs /etc
```

Remediation and Hardening

Mandiant has published remediation and hardening strategies for Microsoft 365.

Attribution

The methodologies Mandiant observed during UNC3524 intrusions overlapped with techniques used by multiple Russia-based espionage threat actors including both EWS impersonation and SPN credential addition. Mandiant has only observed APT29 performing SPN credential addition; however, this technique has been reported on publicly since early 2019. The NSA has previously reported automated password spraying using Kubernetes, Exchange Exploitation, and REGEORG as associated with APT28. While the activity reported by the NSA used TOR and commercial VPNs, UNC3524 primarily used compromised internet facing devices. One interesting aspect of UNC3524's use of REGEORG was that it matched identically with the version publicly reported by the NSA as used by APT28. At the time of writing, Mandiant cannot conclusively link UNC3524 to an existing group currently tracked by Mandiant.

Acknowledgements

We would like to thank our incident response consultants, Managed Defense responders, and FLARE reverse engineers who enabled this research. Thanks to Kirstie Failey, Jake Nicastro, John Wolfram, Sarah Hawley and Nick Richard for technical review, and Ryan Hall and Alyssa Rahman for research contributions.

MITRE ATT&CK

Mandiant has observed UNC3524 use the following techniques.

ATT&CK Tactic Category	Techniques
Defense Evasion	T1027: Obfuscated Files or Information
Discovery	T1012: Query Registry T1016: System Network Configuration Discovery

	T1049: System Network Connections Discovery T1057: Process Discovery T1518: Software Discovery
Credential Access	T1003.004: LSA Secrets T1003.006: DCSync T1111: Two-Factor Authentication Interception
Collection	T1114: Email Collection T1114.002: Remote Email Collection
Lateral Movement	T1021.004: SSH
Persistence	T1037.004: RC Scripts T1098.001: Additional Cloud Credentials T1505.003: Web Shell
Command and Control	T1071: Application Layer Protocol T1090.003: Multi-hop Proxy T1095: Non-Application Layer Protocol T1572: Protocol Tunneling T1573.002: Asymmetric Cryptography
Resource Development	T1583.003: Virtual Private Server T1584: Compromise Infrastructure T1608.003: Install Digital Certificate
Execution	T1059.001: PowerShell T1059.003: Windows Command Shell

YARA Signatures

Note: These rules are designed to broadly capture suspicious files and are not designed to detect a particular malware or threat.

```
rule QUIETEXIT_strings

{

meta:

author = "Mandiant"

date_created = "2022-01-13"
date_modified = "2022-01-13"
rev = 1
strings:
$s1 = "auth-agent@openssh.com"
$s2 = "auth-%.8x-%d"
$s3 = "Child connection from %s:%s"
$s4 = "Compiled without normal mode, can't run without -i"
$s5 = "cancel-tcpip-forward"
$s6 = "dropbear_prng"
$s7 = "cron"
condition:
uint32be(0) == 0x7F454C46 and filesize < 2MB and all of them
}

rule REGEORG_Tuneller_generic
{
meta:
author = "Mandiant"
date_created = "2021-12-20"
date_modified = "2021-12-20"
md5 = "ba22992ce835dadcd06bff4ab7b162f9"
strings:
$s1 = "System.Net.IPEndPoint"
$s2 = "Response.AddHeader"
$s3 = "Request.InputStream.Read"
$s4 = "Request.Headers.Get"
$s5 = "Response.Write"
$s6 = "System.Buffer.BlockCopy"
$s7 = "Response.BinaryWrite"
$s8 = "SocketException soex"
condition:
filesize < 1MB and 7 of them
}
```

```
rule UNC3524_sha1

{

meta:

author = "Mandiant"

date_created = "2022-01-19"

date_modified = "2022-01-19"

strings:

$h1 = { DD E5 D5 97 20 53 27 BF F0 A2 BA CD 96 35 9A AD 1C 75 EB 47 }

condition:

uint32be(0) == 0x7F454C46 and filesize < 10MB and all of them

}
```

Indicators

MALWARE FAMILY	Indicator
QUIETEXIT	Dynamic DNS cloudns.asia dynu.net mywire.org webredirect.org

MALWARE FAMILY	MD5	SHA1	SHA256
REGEORG GitHub version	ba22992ce835dadcd06bff4ab7b162f9	3d4dcc859c6ca7e5b36483ad84c9ceef34973f9a	7b5e3c1c06d82b3e7309C258dfbd4bfcd476c8ffcb4cebda7614