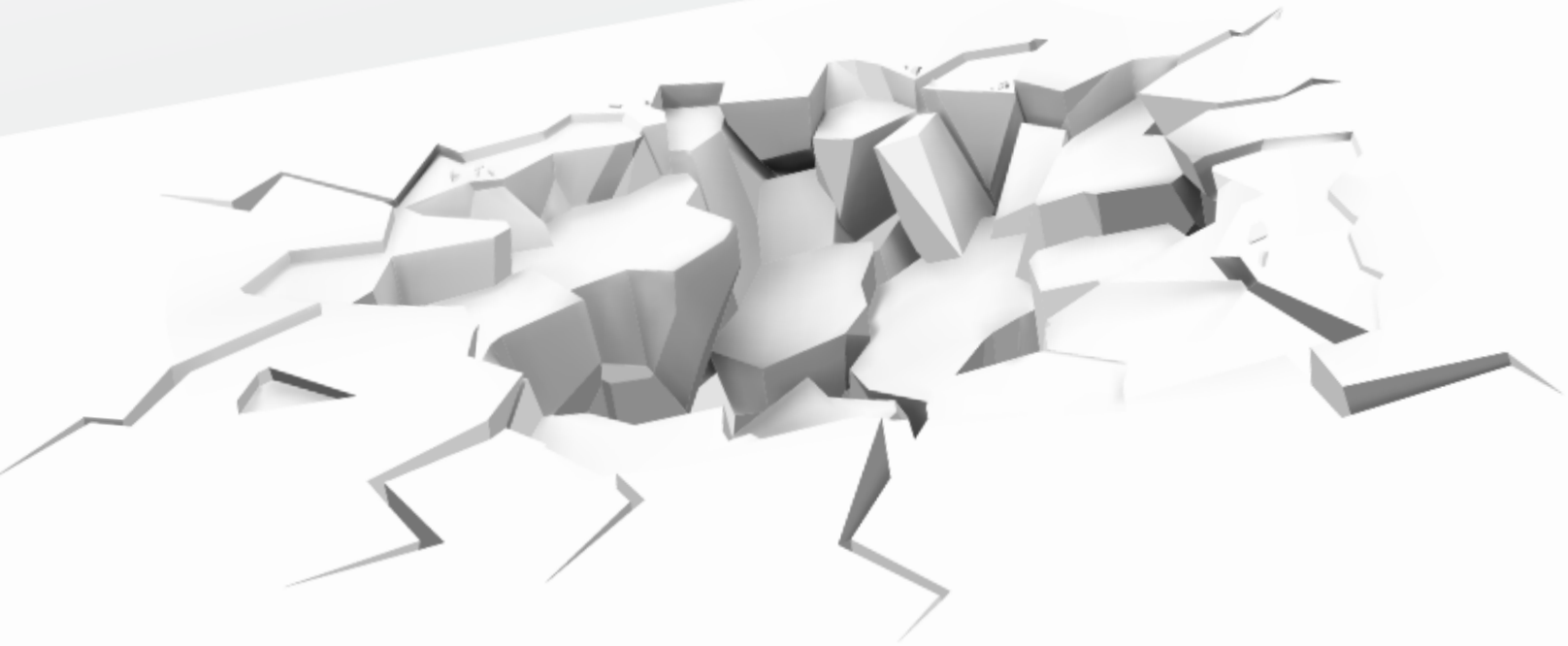
Mandiant

APT44: Unearthing Sandworm



Authors

Gabby Roncone, Dan Black, John Wolfram, Tyler McLellan, Nick Simonian, Ryan Hall, Anton Prokopenkov, Dan Perez, Lexie Aytes, Alden Wahlstrom

Acknowledgements

Collaboration with companies and governments to track and mitigate threats is critical to our collective efforts to defend our networks against adversaries. The efforts of Mandiant Consulting across many incident response engagements in Ukraine since 2022 enabled much of the analysis included in this report. We'd additionally like to thank Mandiant's FLARE team, former Mandiant employees, ESET, Microsoft, Google TAG, numerous global government organizations, and most importantly, all of our partners in Ukraine. Our work would not be possible without their contributions.

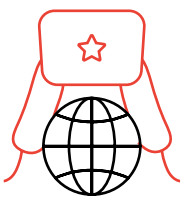
Executive Summary

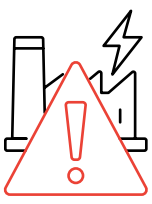
With Russia's war in Ukraine in its third year, Sandworm remains a formidable threat to Ukraine. The group's operations in support of Moscow's war aims have proven tactically and operationally adaptable, and as of today, appear to be better integrated with the activities of Russia's conventional forces than in any other previous phase of the conflict. To date, no other Russian government-backed cyber group has played a more central role in shaping and supporting Russia's military campaign.

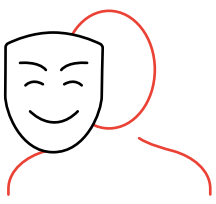
Yet the threat posed by Sandworm is far from limited to Ukraine. Mandiant continues to see operations from the group that are global in scope in key political, military, and economic hotspots for Russia. Looking forward, a record number of people will participate in national elections in 2024, and Sandworm's history of attempting to interfere in democratic processes further elevates the threat the group may pose in the near-term. Given the active and persistent threat to governments and critical infrastructure operators globally, Mandiant has decided to graduate the group into APT44.

Key Judgments

- Sponsored by Russian military intelligence, APT44 is a dynamic and operationally mature threat actor, actively engaged in the full spectrum of espionage, attack, and influence operations.
- APT44 has aggressively pursued a multi-pronged effort to help the Russian military gain a wartime advantage and is responsible for nearly all of the disruptive and destructive operations against Ukraine over the past decade.
- We assess with high confidence that APT44 is seen by the Kremlin as a flexible instrument of power capable of servicing Russia's wide ranging national interests and ambitions, including efforts to undermine democratic processes globally.
- Due to the group's history of aggressive use of network attack capabilities across political and military contexts, APT44 presents a persistent, high severity threat to governments and critical infrastructure operators globally where Russian national interests intersect.







Overview of APT44

APT44 (commonly known as Sandworm, FROZENBARENTS, and Seashell Blizzard) is a Russian Federation backed threat group attributed by multiple governments to Unit 74455, the Main Centre for Special Technologies (GTsST) within the Main Directorate of the General Staff of the Armed Forces of the Russian Federation (GU), commonly known as the Main Intelligence Directorate (GRU). Mandiant has tracked APT44 operations for over a decade, with publicly available images of the unit's anniversary insignia placing the group's formation in 2009.

While most Russian state-backed threat groups tend to specialize in a specific mission, APT44 is a uniquely dynamic threat actor that is actively engaged in the full spectrum of cyber espionage, attack, and influence operations. These respective components constitute the gamut of special activities typically carried out by the GRU's Information Operation Troops (VIO), to which we assess APT44 is highly likely subordinated. We therefore view APT44 as a characteristic representation of the information confrontation (IPb) concept that underpins Russia's present-day cyber forces.

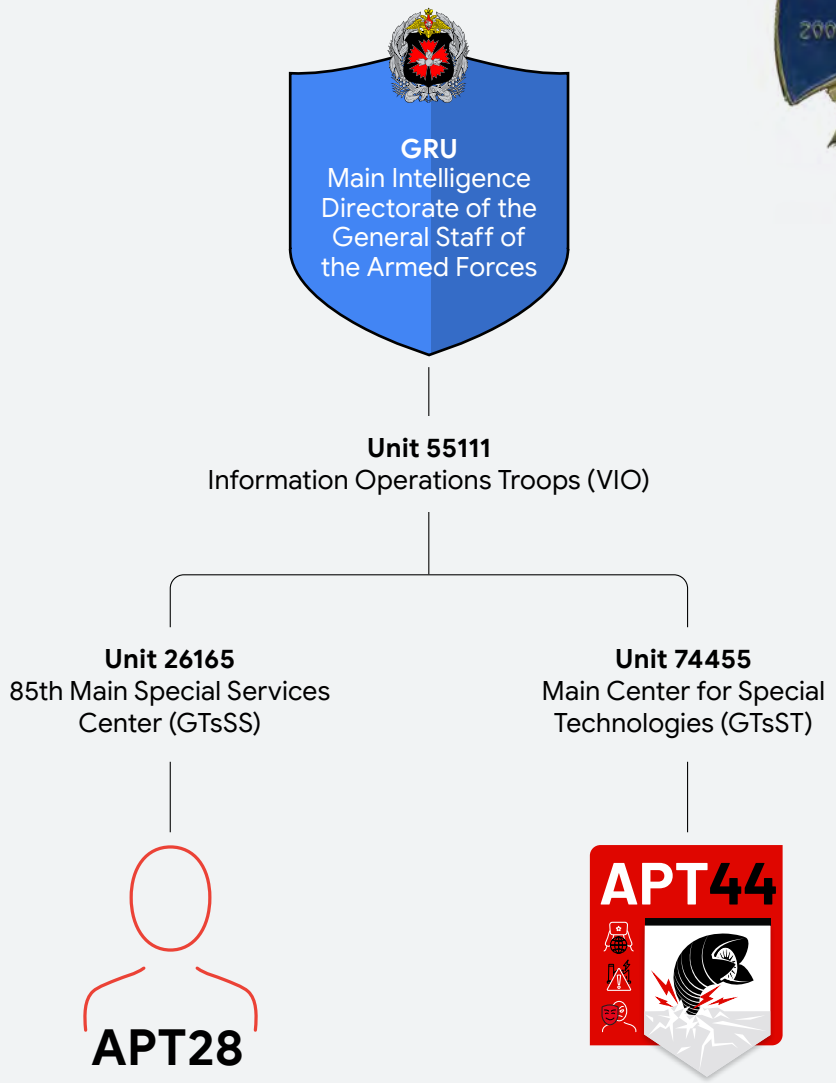


Figure 1. GRU VIO Structure including Unit 74455

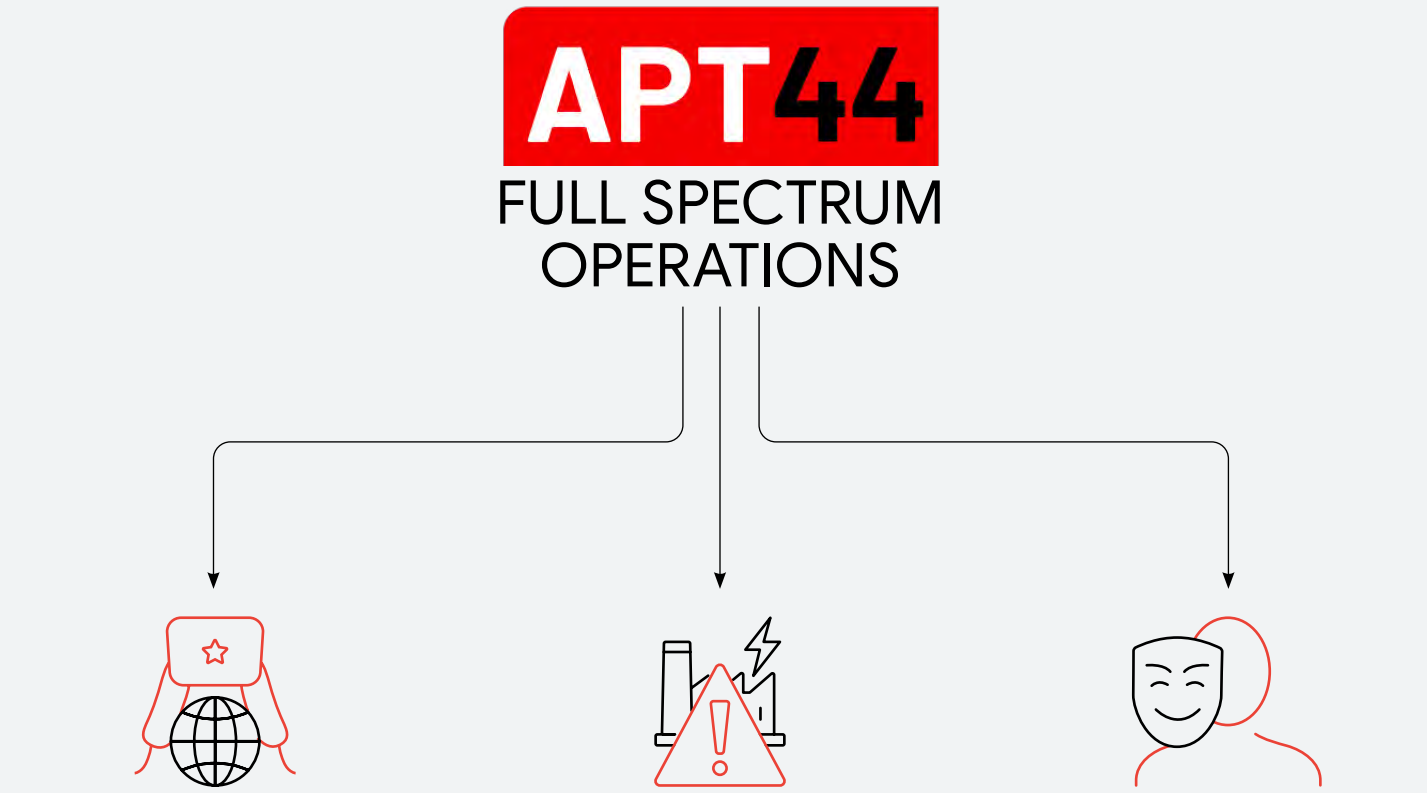
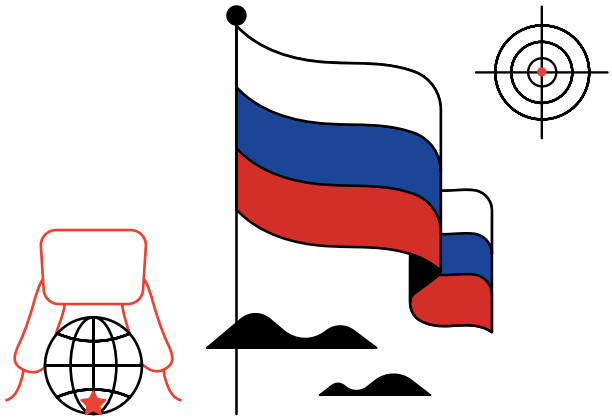


Figure 2. APT44 Full Spectrum Cyber Operations



A Global Targeting Mandate

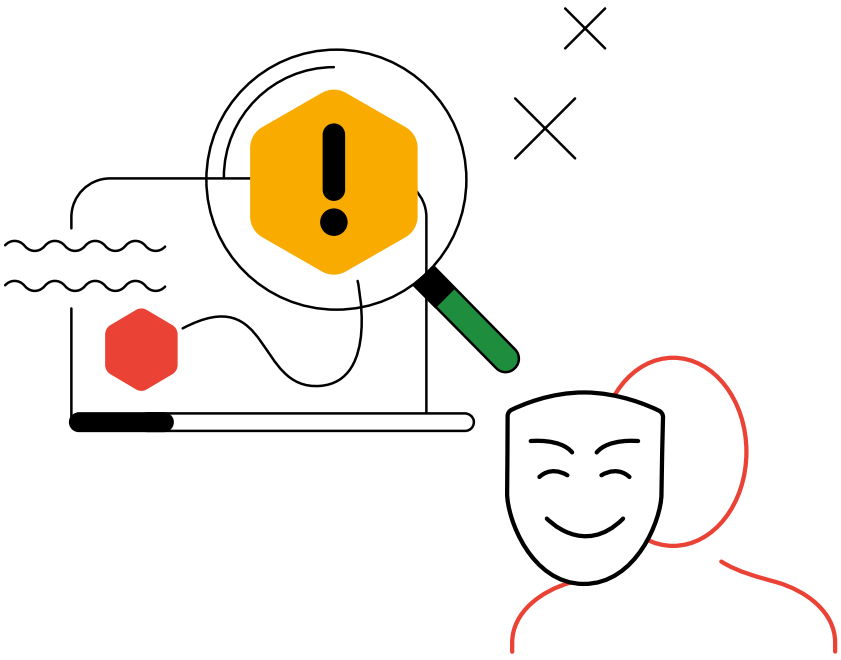
APT44 operations are global in scope and mirror Russia's wide ranging national interests and ambitions. In the post-Maidan Revolution era, this has led to cyber operations primarily centered on Ukraine, the epicenter of Russia's revanchist geopolitical aims over the past decade. However, even with an ongoing war, we have observed the group sustain access and espionage operations across North America, Europe, the Middle East, Central Asia, and Latin America. Patterns of activity over time indicate that APT44 is tasked with a range of different strategic priorities and is highly likely seen by the Kremlin as a flexible instrument of power capable of serving both enduring and emerging intelligence requirements.

- APT44 primarily targets government, defense, transportation, energy, media, and civil society organizations in Russia's near abroad. Government bodies and other Critical Infrastructure and Key Resources (CIKR) operators in Poland, Kazakhstan, and within Russia have frequently been included in the group's recent targeting.
- APT44 has repeatedly targeted Western electoral systems and institutions, including those in current and prospective North Atlantic Treaty Organization (NATO) member countries. As part of this activity, APT44 has attempted to interfere with democratic processes in select countries by leaking politically sensitive information and deploying malware to access election systems and misreport election data.
- In less discriminate operations, Mandiant continues to observe APT44 conduct widespread credential theft targeting public and private sector mail servers globally. This campaign, which dates back to at least 2019, has targeted various mail environments including Exim, Zimbra and Exchange servers across a wide-range of industry verticals.
- APT44 also frequently targets journalists, civil society organizations, and non-governmental bodies involved in research or investigations into the Russian government. Examples include the 2018 operation targeting the Organization for the Prohibition of Chemical Weapons (OPCW) for its role in the Novichok poisoning investigations and a phishing campaign by an assessed APT44 initial access cluster between December 2023 and January 2024 which targeted Bellingcat and other investigative journalism entities .

A Highly Adaptive Adversary

APT44 is a persistent and operationally mature adversary that uses diverse initial access methods ranging from common vectors such as phishing, credential harvesting, and known vulnerability exploitation to targeted supply chain compromises. The group commonly leverages nonselective initial access vectors that provide wide-ranging access to targets of interest, later down-selecting victims of interest for the full spectrum of follow-on activity.

- APT44 frequently achieves initial access through the exploitation of edge infrastructure such as routers and virtual private network (VPN) appliances. We have observed the group fulfill a variety of missions from footholds gained on network perimeters, including reconnaissance, information theft, downstream phishing, and the deployment of wiper malware.
- Following in the footsteps of ETERNALPETYA (aka NotPetya), APT44 also continues to subvert software supply chains for initial access. In one recent case, access to a software developer resulted in the downstream compromise of critical infrastructure networks in Eastern Europe and Central Asia, followed by the deployment of wiper malware to a select victim organization.
- APT44 is also known to employ unconventional methods to compromise targets of interest. As of February 2024, the group continues to leverage trojanized software installers distributed via torrents on Ukrainian- and Russian-language forums as a means of achieving opportunistic initial access to potential targets of interest. Once downloaded, victims of interest are manually flagged by APT44 operators with specifics such as the victim organizations or unit names, designating them for follow-on exploitation. We have seen these victims receive payloads such as DARKCRYSTALRAT (or DCRAT), commodity malware that APT44 has also used to target telecommunications entities in Ukraine.



Once inside a network, APT44 commonly uses living-off-the-land (LOTL) techniques to further its access, establish persistence, and exfiltrate information. The group is also known for its "low-equity" approach to malware delivery that prioritizes open source or criminally sourced tools over using its own custom implants.

- APT44 operates with a high degree of operational security and continuously adapts to circumvent best-practice defensive principles. To achieve this outcome, we have seen the group generally adhere to a playbook designed to help scale its operations, limit forensic evidence in victim environments, and make post-exploitation activity hard to detect (see Figure 3).
- Once inside a network, APT44 is highly judicious about deploying its most advanced, and likely most costly to develop, tools. When custom malware is needed, APT44 typically deploys lightweight tools that are expendable and do not pose any significant attrition to the group's overall capabilities when used or exposed.
- APT44 almost certainly relies on a diverse set of Russian companies and criminal marketplaces to source and sustain its more frequently operated offensive capabilities.
  - Leaked documents from Russian company NTC Vulkan detailed project requirements for a framework used to enable cyber operations contracted by APT44's parent military unit.
  - We also assess that at least one additional Russian cybersecurity company has provided direct operational support to APT44's operations in Ukraine.
  - Since Russia's re-invasion of Ukraine in early 2022, we have observed a relative increase in APT44's use of tools and bulletproof hosting infrastructure acquired from criminal marketplaces. We assess that APT44 has likely long viewed criminally sourced tools and infrastructure as a latent pool of disposable capabilities that can be operationalized on short notice without immediate attributive links to its past operations.

Figure 3. Phases of Activity Commonly Observed in APT44 Operations

Moscow's Primary Cyber Sabotage Unit

Over the past decade, APT44 has established itself as Russia's preeminent cyber sabotage unit. As an arm of Russia's military, it has been responsible for the majority of the GRU's cyber-enabled sabotage in Ukraine stretching back to the initial invasion of the country in 2014. However, APT44's attack operations are not limited to military objectives and also span Russia's wider national interests, such as the Kremlin's political signaling efforts, responses to crises, or intended non-escalatory responses to perceived slights to Moscow's stature in the world.

- Since Russia's re-invasion of Ukraine in February 2022, APT44 has been responsible for almost all of the disruptive and destructive cyber attacks against Ukrainian CIKRs that Mandiant has responded to. We assess with high confidence it is the primary cyber attack unit both within the GRU and across all Russian state-sponsored cyber units.
- Since at least 2015, APT44 has operated and advanced a set of attack capabilities intended to disrupt industrial control and safety systems with the potential to cause significant physical damage. Since Russia's reinvasion, further advancements in APT44's cyber-physical attack capabilities have been observed, including a new variant of Industroyer and Operational Technology (OT)-specific living-off-the-land attack capabilities abusing a native MicroSCADA binary. While operations to date have primarily targeted Ukraine's energy grid, the underlying technologies exploited hold the potential to impact a wider range of sectors including railways, seaports, airports, and hospitals.
- APT44 has also periodically engaged in cyber sabotage operations intended to signal bilateral displeasure, retaliate for political grievances, or otherwise signal the weight of the threat posed by Russia's cyber program. For example:
  - In June 2017, APT44 deployed ETERNALPETYA (aka NotPetya), a wiper disguised as ransomware, timed to coincide with Ukraine's Constitution Day marking its sovereignty and independence from Russia.
  - In February 2018, APT44 used SOURGRAPES (aka OlympicDestroyer) destructive malware against IT systems during the opening ceremony of the Pyeongchang Olympic Games as likely retaliation for Russia's doping suspension. According to the UK government, preparations were also carried out to disrupt the 2020 Summer Olympics in Tokyo before they were postponed.
  - In October 2022, a cluster believed to be APT44 with medium confidence deployed PRESSTEA (aka Prestige) ransomware against logistics entities in Poland and Ukraine, likely to signal its ability to threaten supply lines transiting lethal aid to Ukraine. Notably, this operation is a rare instance where APT44 has shown a willingness to use a disruptive capability intentionally against a NATO member country, and reflects the group's penchant for risk taking.

Due to its history of aggressive cyber attacks across political and military contexts, we judge APT44 to present a persistent, high severity threat to governments and critical infrastructure operators globally where Russian national interests intersect. The threat of future disruptive or destructive cyber operations likely extends to individuals or entities involved in war crimes investigations or other inquiries into the Russian Federation's transgressions in Ukraine.

We also judge APT44 to present a significant proliferation risk for new cyber attack concepts and methods. Continued advancements and in-the-wild use of the group's information technology (IT) and OT cyber attack capabilities have also likely lowered the barrier of entry for other state and non-state actors to replicate and develop their own cyber attack programs. Russia itself is almost certainly alert to and concerned about this proliferation risk, as Mandiant has observed Russian cybersecurity entities exercise their ability to defend against categories of disruptive cyber capabilities originally used by APT44 against Ukraine.

APT44's Wartime Cyber Operations

APT44 has aggressively pursued a multi-pronged effort to help the Russian military gain a wartime advantage with its cyber operations. Of the Russian government-backed cyber groups that we have tracked contributing to Russia's military campaign in Ukraine, APT44 has and continues to play the most central role, seeking to advance Moscow's war aims in multiple distinct ways. The group's operational focus and methods have adapted significantly in the second year of the war to support Russia's evolving theory of victory, with increasing emphasis placed on military-relevant targets and tactical intelligence collection.

- **Disruptive Operations:** APT44 is responsible for an intensive campaign of cyber disruptions stretching from invasion day in February 2022 to present. The group has aggressively deployed wiper malware against a mix of civilian and military targets, and has attempted to make the effects of the war felt beyond the front lines in the day-to-day lives of Ukrainians.
- **Military Enablement:** APT44 has also increasingly conducted espionage likely intended to enable Russian conventional military operations. These operations appear to focus on mobile networks, devices, applications and other technologies that could help to intercept communications and gain tactical and operational battlefield advantages.
- **Information Operations:** APT44 has used front personas embedded in the pro-Russian Telegram ecosystem to attempt to shape the information environment and draw attention to the alleged "impact" of select cyber operations.

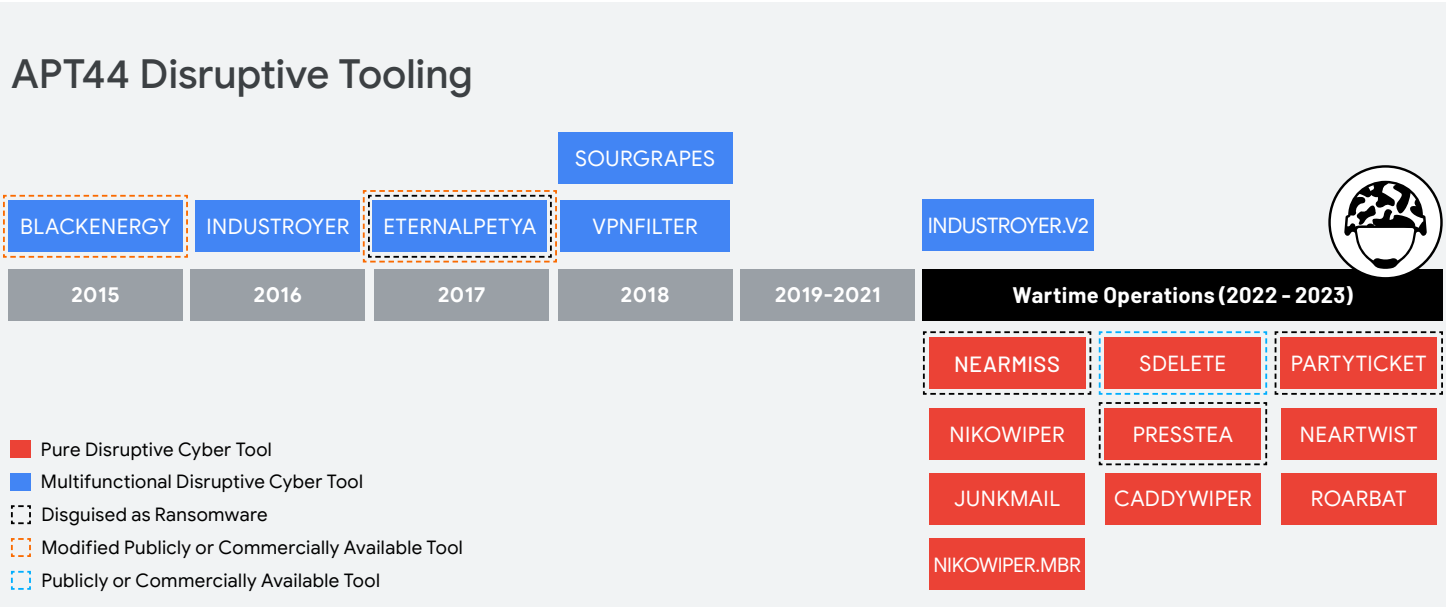


Figure 4. Categories of Disruptive Malware Used by APT44

Disruptive Operations Against Ukrainian Critical Infrastructure

Mandiant has tracked an intensive campaign of cyber attacks against Ukrainian entities by APT44 that reflect its primary mandate. These disruptive cyber operations have surpassed the scale, scope, and intensity of the group's operations conducted in the war's eight prior years, and have incorporated a wide arsenal of different disruptive or destructive malware families.

- APT44's disruptive activities have occurred in punctuated phases, mirroring the main stages of the war. Gaps between waves of disruptive activity have likely provided necessary windows to retool and replenish access to operationally relevant targets.
- Targets of APT44's disruptive activity have primarily been government networks and critical infrastructure operators, with an emphasis on Ukraine's energy sector. We continue to see malware delivery operations seeking access to energy sector targets from a subcluster of APT44 activity tracked by CERT-UA as UAC-0099.
- We assess with high confidence that, in specific operations, APT44 has coordinated the timing of these cyber attacks with conventional military activity, such as kinetic strikes or other forms of sabotage, in order to achieve joint military objectives in Ukraine. This repeated pattern of activity indicates either unity of command or operational coordination with other elements of Russia's military.
  - For example, in October 2022, APT44 disrupted IT and OT systems at a power distribution entity in the midst of Russia's winter campaign of military and drone strikes targeting Ukraine's energy grid. Notably, this activity aligns with Microsoft's independent analysis identifying a similar pattern of coordination between APT44 and other elements of the Russian military in the same timeframe.

Mandiant has previously written about APT44's shift to pure disruptive tools as a strategy to sustain its wartime tempo of operations. In furtherance of this arsenal management strategy, the second year of the war has seen the group progress from "low equity" to "no equity" tooling, abusing common utilities and publicly available tools like SDELETE, WinRAR, or native MicroSCADA binaries instead of custom-developed tools to achieve disruptive objectives.

- As the war has progressed, Mandiant has also observed APT44 rely more heavily on open source tooling (e.g. webshells such as WEEVELY and REGEORG.NEO and tunnelers such as CHISEL) to gain and further access to networks preceding its disruptive activity.
- The ready availability of these open source tools, variants of which can be created on-demand, has almost certainly provided the group an expendable reserve of new malware to cycle into its disruptive operations, helping to trivially replenish variants exhausted through prior use.

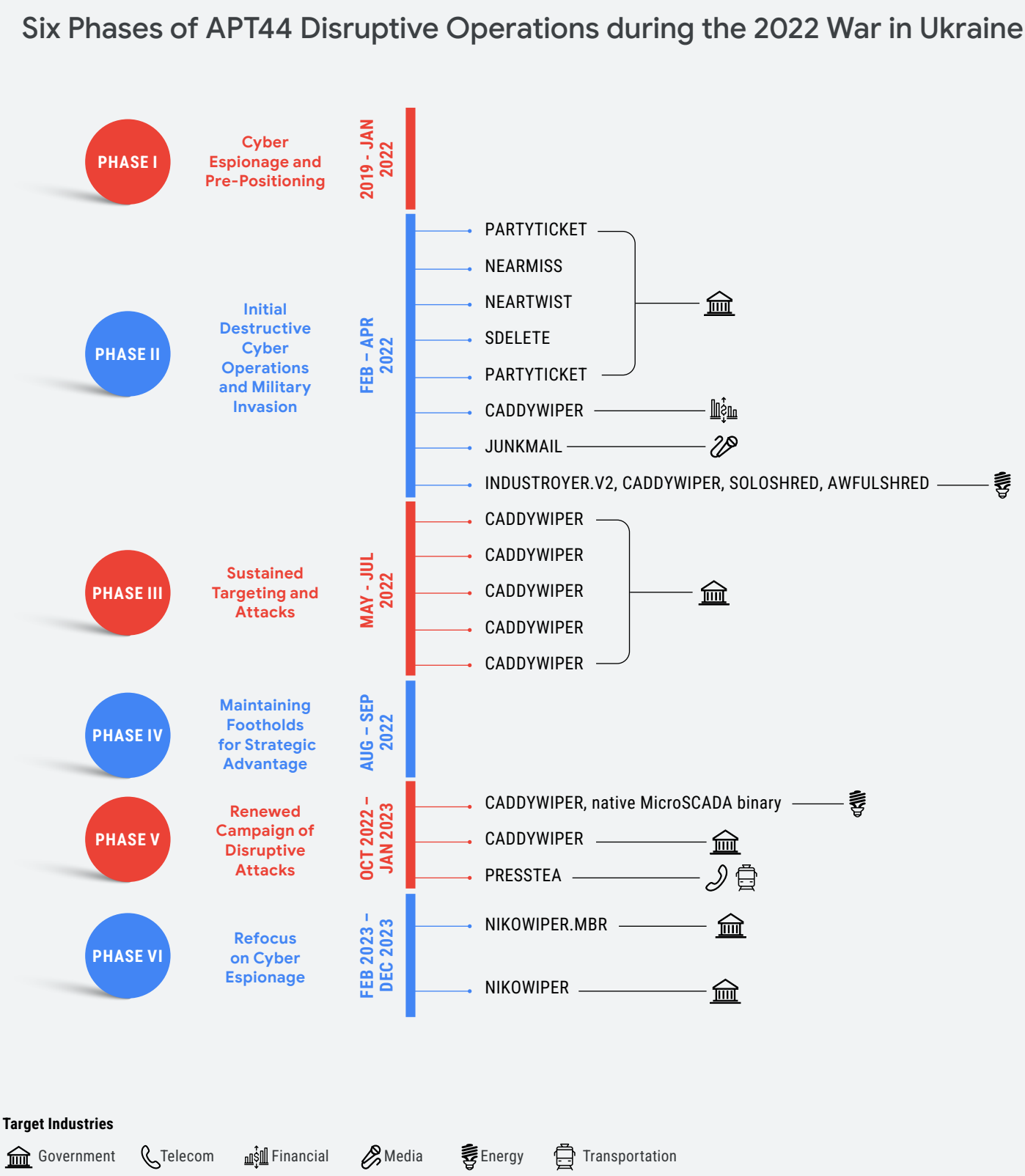


Figure 5. APT44 Disruptive Operations Against Ukraine

Espionage Operations for Military Enablement

In the second year of Russia's re-invasion, we have also seen a relative increase in APT44's espionage activity to support battlefield reconnaissance and other tactical military needs. This activity has included an apparent focus on communication systems and mobile devices, and is part of a wider transition amongst Russian military-linked actors to attempt to collect tactically relevant information from networks, devices, and applications used by the Ukrainian military.

- Extending back to at least April 2023, APT44 has provisioned infrastructure for use by likely forward-deployed Russian military forces to exfiltrate encrypted Telegram and Signal communications from mobile devices captured on the battlefield.
  - Related infrastructure contains step-by-step Russian language instructions on how to link the victim's chat applications to actor-controlled infrastructure (See Figure 7). In order to follow these instructions, an operator would almost certainly require physical access to the devices being paired.
  - The infrastructure also contains a link to contact an APT44 developer on an actor-controlled Telegram account, indicating efforts to provide troubleshooting and support to non-technical operators, such as forward deployed Russian military units in Ukraine.
  - As noted by Google's TAG, this operation's infrastructure and tooling also contained derogatory language towards Ukrainians, providing a lens into the mindset of APT44's operators as they support Russia's military campaign.
- With drones becoming increasingly crucial for battlefield success, APT44 has also conducted multiple phishing waves targeting organizations involved in drone manufacturing and logistics. Conforming with APT44's tendency to use criminally-sourced malware, this activity exploited a known WinRAR vulnerability to deliver a SMOKELOADER dropper that subsequently loaded RADTHIEF (aka Rhadmanthys Stealer) in-memory.
- We have also observed a surge in APT44 activity focused on gaining access to internet service providers and telecommunications entities providing mobile connectivity to Ukrainian civilians and military personnel. As highlighted by CERT-UA, these APT44 operations have periodically been used to enable disruptive activity as well.
- In August 2023, multiple governments disclosed an additional espionage-focused capability, "Infamous Chisel," operated by APT44 to collect information from Android devices, including system device information, commercial application information, as well as information from applications specific to the Ukrainian military.

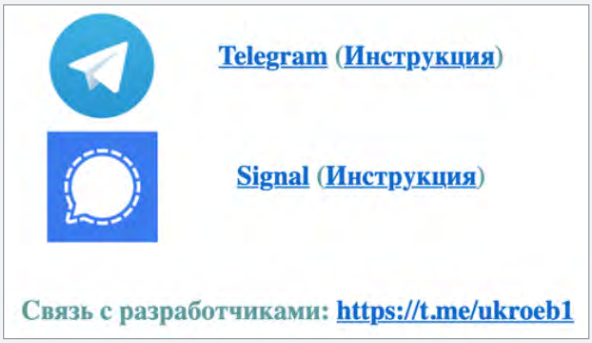


Figure 6. Links and instructions for Signal and Telegram exploitation

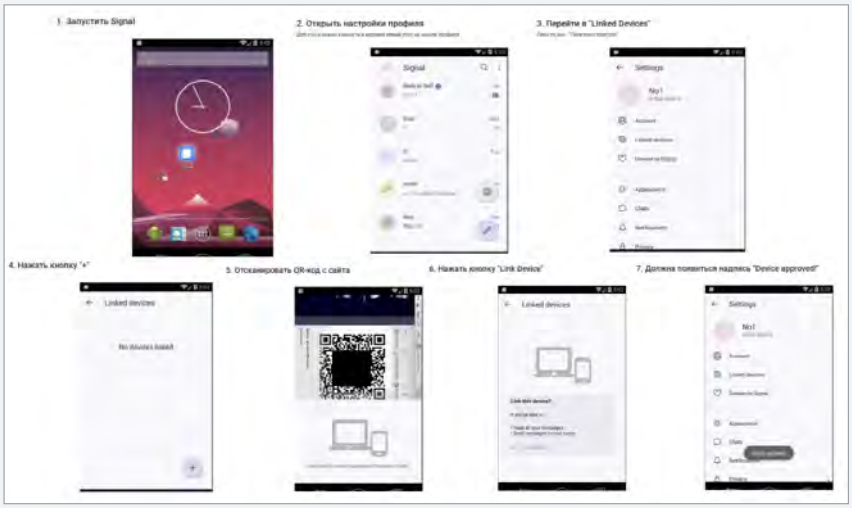


Figure 7. APT44 Provisioned Instructions for Linking Signal Accounts via QR Code

Information Operations Amplifying Cyber Activity

A particular feature of APT44's approach to cyber operations over the years has been its emphasis on attempting to generate second-order psychological effects to augment its espionage and sabotage activities. These efforts have evolved since Russia's re-invasion of Ukraine, with APT44 leveraging a series of front personas primarily on Telegram to publicly claim credit for data leaks and disruptive operations. Beyond a crude attempt to maximize its operational impact, we assess that these follow-on information operations are likely intended by APT44 to serve multiple wartime objectives. These aims include priming the information space with narratives favorable to Russia, generating perceptions of popular support for the war for domestic and foreign audiences, and making the GRU's cyber capabilities appear more potent through exaggerated claims of impact.

APT44 relies upon conventional information operations methods to achieve its wider objectives. The group's efforts are primarily focused on hack-and-leak or attack-and-leak operations, where sensitive documents or other "proof" of preceding cyber operations are posted primarily to Telegram to draw attention to their alleged "impacts". Consistent with the group's pre-war activity leveraging personas such as Anonymous Poland and Guccifer 2.0, APT44 continues to cultivate hacktivist identities as assets for its follow-on information operations. It has cycled through at least three primary hacktivist-branded Telegram channels to claim responsibility for its wartime disruptive operations: **XakNet Team, CyberArmyofRussia_Reborn[1],** and **Solntsepek.**

- We assess that APT44 continues to use these specific channels due to their established followings and their positions of influence in the wider pro-Russian Telegram ecosystem, and suspect that the GRU has played a role in cultivating their prominence over time. Although the channels are operated in parallel, they do not post the same content concurrently.
- APT44's exact relationship and control over each of these front personas likely varies. However, we have observed the closest operational relationship between APT44 and CyberArmyofRussia_Reborn (Russian: Народная СуберАрмия) and judge that the operators behind APT44 have the ability to direct and influence CyberArmyofRussia_Reborn's activity across multiple platforms.
  - Google's TAG observed CyberArmyofRussia_Reborn's YouTube channel being created from infrastructure attributed to APT44. The YouTube channel received minimal engagement and was terminated upon identification.
  - Mandiant has observed known APT44 infrastructure used to exfiltrate data from victims later leaked in the CyberArmyofRussia_Reborn Telegram channel, as well as egress to Telegram in close temporal proximity to the persona's posted claims.
  - In one case, a series of APT44 operator errors resulted in CyberArmyofRussia_Reborn's claims on Telegram preceding the network attack they referenced.
  - These patterns of interaction align with TAG's assessment that CyberArmyofRussia_Reborn is created and controlled by APT44.
  - Prior to rebranding as a "hacker group" in 2023 and claiming responsibility for APT44 disruptive cyber operations, Solntsepek (Russian: Солнцепек) conducted a long-term campaign of leaking personally identifiable information from Ukrainian military and security personnel, indicating the persona's likely established relationship with the GRU.
  - Mandiant assesses that, since its rebranding, Solntsepek has likely been used as a primary vehicle to claim responsibility for and leak stolen information from APT44 disruptive cyber attacks.
  - Solntsepek's posts have mirrored APT44's operational focus in the second year of the war, claiming cyber attacks on military-relevant targets more often than previous APT44 front personas.

Efforts at follow-on amplification of APT44 associated Telegram posts appear to be largely constrained to cross-posting within established pro-Russian Telegram communities. For example, JokerDNR played a significant role in amplifying Solntsepek when the Telegram channel first launched and then again as it rebranded into a hacker group. While we have observed limited attempts to break into other communities through defacements, we suspect the group primarily relies upon organic media coverage for reach and credibility rather than investing resources to spread its messaging on other platforms. Mandiant does not currently attribute the JokerDNR persona to a specific threat group or sponsor.

[1] Mandiant had previously attributed XakNet and CyberArmyofRussia_Reborn activity to APT28 based on a case of cohabitation where APT28 and APT44 were both operating in the same network. Re-analysis of the relevant incident data allowed us to parse the two sets of overlapping activity and link the CyberArmyofRussia_Reborn-associated intrusion activity to APT44 with high confidence

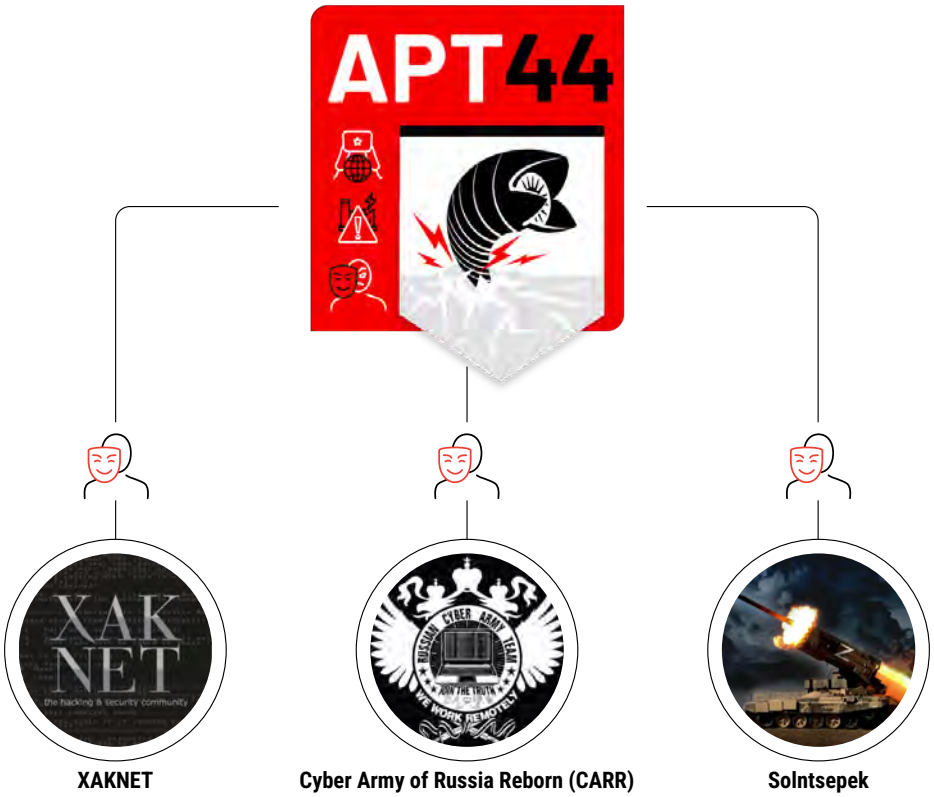


Figure 8. Hacktivist Telegram Personas Associated with APT44

CyberArmyofRussia_Reborn Video Content Claims Manipulation of of US and European Critical Infrastructure OT Assets

A majority of the attack-and-leak activity that Mandiant has tracked from GRU linked Telegram personas has centered on Ukrainian entities. However, CyberArmyofRussia_Reborn's claimed intrusion activity has not been so limited.

- Between 17 and 18 January 2024, the group's Telegram channel posted videos taking credit for the manipulation of human machine interfaces (HMI) controlling operational technology (OT) assets at Polish and U.S. water utilities.
- On 02 March 2024, the group posted an additional video claiming to disrupt electricity generation at a French hydroelectric facility by manipulating water levels.
- Each of the videos posted by CyberArmyofRussia_Reborn appear to show an actor haphazardly interacting with interfaces controlling the respective water or hydroelectric facilities' OT assets.

Mandiant cannot independently verify the above claimed intrusion activity or its links to APT44 at this time. However, we note that officials from the affected U.S. utilities later publicly acknowledged incidents at entities advertised as victims in the CyberArmyofRussia_Reborn video.

- Approximately two weeks after the Telegram post taking credit for the U.S. targeting, a local official publicly confirmed a "system malfunction" that led to a tank overflowing at one of the claimed victim facilities. This activity was reportedly part of a series of cyber incidents impacting multiple local U.S. water infrastructure systems that stemmed from "vendor software they use that keeps their water systems remotely accessible".

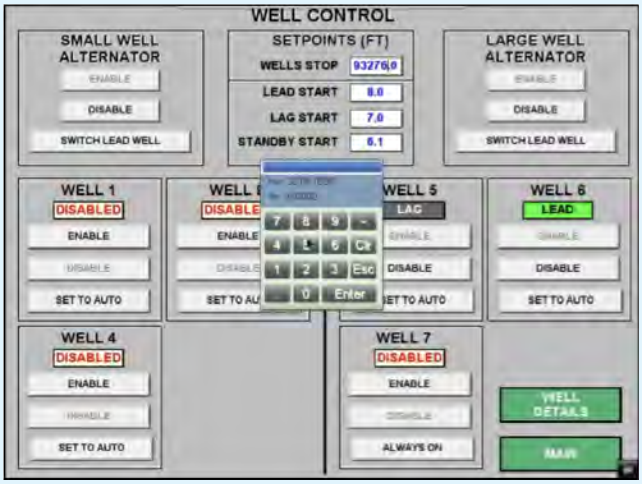


Figure 9. CyberArmyofRussia_Reborn video screenshot showing manual manipulation of well control inputs

Takeaways

APT44 continues to present one of the widest and high severity cyber threats globally. It has been at the forefront of the threat landscape for over a decade and is responsible for a long list of firsts that have set precedents for future cyber attack activity. The combination of APT44's high capability, risk tolerance, and far-reaching mandate to support Russia's foreign policy interests places governments, civil society, and CIKR operators around the world at risk of falling into the group's sights on short notice. Patterns of historical activity, such as efforts to influence elections or retaliate against international sporting bodies, suggests there is no limit to the nationalist impulses that may fuel the group's operations in the future.

Despite its bias for action and emphasis on psychological effect, APT44 has shown itself to be patient, resourceful, and able to remain undetected for long periods of time in victim environments. The group's playbook is almost certainly tailored to carry out intrusions undetected, and its use of both open source and criminally-sourced malware can often result in activity being disregarded as a commodity threat. Organizations at high risk of being targeted by APT44 should prioritize detections of LOTL techniques and carefully investigate instances of commercially available malware as potential APT44 activity. Responses to APT44 should also consider the group's sensitivity to counterintelligence risk. This is an actor that is highly aware of incident response and detection efforts, and, in certain cases, mitigation efforts may drive an intrusion toward disruptive activity.

As Russia's war continues, we anticipate Ukraine will remain the principal focus of APT44 operations. However, as history indicates, the group's readiness to conduct cyber operations in furtherance of the Kremlin's wider strategic objectives globally is ingrained in its mandate. We assess that changing Western political dynamics, future elections, and emerging issues in Russia's near abroad will continue to shape APT44's operations for the foreseeable future.

Protecting The Community

As part of our efforts to combat serious threat actors, TAG uses the results of our research to improve the safety and security of Google's products. Upon discovery, all identified websites and domains are added to Safe Browsing to protect users from further exploitation. TAG also sends all targeted Gmail and Workspace users government-backed attacker alerts notifying them of the activity and encourages potential targets to enable Enhanced Safe Browsing for Chrome and ensure that all devices are updated. Where possible, Mandiant sends victim notifications via the Victim Notification Program.

To protect high risk user accounts, we offer the Advanced Protection Program (APP), which is our highest form of account security and has a strong track record protecting users. If you are a Google Chronicle Enterprise+ customer, Chronicle rules were released to your Emerging Threats rule pack, and IOCs listed in this report are available for prioritization with Applied Threat Intelligence.

We are committed to sharing our findings with the security community to raise awareness, and with companies and individuals that might have been targeted by these activities. We hope that improved understanding of tactics and techniques will enhance threat hunting capabilities and lead to stronger user protections across the industry.

Technical Annex

APT44 Indicators of Compromise

For IOCs, please see our VT Collection.

Malware

This section includes malware Mandiant observed APT44 using since 2018, with the exception of ETERNALPETYA (aka NotPetya), which was deployed by APT44 in 2017. We have split this section into three: custom malware unique to APT44, malware that is publicly or commercially available but modified and customized by APT44, and publicly or commercially available malware used by APT44.

Custom

Malware	Role	Description
ARGUEPATCH	Launcher	ARGUEPATCH is a malicious launcher that decrypts a file on disk using a trivial XOR algorithm and executes a second stage payload in memory.
AXETERROR	Backdoor	AXETERROR is a backdoor written in Go. Upon startup, the malware creates persistence as either a cron job or a system startup script. AXETERROR communicates over HTTPS and supports the following commands: update or delete itself, download or upload files to C2, execute shell commands, set proxy configuration, update C2, and update beacon interval.
BACKORDER	Downloader	BACKORDER is a downloader written in Go which targets Windows machines. It downloads and executes a second stage payload from a remote server. BACKORDER is usually delivered within trojanized installer files and is hard coded to execute the original setup executable.
BACKORDER.V2	Downloader	BACKORDER.v2 is a downloader written in Go that targets the Windows environment. It downloads and executes a second stage payload from a remote server. The malware can set the %TEMP% directory as an excluded folder from Windows Defender before downloading a zip file to that folder from its C2 server. The malware unzips the downloaded zip file and executes the file inside.
BRUSHPASS	Webshell	BRUSHPASS is a Webshell written in C# which provides a threat actor with the means to execute commands, alter victim firewall configurations, upload files to the victim device, perform directory listing, file deletion and file collection. Additional capabilities in some BRUSHPASS samples include the collection of the current page URL as an absolute path, both uploading and downloading of files, and opening ports on the victim machine.
CADDYWIPER	Wiper	CADDYWIPER is a disruptive file wiper written in C which enumerates the file system physical drives and overwrites both file content and partitions with null bytes. CADDYWIPER has both executable and shellcode variants.
COLDWELL	Dropper	COLDWELL is a dropper written in C that contains an encrypted and embedded payload. Upon execution COLDWELL generates a random filename to write an embedded payload, configures persistence, and blends the next-stage timestamp with a legitimate file on disk.
EARLYBLOOM	Backdoor	EARLYBLOOM is a backdoor written in C++ that communicates over HTTPS. Supported backdoor commands include shell command execution, file transfer, file execution, and uninstall.
EXARAMEL	Backdoor	EXARAMEL is a backdoor that is capable of encrypting and exfiltrating files from a configured directory as well as receiving and executing commands from C2. EXARAMEL stores its configuration, structured as XML, in the registry. The configuration defines its C2, exfiltration directory, proxy, and beacon interval. EXARAMEL is capable of executing the following tasks: launch a process, create a file, upload a file, execute a shell command, and execute a VB Script.
FACEFISH	Dropper	FACEFISH is a dropper, which releases a rootkit, and its main function is determined by the rootkit module, which works at the Ring3 layer and is loaded using the LD_PRELOAD feature to steal user login credentials by hooking ssh/sshd program related functions. FACEFISH also supports some backdoor functions and supports pretty flexible configuration, uses Diffie-Hellman exchange keys, Blowfish encrypted network communication, and targets Linux x64 systems. The main functions of FACEFISH are: upload device information, stealing user credentials, bounce Shell and execute arbitrary commands.
FAIRROOT	Dropper	FAIRROOT is a VBScript macro used to deliver an encoded payload usually decoded using a fixed string as a key. FAIRROOT is capable of determining if the system is running in a sandbox.
FELIXROOT	Backdoor	FELIXROOT is a memory-only DLL backdoor that is capable of system reconnaissance, data exfiltration, and remote code execution. All communications, including exfiltrated data, are AES-encrypted with one of two hard-coded public keys, and sent back to the C2 server via HTTP or HTTPS.
FIZZLESHELL	Webshell	FIZZLESHELL is a PHP webshell that employs well written cryptographic code to obfuscate three supported commands. These commands are invoked by setting a cookie to a specific value. FIZZLESHELL sends its HTTP(S) POST data with the content of a MIME message to a hardcoded C2 server.
FREETOW	Memory-Only Dropper	FREETOW is an in-memory dropper for a shellcode payload. FREETOW has been identified as a payload patched into legitimate Microsoft applications. FREETOW contains an anti-emulation technique where it requires the first character of the command line argument to be "z" otherwise the application will crash. FREETOW has been identified loading a custom variant of METASPLOIT.

GOGETTER	Tunneler	GOGETTER is a tunneler written in Go that proxies communications for its C2 server using the open-source library Yamux over TLS.
ICYWELL	Backdoor	ICYWELL is a backdoor written in C++ that gives a threat actor a reverse shell, executes arbitrary commands, can write and read files, and in some instances updates itself on host.
ILLICITORDER	Dropper	ILLICITORDER is a dropper written in C++ which contains an XOR and Base64-encoded second stage payload. Upon execution, ILLICITORDER drops the second stage payload to disk and executes it alongside a legitimate installer. ILLICITORDER is usually embedded into trojanized software installation media.
INDUSTROYER	Disruptive Malware Framework	INDUSTROYER is a modular malware framework that is designed to survey and manipulate power grid control systems. Included in the framework are four modules that issue commands to open (and close) circuit breakers, a wiper module designed to search for and overwrite several control system specific files, and a SIPROTEC DoS module.
ITCHYSPARK	Utility	ITCHYSPARK is a lateral movement tool used to deploy the NEARMISS wiper. ITCHYSPARK enumerates the local network via various APIs, attempts an SMB connection, and is capable of port scanning.
ITCHYSPARK.SMB	Utility	ITCHYSPARK.SMB is a lateral movement tool used to copy an executable to a remote SMB server, and to execute the file.
ITCHYSPARK.SMB	Utility	ITCHYSPARK.WMI is a lateral movement tool used to copy an executable to a remote path, and execute the file as a Windows service or a standard process via WMI/COM.
JUNKMAIL	Wiper	JUNKMAIL is a .NET wiper which uses an unknown obfuscator and junk code to obfuscate control flow. JUNKMAIL enumerates each domain controller under the domain as well as drives, their respective directories, and files. It wipes files by overwriting them with null bytes.
LUCKYPIE	Launcher	LUCKYPIE is a launcher that loads and executes a DLL from its resource section. The malware is embedded into the zlib library code and exports many zlib functions.
NEARMISS	Wiper	NEARMISS is a master boot record (MBR) wiper that disables the Shadow Volume Copy and CrashDumps before wiping the MBR. After successful execution, the wiper will initiate a system shutdown, rendering the targeted device inoperable.
NEARTWIST	Wiper	NEARTWIST is a disruptive file wiper written in C which enumerates the device's physical drives and attempts to wipe them either directly or through overwriting the content of each file using data obtained from a pseudorandom number generator.
NEWRETURN	Memory-Only Dropper	NEWRETURN is an in-memory .Net dropper which contains an embedded binary that is decompressed and executed as the main functionality of this malware family. Some identified samples are padded with a huge number of null bytes, likely to make this sample infeasible for submission to automated analysis tools.
NIKOWIPER	Wiper	NIKOWIPER is a disruptive tool written in C that contains an embedded SysInternal's SDelete executable that is used to delete files on disk.
NIKOWIPER.MBR	Wiper	NIKOWIPER.MBR is a disruptive tool written in C that contains an embedded SysInternal's SDelete executable that is used to delete files on disk. NIKOWIPER.MBR contains additional functionality that wipes the Master Boot Record on victim devices.
PARTYTICKET	Wiper	PARTYTICKET is a disruptive file wiper written in Go that enumerates the file system and selects files to wipe based on the file extension. PARTYTICKET will then encrypt the content of the files with AES.
PENNYBAG	Dropper	PENNYBAG is a malicious macro dropper used to decode and write a payload to disk. Encoded payloads are stored as a series of byte arrays. PENNYBAG has historically been used to distribute BLACKENERGY.V2 and V3 and STRAYKEY in targeted attacks.
PRESSTEA	Ransomware	PRESSTEA is ransomware written in C++ that encrypts local files. Observed extensions for encrypted files include ".enc". PRESSTEA uses wbadmin to delete the backup catalog on a computer then wipes the volume shadow copies.
QUICKTOW	Backdoor	QUICKTOW is a lightweight backdoor written in Go that communicates via HTTP. Its supported backdoor commands include command execution, opening a new session, and disconnecting. QUICKTOW can also connect to other instances of the backdoor to forward commands.
ROARBAT	Wiper	ROARBAT is a batch disruptive wiper responsible for enumerating drives and directories and using WinRAR to delete data.
SOURGRAPES	Disruptive Malware	SOURGRAPES is a disruptive malware which is responsible for destroying files on network shares and disabling all services on a victim system.
SHARPCOFFEE	Downloader	SHARPCOFFEE is a downloader written in JavaScript which retrieves payloads via HTTP. Downloaded payloads are executed from memory using a PowerShell sub-process, and console output is uploaded to a remote server via HTTP. SHARPCOFFEE has been observed being delivered via SHARPIVORY, and subsequently downloading SHARPENTRY.
SHARPCOFFEE.VBS	Downloader	SHARPCOFFEE.VBS is a Windows downloader written in Visual Basic used to download other malware and upload data via Powershell.
SHARPENTRY	Downloader	SHARPENTRY is a downloader written in C that retrieves payloads via TCP. Details of the remote server are provided as command-line arguments. Payloads are decoded and mapped into memory, with the entry-point being determined at run-time. SHARPENTRY has been observed being deployed via SHARPCOFFEE and subsequently deploying METERPRETER.
SHARPIVORY	Dropper	SHARPIVORY is a dropper written in .NET that writes an embedded payload to disk and establishes persistence via scheduled tasks. The dropper also drops and opens a decoy Microsoft Office Word document. SHARPIVORY has been observed dropping SHARPCOFFEE.
SPAREPART	Backdoor	SPAREPART is a lightweight backdoor written in C that uses the device's UUID as a unique identifier for communications with the C2. Upon successful connection to a C2, SPAREPART will download the tasking and execute it through a newly created process.
SWEETTREAT	Utility	SWEETTREAT is a utility service which provides cryptographic functionality upon request via a named pipe or RPC. SWEETTREAT appears to represent a class of functionality that is uncommon.
VPNFILTER	Backdoor	VPNFILTER is a modular backdoor capable of a variety of commands and can extend functionality through the use of plugins.

Modified Publicly or Commercially Available

Malware	Role	Description
BLACKENERGY	Backdoor	Early BLACKENERGY malware variants were used to create distributed denial-of-service (DDoS)-focused botnets and have since evolved over time. Variants of BLACKENERGY, BLACKENERGY.V2 and BLACKENERGY.V3, are modular backdoors which have the ability to download additional conditional modules to targeted machines. BLACKENERGY.V2 and BLACKENERGY.V3 samples used by Sandworm only utilized modules to conduct espionage. Often configured to communicate with two C2 servers, these BLACKENERGY variants contain basic capabilities such as victim profiling, updating its configuration block, downloading and executing files, downloading and loading plugins, unloading and deleting plugins, and uninstalling the backdoor.
HEXCHAMBER	Builder	HEXCHAMBER is a custom implementation of open-source project, Malicious Macro Generator (MMG). The variation of this macro breaks encoded strings into binary and hex-based counterparts and concatenates them into an encoded command string later decoded with a constant. HEXCHAMBER has been used to distribute PowerShell Empire.
ETERNALPETYA	Wiper	Petya is ransomware family that is atypical in that the malware does not encrypt individual files on victims' systems, but instead overwrites the master boot record (MBR) and encrypts the master file table (MFT), which renders the system inoperable until the ransom has been paid. The malware contains a dropper, custom boot loader, and a small Windows kernel that executes additional encryption routines. The ETERNALPETYA variant of PETYA is a disruption tool capable of encrypting files, encrypting the MBR, installing a bootkit, extracting credentials, performing lateral movement, and remote exploitation via known vulnerabilities.
POWERDISCO	Utility	POWERDISCO is a Windows PowerShell script that has the capability to enumerate Group Policies Objects (GPO) using the Active Directory Service Interface (ADSI). It may only be able to find policies linked to the Root domain. POWERDISCO may have been sourced from a blog post by Medium user @pentesttas called "Discover Hidden GPO(s) on Active Directory using PS>ADSI" and then modified by the attacker.
TANKTRAP	Utility	TANKTRAP is a utility written in PowerShell that utilizes Windows group policy to spread and launch a wiper. TANKTRAP has been observed being used with NEARMISS, SDELETE, PARTYTICKET, and CADDYWIPER. TANKTRAP is likely inspired by public projects like SharpGPOAbuse and PowerGPOAbuse.
WILDDIME	Backdoor	WILDDIME is a PowerShell backdoor capable of downloading, uploading and executing files. WILDDIME has been identified deployed via a LNK file and is responsible for opening a decoy document. WILDDIME is a modified variant of the public tool HTTP-Shell, a tool developed by the developer JoelGMSec.

Publicly or Commercially Available

Malware	Role	Description
BEACON	Backdoor	BEACON is a backdoor written in C/C++ that is part of the Cobalt Strike framework. Supported backdoor commands include shell command execution, file transfer, file execution, and file management. BEACON can also capture keystrokes and screenshots as well as act as a proxy server. BEACON may also be tasked with harvesting system credentials, port scanning, and enumerating systems on a network. BEACON communicates with a C2 server via HTTP or DNS.
COLIBRI	Downloader	COLIBRI is an evasive, nimble and highly obfuscated C++ Win32 downloader that primarily downloads and executes second-stage implants or malware in-memory, COLIBRI is able to load arbitrary binary files and persist on victim systems. COLIBRI communicates with its Command-and-Control (C2) server using TLS over port 443 with communications Base64-encoded and RC4-encrypted.
DARKCRYSTALRAT	Backdoor	DARKCRYSTALRAT is a .NET-based backdoor that communicates via HTTP. Its capabilities include remote desktop, file transfer, file execution, and shell command execution. DARKCRYSTALRAT can also capture audio, screenshots, keystrokes, and camera images. Credentials stored by browsers and FTP clients are also targeted. DARKCRYSTALRAT can also compile and execute arbitrary C# code.
EMPIRE	Framework	EMPIRE is a post-exploitation framework written in PowerShell. EMPIRE is commonly used to generate a stager payload, which is responsible for downloading and executing the framework's backdoor. The backdoor communicates via HTTP and HTTPS. Supported backdoor commands include shell command execution, PowerShell execution, and file transfer. The EMPIRE backdoor can also be extended via plugins. Supported plugins include remote desktop, screenshot capture, keylogging, lateral movement, credential theft, and reconnaissance.
EMPYRE	Framework	EMPYRE is a pen-testing framework that is an OSX/Linux targeted tool inspired by Powershell Empire.
METASPLOIT	Framework	METASPLOIT is a penetration testing framework whose features include vulnerability testing, network enumeration, payload generation and execution, and defense evasion. The framework contains exploits for numerous applications and popular operating systems such as Windows, Linux, and macOS. METASPLOIT is commonly used to generate a stager payload, which is responsible for downloading and executing the framework's METERPRETER backdoor.
METERPRETER	Backdoor	METERPRETER is a backdoor written in C that communicates via HTTP, HTTPS, or a custom binary protocol over TCP. Supported commands include reverse shell, file transfer, file execution, keylogging, and screenshot capture. METERPRETER is generated by the METASPLOIT framework.
METERPRETER. PYTHON	Backdoor	METERPRETER.PYTHON is Python implementation of METERPRETER. This version provides an in-memory Python interpreter capable of loading Python scripts and running adhoc Python commands.
PASWEB	Webshell	PASWEB is the publicly available P.A.S. PHP webshell.

PIVOTNACCI	Tunneler	PIVOTNACCI is an open-source tunneler tool which allows pivot into the internal network by deploying HTTP agents. PIVOTNACCI allows the creation of a SOCKS server which communicates with HTTP agents. This tool was inspired by another open-source tunneler REGEORG. However, it includes some improvement, including support for balanced servers, customizable polling intervals, auto-dropping connections closed by a server or password-protected agents.
POSHC2	Framework	POSHC2 is a proxy-aware C2 framework used to aid penetration testers with red teaming, post-exploitation, and lateral movement. POSHC2 generates PowerShell, .NET, and Python backdoor payloads that communicate via HTTP or HTTPS. Supported backdoor commands include screenshot capture, keystroke capture, standard shell and PowerShell command execution, file transfer, and file execution. POSHC2 also supports collecting system credentials using a MIMIKATZ payload and can act as a proxy server.
PWNKIT	Exploit	PWNKIT is an implementation of CVE-2021-4034, used for privilege escalation.
RADTHIEF	Infostealer	RADTHIEF, AKA "Rhadamanthys Stealer", is a C++ based infostealer malware. RADTHIEF collects information including the computer name, username, system information, network information, installed programs, screenshots taken by the malware, browser data, and crypto wallet data.
REGEORG	Tunneler	REGEORG is an open-source utility used to tunnel webshell traffic.
REGEORG.NEO	Tunneler	REGEORG.NEO is an open-source utility used to tunnel webshell traffic via SOCKS proxies. It is a refactored/updated fork of the open-source REGEORG project.
REMCOM	Utility	REMCOM is a lateral movement tool written in C/C++ that reimplements the logic of the Sysinternals PsExec application. REMCOM supports creating an interactive command prompt on a remote system as well as executing files and shell commands. It can also be used to copy files to a remote system.
SMOKELOADER	Downloader	SMOKELOADER is a downloader that retrieves additional payloads via HTTP. Retrieved payloads are mapped into memory and may include plugins that expand SMOKELOADER's functionality. Capabilities added via plugins include keylogging, credential theft, and DDoS.
STOWAWAY	Backdoor	STOWAWAY is a publicly available backdoor and proxy. The project supports several types of communication like ssh, SOCKS5. Backdoor component supports upload and download of files, remote shell and basic information gathering.
WARZONE	Backdoor	WARZONE is a backdoor written in C++ that communicates via a custom protocol over TCP. Its capabilities include video and screenshot capture, remote desktop, keylogging, file transfer, file execution, and reverse shell creation. WARZONE can also extract credentials stored by web browsers, email clients, and the Windows Credential Manager.
WEEVELY	Webshell	WEEVELY is an open-source, small and polymorphic PHP webshell that can be extended over the network at runtime. It has more than 30 modules to assist administrative tasks, maintain access, provide situational awareness, elevate privileges, and spread into the target network.
WMIEXEC	Backdoor	WMIEXEC is a lightweight backdoor written in VBScript that utilizes Windows Management Instrumentation (WMI) to execute shell commands or create a reverse shell on a remote system. The remote system's hostname or IP address is specified via a command-line argument. WMIEXEC creates a file share on the remote system to store command output.
WSO	Webshell	WSO is a PHP-based webshell that functions as a backdoor. Supported backdoor commands include shell command execution, reverse shell, file transfer, arbitrary PHP code execution, SQL database management, and file management. WSO requires a password to operate.

APT44 Related Hunting Rules

Malware	Hunting Rule

BACKORDER	rule M_APT_Downloader_BACKORDER_1
	{
	meta:
	author = "Mandiant"
	description = "This rule is designed to detect on events related to BACKORDER. BACKORDER is a downloader written in GoLang which download and executes a second stage payload from a remote server. BACKORDER is usually delivered within trojanized installer files and is hard coded to execute the original setup executable."
	
	strings:
	$go = "Go build ID:" ascii wide
	$a1 = "main.proc1esar"
	$a2 = "main.obt_zip"
	$a3 = "main.un1_zip"
	$a4 = "main.primer1_paso"
	
	condition:
	uint16(0) == 0x5a4d and filesize < 10MB and all of them
	}

BACKORDER	rule M_APT_Downloader_BACKORDER_2
	{
	meta:
	author = "Mandiant"
	description = "Detects strings and sleep timer in the BACKORDER downloader"
	
	strings:
	$ = "data/setup.exe"
	$ = "http://"
	$ = {c7 04 ?? 00 CA 9A 3B C7 44 ?? 04 00 00 00 00 e8} // Sleep timer
	
	condition:
	uint16(0) == 0x5a4d and filesize < 10MB and all of them
	}

NIKOWIPER	rule M_APT_Disrupt_NIKOWIPER_1
	{
	meta:
	author = "Mandiant"
	description = "Detects code in NIKOWIPER"
	strings:
	$ = "SDelete"
	$ = "-accepteula -r -s -q " wide
	$ = {68 ?? ?? 02 00 68 }
	condition:
	uint16(0) == 0x5a4d and filesize < 2MB and all of them
	}

NIKOWIPER.MBR	rule M_APT_Disrupt_NIKOWIPER_2
	{
	meta:
	author = "Mandiant"
	description = "NikoWiper unique strings"
	strings:
	$sdelete = "SDelete is set for %d pass" ascii wide
	$dlihost = {77 00 73 00 [3] 5C 00 53 00 [3] 79 00 73 00 [3] 74 00 65 00 [3] 6D 00 33 00 [3] 32 00 5C 00 [3] 63 00 6D 00 [3] 64 00 2E 00 [3] 65 00 78 00 [3] 65 00 00 00 [3] 43 00 3A 00 [3] 5C 00 57 00 [3] 69 00 6E 00 [3] 64 00 6F 00 [3] 77 00 73 00 [3] 5C 00 64 00 [3] 6C 00 49 00 [3] 68 00 6F 00 [3] 73 00 74 00 [3] 2E 00 65 00 [3] 78 00 65 00}
	condition:
	uint16(0) == 0x5a4d and filesize < 2MB and all of them
	}

NIKOWIPER.MBR	rule M_APT_Disrupt_NIKOWIPER_MBR_1
	{
	meta:
	author = "Mandiant"
	description = "Detects code in NIKOWIPER.MBR"
	strings:
	$ = {FF 37 FF 15 [4] 8B 4D F8}
	$ = {69 C0 60 EA 00 00 50 FF 15}
	$ = {8D 85 90 FB FF FF 68 00 02 00 00 50 E8}
	$ = {68 ?? ?? 02 00 68 [4] 56 FF 15}
	$ = {68 00 00 07 00 57 FF D0}
	$ = {8B B5 9C FB FF FF C1 E6 04}
	condition:
	uint16(0) == 0x5a4d and filesize < 2MB and all of them
	}

REGEORG.NEO	rule M_Hunting_Windows_Powershell_CharSubstitutionFunction_1
	{
	meta:
	author="Mandiant"
	description="Finds a function that does a character substitution"
	strings:
	$func_strTr = /public\sString.{1,100}\(string\s.{1,100},\sstring\s.{1,100},\sstring\s.{1,100}\)\s*\{\s*String\s[\w\d_]+\s?=\s?\"\";\s*for\(int\s[\w\d_]+=0;\s?[\w\d_]+\<\s?[\w\d_]+\.Length;\s?[\w\d_]+\+\+\)\s*\{\s*int\s[\w\d_]+\s?=\s?[\w\d_]+\.IndexOf\([\w\d_]+\[[\w\d_]+\]\);\s*if\([\w\d_]+\s?!=\s?-\d\)\s*[\w\d_]+\s?\+=\s?[\w\d_]+\[[\w\d_]+\];\s*else\s*[\w\d_]+\s?\+=\s?[\w\d_]+\[[\w\d_]+\];\s*\}\s*return\s[\w\d_]+;\s*\}/is
	condition:
	filesize < 2MB and all of them
	}

REGEORG.NEO	rule M_Hunting_Windows_Powershell_HTTPHeaderParsing_1
	{
	meta:
	author="Mandiant"
	description="Finds powershell 1-liners typically used in webshells to decode an HTTP header variable and use it as a command"
	strings:
	$httpParser1 = /getstring\(convert\.frombase64string\(([\w\d_]+)?\(request\.headers\.get\(['"][\w\d_]+['"]/ ascii wide nocase
	condition:
	filesize < 2MB and all of them
	}

REGEORG.NEO	rule M_Hunting_REGEORG_Tunneller_Generic_1
	{
	meta:
	author = "Mandiant"
	strings:
	$s1 = "System.Net.IPEndPoint"
	$s2 = "Response.AddHeader"
	$s3 = "Request.InputStream.Read"
	$s4 = "Request.Headers.Get"
	$s5 = "Response.Write"
	$s6 = "System.Buffer.BlockCopy"
	$s7 = "Response.BinaryWrite"
	$s8 = "SocketException soex"
	condition:
	filesize < 1MB and 7 of them
	}

BRUSHPASS	rule M_APT_Webshell_BRUSHPASS_1
	{
	meta:
	author = "Mandiant"
	description = "Detects the string in the BRUSHPASS webshell"
	
	strings:
	$ = ".DataSource = "
	$ = "<%@ Page Language="
	$ = "RedirectStandardOutput = true;"
	$ = "UseShellExecute = false;"
	$ = ".WindowStyle = ProcessWindowStyle.Hidden;"
	$ = " -Direction inbound -Profile Any -Action Allow -LocalPort"
	
	condition:
	filesize < 5MB and all of them
	}

NEWRETURN	rule M_APT_Dropper_NEWRETURN_2
	{
	meta:
	author = "Mandiant"
	description = "Detects strings in the NEWRETURN payloads"
	
	strings:
	$a1 = "GetLists"
	$a2 = "GetBuffer"
	$a3 = "Delays"
	$a4 = "InvokeMember"
	$a5 = "Array"
	$o1 = {1f 8b 08 00 00 00 00 00 04 00}
	$o2 = "http://"
	$a6 = "Form1"
	$a7 = "mscoree.dll"
	
	condition:
	all of ($a*) and ($o1 or $o2)
	}

ILLICITORDER	rule M_APT_Dropper_ILLICITORDER_1
	{
	meta:
	author = "Mandiant"
	description = "Detects code segments in the ILLICITORDER dropper"
	
	strings:
	$push_8nn = {41 B8 3? 03 00 00 48 8B}
	$mov_8nn = {49 8B 9D 3B 03 00 00}
	$string_autorun = {C7 [3] 4f 66 66 69 C7 [3] 63 65 5c 41 c7 [3] 55 54 4f 52 c7 [3] 55 4e 2e 65 c7 [3] 78 65}
	$xor_13 = {B8 4F EC C4 4E F7 E9 C1 FA 02 8B C2 C1 E8 1F 03 D0 6B D2 0D 8B C1 2B C2}
	$xor_13_2 = {B9 0D 00 00 00 F7 F9 8B C2}
	$import_CryptStringToBinaryA = "CryptStringToBinaryA\x00"
	condition:
	uint16(0) == 0x5a4d and filesize < 10MB and ($push_8nn or $string_autorun or $mov_8nn) and ($xor_13 or $xor_13_2) and $import_CryptStringToBinaryA
	}

SPAREPART	rule M_APT_Backdoor_SPAREPART_Strings
	{
	meta:
	author = "Mandiant"
	description = "Detects the PDB and a struct used in SPAREPART"
	strings:
	$pdb = "c:\\Users\\user\\Desktop\\ImageAgent\\ImageAgent\\PreAgent\\src\\builder\\agent.pdb" ascii nocase
	$struct = { 44 89 ac ?? ?? ?? ?? ?? 4? 8b ac ?? ?? ?? ?? ?? 4? 83 c5 28 89 84 ?? ?? ?? ?? ?? 89 8c ?? ?? ?? ?? ?? 89 54 ?? ?? 44 89 44 ?? ?? 44 89 4c ?? ?? 44 89 54 ?? ?? 44 89 5c ?? ?? 89 5c ?? ?? 89 7c ?? ?? 89 74 ?? ?? 89 6c ?? ?? 44 89 74 ?? ?? 44 89 7c ?? ?? 44 89 64 ?? ?? 8b 84 ?? ?? ?? ?? ?? 44 8b c8 8b 84 ?? ?? ?? ?? ?? 44 8b c0 4? 8d 15 ?? ?? ?? ?? 4? 8b cd ff 15 ?? ?? ?? ?? }
	condition:
	(uint16(0) == 0x5A4D) and uint32(uint32(0x3C)) == 0x00004550 and
	$pdb and
	$struct and
	filesize < 20KB
	}

SPAREPART	rule M_APT_Backdoor_SPAREPART_SleepGenerator
	{
	meta:
	author = "Mandiant"
	description = "Detects the algorithm used to determine the next sleep timer"
	
	strings:
	$ = {C1 E8 06 89 [5] C1 E8 02 8B}
	$ = {c1 e9 03 33 c1 [3] c1 e9 05 33 c1 83 e0 01}
	$ = {8B 80 FC 00 00 00}
	$ = {D1 E8 [4] c1 E1 0f 0b c1}
	condition:
	all of them
	}

QUICKTOW	rule M_APT_Backdoor_QUICKTOW_1
	{
	meta:
	author = "Mandiant"
	description = "Hunting rule looking for QUICKTOW by strings."
	strings:
	$useragent = {4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57 4f 57 36 34 29 20 41 70 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 30 31 2e 30 2e 34 39 35 31 2e 35 34 20 53 61 66 61 72 69 2f 35 33 37 2e 33 36}
	$s1 = "NewErgoClientSessions" ascii nocase
	$s2 = "SetDisconnected" ascii nocase
	$s3 = "IsDisconnected" ascii nocase
	$s4 = "getDelay" ascii nocase
	$s5 = "setDelay" ascii nocase
	$s6 = "getMessagesFromServer" ascii nocase
	$s7 = "getOneMessageFromServer" ascii nocase
	$s8 = "getMessagesFromServer" ascii nocase
	
	condition:
	((uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) or uint16(0) == 0x457f) and filesize < 20MB and $useragent and (6 of ($s*))
	}

QUICKTOW

```
rule M_APT_Backdoor_QUICKTOW_2
{
  meta:
        author = "Mandiant"
        description = "Function names matching QUICKTOW"
  strings:
        $go = "Go build" ascii wide
        $str1 = "main.(*Client).Auth" ascii wide
        $str2 = "main.(*Client).Disconnect" ascii wide
        $str3 = "main.(*Client).Disconnect.func1" ascii wide
        $str4 = "main.(*Client).IsDisconnected" ascii wide
        $str5 = "main.(*Client).MakeMessage" ascii wide
        $str6 = "main.(*Client).NewErgoClientSessions" ascii wide
        $str7 = "main.(*Client).NewHTTPHandler" ascii wide
        $str8 = "main.(*Client).NewSession" ascii wide
        $str9 = "main.(*Client).ProcessingMessages" ascii wide
        $str10 = "main.(*Client).RandomSleep" ascii wide
        $str11 = "main.(*Client).SetDisconnected" ascii wide
        $str12 = "main.(*Client).getDelay" ascii wide
        $str13 = "main.(*Client).getMessagesFromServer" ascii wide
        $str14 = "main.(*Client).getOneMessageFromServer" ascii wide
        $str15 = "main.(*Client).setDelay" ascii wide
        $str16 = "main.(*ErgoHTTPHandler).Lock" ascii wide
        $str17 = "main.(*ErgoHTTPHandler).Unlock" ascii wide
        $str18 = "main.(*ErgoHTTPHandler).doRequest" ascii wide
        $str19 = "main.(*Session).IsAlive" ascii wide
        $str20 = "main.(*Session).Lock" ascii wide
        $str21 = "main.(*Session).MakeMessage" ascii wide
        $str22 = "main.(*Session).ResetAlive" ascii wide
        $str23 = "main.(*Session).SetAlive" ascii wide
        $str24 = "main.(*Session).Unlock" ascii wide
        $str25 = "main.(*Session).getDelay" ascii wide
        $str26 = "main.(*Session).getMessagesForSession" ascii wide
        $str27 = "main.(*Session).getOneMessageForSession" ascii wide
        $str28 = "main.(*Session).handle" ascii wide
        $str29 = "main.(*Session).handle.func1" ascii wide
        $str30 = "main.(*Session).processingMessage" ascii wide
        $str31 = "main.(*Session).setDelay" ascii wide
        $str32 = "main.(*Sessions).Add" ascii wide
        $str33 = "main.(*Sessions).Range" ascii wide
        $str34 = "main.GetHash" ascii wide
        $str35 = "main.NewAddress" ascii wide
        $str36 = "main.NewClient" ascii wide
  condition:
        (uint16(0) == 0x5a4d or uint16(0) == 0x457f) and filesize < 20MB and $go and 30 of ($str*)
}
```

EARLYBLOOM

```
rule M_APT_Backdoor_EARLYBLOOM_1
{
 meta:
   author = "Mandiant"
   description = "Code blocks indicative of EARLYBLOOM."
 strings:
   $code1 = { 8B 4D ?? 3B 4D ?? 73 24 8B 55 ?? 8B 45 ?? 8B 0A 33 48 ?? 8B 55 ?? 89 0A 8B 45 ?? 83 C0 ?? 89 45 ?? 8B 4D ?? 83 C1 ?? 89 4D ?? EB CB
}
   $code2 = { 83 7D ?? 00 7C 20 8B 45 ?? 83 E0 ?? 83 E8 ?? F7 D0 89 45 ?? 8B 4D ?? D1 E9 8B 55 ?? 23 55 ?? 33 CA 89 4D ?? EB D1 }
 condition:
   uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and all of them
}
```

EARLYBLOOM	rule M_APT_Backdoor_EARLYBLOOM_2
	{
	meta:
	author = "Mandiant"
	description = "Hunting rule looking for EARLYBLOOM, a backdoor written in C++ that communicates over HTTPS."
	
	strings:
	$a1 = "bsd.bst" xor
	$a2 = "bat.bdt" xor
	$a3 = "chkdsk.exe" xor
	$a4 = "Windows check disk" xor
	$a5 = "https://" xor
	
	condition:
	uint16(0) == 0x5a4d and uint32(uint32(0x3C)) == 0x00004550 and filesize <300KB and 3 of ($a*)
	}

TANKTRAP	rule M_Hunting_TANKTRAP_XML_1
	{
	meta:
	author = "Mandiant"
	description = "Strings associated TANKTRAP XML GPO policy"
	strings:
	$r1 = /ImmediateTask clsid=\"\{9F030D12-DDA3-4C26-8548-B7CE9151166A\}\" name=\"[a-zA-Z]{5}\"/
	condition:
	filesize < 5MB and all of them
	}

TANKTRAP	rule M_Hunting_TANKTRAP_PS1_1
	{
	meta:
	author = "Mandiant"
	description = "Strings associated TANKTRAP PowerShell"
	strings:
	$s1 = "ImmediateTaskV2 clsid = \"{9756B581-76EC-4169-9AFC-0CA8D43ADB5F}\""
	$s2 = "SharpGPOAbuse"
	$s3 = "GuidExtension \"AADCED64-746C-4633-A97C-D61349046527\""
	$s4 = "ImmediateTaskV2 clsid = \"\"{9756B581-76EC-4169-9AFC-0CA8D43ADB5F}\""
	condition:
	filesize < 5MB and 3 of them
	}

ARGUEPATCH	rule M_APT_Launcher_ARGUEPATCH_1
	{
	meta:
	author = "Mandiant"
	description = "Identifies the code used by the sleep functionality in ARGUEPATCH"
	
	strings:
	$ = {2b ?? 81 f? 00 2E 93 02}
	$ = {83 C0 18 6B C0 3C [5-12] 69 C0 60 EA 00 00}
	$ = {68 00 DD 6D 00}
	
	condition:
	filesize < 5MB and all of them
	}

ARGUEPATCH	rule M_APT_Launcher_ARGUEPATCH_2
	{
	meta:
	description = "To detect executable with patched function used to load encrypted shellcode"
	author = "Mandiant"
	strings:
	/*
	XOR loop:
	.text:004719C6 xor_loop: ; CODE XREF: PATCHED+468
	.text:004719C6 8A 01 mov al, [ecx] ; 8A 01
	.text:004719C8 33 D2 xor edx, edx ; 33 D2
	.text:004719C8
	.text:004719CA
	.text:004719CA xor_loop_inner: ; CODE XREF: PATCHED+45F
	.text:004719CA 8B 7D F8 mov edi, [ebp+String] ; 8B 7D ??
	.text:004719CD 32 04 57 xor al, [edi+edx*2] ; 32 04 57
	.text:004719D0 42 inc edx ; 42
	.text:004719D1 88 01 mov [ecx], al ; 88 01
	.text:004719D3 83 FA 10 cmp edx, 16 ; 83 FA 10
	.text:004719D6 72 F2 jb short xor_loop_inner ; 72 F2
	.text:004719D6
	.text:004719D8 FF 4D FC dec [ebp+var_4] ; FF 4D ??
	.text:004719DB 41 inc ecx ; 41
	.text:004719DC 39 5D FC cmp [ebp+var_4], ebx ; 39 5D ??
	.text:004719DF 75 E5 jnz short xor_loop ; 75 E5
	*/
	$xor_loop = {8A 01 33 D2 8B 7D ?? 32 04 57 42 88 01 83 FA 10 72 F2 FF 4D ?? 41 39 5D ?? 75 E5}
	
	condition:
	(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and
	$xor_loop
	}

ARGUEPATCH	rule M_APT_Launcher_ARGUEPATCH_3
	{
	meta:
	description = "arguepatch malware family"
	
	strings:
	$p00_0 = {85ff74??83ff???75??33db8bfbeb??a1[4]6a}
	$p00_1 = {8a064684c075??2bf23bf35e73??51}
	$p01_0 = {2bc183e0??3d[4]72??8b51??83c0???2bca83c1??83f9}
	$p01_1 = {75??eb??803d[5]74??cc68[4]e8[4]803d[5]74}
	condition:
	uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and
	(
	($p00_0 in (92000..99000) and $p00_1 in (640..8300))
	or
	($p01_0 in (170000..190000) and $p01_1 in (140000..160000))
	)
	}

FREETOW	rule M_APT_Dropper_FREETOW_1
	{
	meta:
	author = "Mandiant"
	strings:
	$hex49_add_arg_check = { 83 C1 49 88 08 FF D0 }
	$shell32_stack_string = { C7 (41 | 42 | 43 | 45 | 46 | 47) ?? 73 68 65 6C C7 (41 | 42 | 43 | 45 | 46 | 47) ?? 6C 33 32 2E C7 (41 | 42 | 43 | 45 | 46 | 47) ?? 64 6C 6C 00 }
	condition:
	filesize < 5MB and all of them
	}

FREETOW	rule M_APT_Dropper_FREETOW_2
	{
	meta:
	author = "Mandiant"
	strings:
	$push_ror13_api_hash_getcommandlinew = { 68 55 CE E0 2E }
	$push_ror13_api_hash_loadlibrary = { 68 4C 77 26 07 }
	$push_ror13_api_hash_virtualalloc = { 68 58 A4 53 E5 }
	$ror13_api_hash_commandlinetoargw = { 11 4B AF 1C }
	condition:
	filesize < 5MB and all of them
	}

FREETOW	rule M_APT_Dropper_FREETOW_3
	{
	meta:
	author = "Mandiant"
	strings:
	$func_args1 = { 6A 40 68 00 10 00 00 6A 10 6A 00 }
	$func_args2 = { 6A 40 68 00 10 00 00 68 00 00 40 00 6A 00 }
	condition:
	all of them
	}

FREETOW	rule M_APT_Dropper_FREETOW_4
	{
	meta:
	author = "Mandiant"
	description = "Patched ftp with shellcode, run with z option to launch."
	strings:
	$h1 = {0FB70983 C1498808 FFD0}
	// 0F B7 09 movzx ecx, word ptr [ecx]
	// 83 C1 49 add ecx, 49h ; 'I'
	// 88 08 mov [eax], cl
	// FF D0 call eax
	$h2 = {80CAFF2A 11881141 3BC876F4}
	// 80 CA FF or dl, 0FFh ;
	// 2A 11 sub dl, [ecx]
	// 88 11 mov [ecx], dl
	// 41 inc ecx
	// 3B C8 cmp ecx, eax
	// 76 F4 jbe short loc_1001757
	$s1 = "local-file:"
	$s2 = "xpsp2res.dll"
	$s3 = "anonymous"
	condition:
	uint16(0) == 0x5A4D and filesize < 50KB and all of them
	}

ITCHYSPARK	rule M_APT_Worm_Win32_ITCHYSPARK_1
	{
	
	meta:
	author = "Mandiant"
	description = "Looking for ITCHYSPARK samples based on opcode patterns observed on relevant functions."
	strings:
	$b1 = {5? 5? 8B ?? 8B [2] 2B ?? C1 ?? 02 8D [2] 8D [2] 8D [2] 85 ?? 7? ?? 8B ?? 8D [2] 8B ?? 33 ?? 8B ?? 4? 89 ?? 8D ?? 85 ?? 7? ?? 8B ?? 81 ?? A3 B1 29 4A 5? 5? 5? C3}
	$b2 = {6A 01 5? 6A 00 FF ?? 83 F8 ?? 0F 8? [4] 8B [2] E8 [4] 8B ?? 85 ?? 0F 8? [4] 6A 01 8D [2] 5? 5? FF ?? 85 C0 0F 8? [4] 33 ?? 89 [2] 39 ?? 0F 8? [4] 8D [2] 89 [2] 83 [2] 02 0F 8? [4] 83 [2] 04 0F 8?}
	$b3 = {5? 5? 5? 68 AE 00 00 00 6A 02 89 [2] 89 [2] FF ?? 83 ?? 6F 0F 8? [4] 8B [2] E8 [4] 8B ?? 89 [2] 85 ?? 0F 8? [4] 8D [2] 5? 5? 6A 00 68 AE 00 00 00 6A 02 FF ?? 85 ?? 0F 8?}
	$b4 = {5? 6A 65 5? 89 [2] FF 15 [4] 85 C0 0F 8? [4] 8B [2] 85 C0 0F 8? [4] [4-12] 85 ?? 0F 8? [4] 81 ?? F4 01 00 00 7?}
	condition:
	(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and all of them
	}

ITCHYSPARK	rule M_APT_Worm_Win32_ITCHYSPARK_2
	{
	meta:
	author = "Mandiant"
	description = "Looking for WMI spreader component of ITCHYSPARK (ITCHYSPARK.WMI) samples based on opcodes observed at relevant functions."\
	strings:
	$b1 = {5? 6A 03 6A 09 5? 68 [4] 68 [4] FF 15 [4] 85 C0 7? ?? FF 74 24 ?? FF 15 [4] FF 74 24 ?? FF 15}
	$b2 = {8B [5] 68 FF 01 0F 00 FF [2] 5? FF ?? 85 C0 7? ?? 32 ?? EB ?? 8B ?? E8 [4] 88 [2] 84 ?? 7? ?? 68 88 13 00 00 FF 15 [4] 68 FF 01 0F 00 FF [2] 5? FF ?? 85 C0 7?}
	$b3 = {5? [0-2] 5? 5? 5? FF 15 [4] 85 C0 7? ?? FF 15 [4] 3D 1D 04 00 00 7? ?? B0 01 EB ?? 83 ?? 1E 7? ?? 68 E8 03 00 00 FF 15 [4] 4? 8B ?? E8 [4] 83 F8 04 7? ?? EB ?? 32 C0 5? 5? C3}
	$b4 = {6A 00 FF 76 ?? FF 76 ?? FF 15 [4] 85 C0 0F 95 ?? 85 C0 7? ?? 6A 41 [0-12] 6A 44 [0-12] 6A 4D [0-12] 6A 49 [0-12] 6A 4E [0-12] 6A 24 [0-12] FF 76 [0-12] FF 76 [0-12] FF 76 ?? E8 [4] 83 C4 0C 88 [2] 6A 00 FF 76 ?? FF 76 ?? FF 15 [4] 85 C0 0F 95}
	condition:
	(uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and all of them
	}

ITCHYSPARK	rule M_APT_Worm_Win32_ITCHYSPARK_3
	{
	meta:
	author = "Mandiant"
	description = "Looking for SMB spreader component of ITCHYSPARK (ITCHYSPARK.SMB) samples based on op code patterns observed on relevant functions."
	strings:
	$b1 = { E8 [4] 5? 84 C0 7? ?? 8D ?? 24 ?? 5? 68 02 02 00 00 FF 15 [4] 85 C0 7? ?? 8D ?? 24 ?? 8D ?? 24 ?? 4? FF 15 [4] 5? FF 15 }
	$b2 = { 80 ?? 01 7? ?? 80 ?? 02 7? ?? 33 ?? B? [4] 80 ?? 01 6A 04 5? 0F 45 ?? 0F B7 ?? 33 C0 80 F? 01 0F 45 ?? 80 F? 02 7? ?? 6A 02 5? B? [4] 33 ?? 33 ?? 66 3B ?? 7? ?? 8B [2] 0F B7 ?? 8B [2] 89 [2] E8 [4] 8B ?? 83 F? 12 7? ?? 4? 66 3B ?? 7? }
	$b3 = { (68 | FF) [2-4] FF 7? ?? 68 [4] 5? E8 [4] A1 60 F0 04 10 8B ?? 89 45 ?? 66 A1 [4] 66 89 45 ?? [4-12] E8 [4] 6A 12 5? }
	$b4 = { 33 ?? 89 [2] 8B ?? 85 [4] 89 46 ?? 33 C0 89 [2] 66 39 45 ?? 7? ?? 8B [2] EB ?? 0F B7 ?? 4? 89 [2] 8B ?? 85 [4] 8B [2] [8-16] 85 ?? 7? ?? 6A 00 33 C0 4? 5? 6A 02 5? 5? FF }
	$b5 = { E8 [4] 3B ?? 7? ?? 8B [2] B? 06 02 FC 23 3B ?? 7? ?? 7? ?? 3? 05 01 28 0A 7? ?? 3? 05 02 CE 0E 7? ?? 3? 06 00 72 17 7? ?? 3? 06 01 B0 1D 7? ?? 3? 06 01 B1 1D 7? ?? 3? 06 02 F0 23 7? ?? [0-32] 3? 06 03 80 25 7? ?? 3? 0A 00 00 28 7? ?? 3? 0A 00 5A 29 7? ?? 3? 0A 00 39 38 7? ?? 3? 0A 00 D7 3A 7? ?? B? 9A 08 00 00 EB ?? }
	condition:
	(uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550) and 3 of them
	}

GOGETTER	rule M_APT_Tunneler_GOGETTER_1
	{
	meta:
	author = "Mandiant"
	description = "Hunting for GOGETTER ELF files."
	strings:
	$g1 = "go.buildid"
	$g2 = "Go build ID:"
	$g3 = "Go buildinf:"
	$proxy1 = "proxy/pkg/client.(*Client)"
	$proxy2 = "proxy/pkg/"
	$yamux = "hashicorp/yamux"
	condition:
	filesize < 25MB and uint32(0) == 0x464c457f and any of ($g*) and all of ($proxy*) and $yamux
	}

GOGETTER	rule M_APT_Tunneler_GOGETTER_2
	{
	meta:
	author = "Mandiant"
	strings:
	$s1 = "\x00github.com/hashicorp/yamux.Client\x00"
	$s2 = "\x00github.com/hashicorp/yamux.(*Session).AcceptStream\x00"
	$sb1 = { 8D ?? 24 [1-5] 89 04 24 E8 [4-5] 8B 44 24 [1-2] 8B 4C 24 [4-32] 83 ?? 03 75 0D 66 81 3? 65 6E 75 06 80 7? 02 64 7? [1-2] C7 04 24 00 00 00 00 E8 }
	condition:
	(uint32(0) == 0x464c457f) and all of them
	}

GOGETTER	rule M_APT_Tunneler_GOGETTER_3
	{
	meta:
	author = "Mandiant"
	strings:
	$sb1 = { 48 C7 ?? 24 [4] 00 10 00 00 48 C7 ?? 24 [4] 00 10 00 00 48 8D 15 [4] 48 89 ?? 24 [4] 48 8B ?? 24 ?? 48 89 ?? 24 [4] 48 C7 ?? 24 [4] FF FF FF FF 48 C7 ?? 24 [4] FF FF FF FF [32-150] 48 8D ?? 24 [4] 0F 1F 40 00 E8 [4] 48 8? ?? 0F 85 [4] 48 89 ?? 24 ?? 48 89 ?? 24 ?? 48 89 D9 48 89 C3 48 8D 44 24 ?? E8 [4] 48 89 ?? 24 ?? 48 89 ?? 24 ?? E8 [4] 48 8B 4C 24 ?? 0F 1F 40 00 48 3? ?? 7? ?? 48 8? ?? 48 8B 44 24 ?? E8 [4] 84 C0 }
	condition:
	(uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550) and (uint16(uint32(0x3C)+0x18) == 0x020B) and all of them
	}

CADDYWIPER	rule M_APT_Wiper_CADDYWIPER_1
	{
	meta:
	author = "Mandiant"
	description = "Searches code segments in CADDYWIPER"
	strings:
	// C7 45 FC 44 3A 5C 00 mov [ebp+var_4], '\:D'
	$ = {c7 ?? ?? 44 3A 5C 00}
	
	// B8 00 00 A0 00 mov eax, 0A00000h
	$ = {B8 00 00 A0 00}
	
	/*
	51 push ecx
	68 54 C0 07 00 push 7C054h
	*/
	$ = {51 68 54 C0 07 00}
	condition:
	filesize < 3MB and all of them
	}

CADDYWIPER	rule M_APT_Wiper_CADDYWIPER_1
	{
	meta:
	author = "Mandiant"
	description = "Searches for the Physical Device call within CADDYWIPER"
	
	strings:
	// LocalAlloc, push 0xa00000 and 0x40
	$ = {00 00 A0 00}
	
	$ = {43 3A 5C 55 C7 ?? ?? 73 65 72 73}
	
	$ = {C7 45 FC 44 3A 5C 00} // d:\\
	
	//$ = {68 54 C0 07 00}
	
	condition:
	all of them
	}

NEARTWIST

```
rule M_APT_Disrupt_NEARTWIST_1
{
meta:
  author = "Mandiant"
strings:
  $mersenne_alg = { D1 EA 83 E1 01 69 C9 DF B0 08 99 33 CA }
  $s1 = "PhysicalDrive" wide fullword
  $wipe_drive = { 6A 00 6A 00 6A 03 6A 00 6A 03 68 00 00 00 C0 [0-32] FF 15 [4-64] 5? 6A 00 6A 00 6A 00 6A 00 68 18 00 09 00 5? FF 15 [4-256] 68 00 00 01 00 [0-32] FF 15 }
  $wipe_file = { 6A 00 6A 00 6A 03 6A 00 6A 03 68 00 00 00 C0 [0-32] FF 15 [4-64] 5? 5? FF 15 [4-32] 6A 00 68 00 00 01 00 5? 5? E8 }
condition:
  (uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550) and all of them
}
```

NEARTWIST

```
rule M_APT_Wiper_Win32_NEARTWIST_1
{
   meta:
                author = "Mandiant"
                description = "Looking for NEARTWIST samples based on opcode patterns observed on relevant functions."
   strings:
                $b1 = {68 05 01 00 00 8D [4-6] 5? FF 15 [4] 85 C0 0F 8? [4] 3? 05 01 00 00 0F 8? [4] 8B 85 [4-5] 85 C0 0F 8?}
                $b2 = {FF 15 [4] 89 8? [4-6] B? 01 00 00 00 [4-32] C1 ?? 1E  33 ?? 69 ?? 65 89 07 6C 03 ?? 89 [6] 4? (3D|81 FA) 70 02 00 00 7? ?? B? 70 02 00 00 }
                $b3 = {6A 00 5? 68 00 00 01 00 8D ?? 24 [4] 5? 5? FF 15 [4] 85 C0 7? ?? 8B 44 24 ?? 3D 00 00 01 00 7? ?? 2B ?? 83 ?? 00 E9}
   condition:
                (uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and all of them
}
```

NEARTWIST

```
rule M_APT_Wiper_Win_NEARTWIST_1
{
   meta:
      author = "Mandiant"
      description = "Looking for NEARTWIST samples based on strings, imports, and constants for Mersenne Twister / ISAAC PRNG."
   strings:
      $b1 = { 65 89 07 6C }
      $b2 = { AD 58 3A FF }
      $b3 = { 8C DF FF FF }
      $i1 = "GetTickCount"
      $i2 = "DeviceIoControl"
      $i3 = "GetLogicalDrives"
      $i4 = "FindFirstFile"
      $i5 = "FindNextFile"
      $i6 = "WriteFile"
      $i7 = "GetDiskFreeSpaceEx"
      $i8 = "CreateThread"
      $i9 = "GetWindowsDirectory"
      $i10 = "GetTempFileName"
      $n1 = "Cleaner.exe" ascii fullword wide
      $n2 = "Cleaner.dll" ascii fullword wide
      $s1 = "PhysicalDrive" ascii fullword wide
      $s2 = "\\\\.\\" ascii fullword wide
      $s3 = "*.*" ascii fullword wide
      $s4 = "Tmf" ascii fullword wide
      $s5 = "Tmd" ascii fullword wide
   condition:
      (uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550) and any of ($n*) and all of ($s*) and all of ($i*) and all of ($b*)
}
```

PARTYTICKET	rule M_APT_Disrupt_PARTYTICKET_1
	{
	meta:
	author = "Mandiant"
	description = "Looking for PARTYTICKET samples via strings."
	strings:
	$s1 = "/403forBiden/" ascii wide
	$s2 = "/wHiteHousE/" ascii wide
	$s3 = "partyTicket." ascii wide
	$s4 = "vote_result." ascii wide
	$s5 = ".encryptedJB" ascii wide
	$f1 = "/wHiteHousE.baggageGatherings" ascii wide
	$f2 = "/wHiteHousE.primaryElectionProcess" ascii wide
	$f3 = "/wHiteHousE.GoodOffice1" ascii wide
	$f4 = "/wHiteHousE.lookUp" ascii wide
	$f5 = "/wHiteHousE.init" ascii wide
	$m1 = " Thank you for your vote! All your files, documents, photoes, videos, databases etc. have been successfully encrypted! " ascii wide fullword
	$m2 = " Now your computer has a special ID: " ascii wide fullword
	$m3 = " NOTE: Do not send file with sensitive content. In the email write us your computer's special ID (mentioned above)." ascii wide fullword
	condition:
	uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and (3 of ($s*) or 3 of ($f*) or 2 of ($m*))
	}

PARTYTICKET	rule M_APT_Disrupt_PARTYTICKET_2
	{
	meta:
	author = "Mandiant"
	description = "Looking for PARTYTICKET samples via opcode patterns observed on relevant functions."
	strings:
	$b1 = {48 83 F8 1B 0F 8D [4] 48 89 [3] 48 89 [3] 48 89 [3] 48 8D 35 [4] 0F B6 3C 06 81 FF 80 00 00 00 0F 8? [4] 48 FF C0 [16-32] E8 [16-64] E8 [8-32] E8 [4] 48 8B 44 24 ?? 48 85 C0 7? ?? [8-24] E9}
	$b2 = {48 83 F8 37 0F 8D [24-32] E8 [16-32] E8 [16-32] 48 C1 E? 04 [8-16] 7? ?? 0F B6 44 24 ?? EB ?? [8-16] E8 [4] 0F B6 44 24 ?? 84 C0 7? ?? [4-8] B8 01 00 00 00 E9 }
	$b3 = {3D 77 69 6E 64 0F 85 [4-12] 66 3D 6F 77 0F 85 [4-12] 3C 73 0F 85 [4] E8 [4] [8-24] 31 ?? EB}
	condition:
	uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and all of them
	}

NEARMISS	rule M_APT_Distupt_NEARMISS_1
	{
	meta:
	author = "Mandiant"
	strings:
	$code_fat_corruption = { 8B ?? 56 8B ?? 52 [1-64] 0F B? ?? 16 [1-32] 8B ?? 24 [0-32] 0F B? ?? 0D [1-32] 0F B? ?? 10 [1-32] 0F B? ?? 0E }
	$code_ntfs_corruption = { 0F B? ?? 0B 0F B? ?? 0D [1-64] FF ?? 34 FF ?? 30 [1-64] 0F B? ?? 0B [1-64] FF ?? 3C FF ?? 38 }
	$s1 = "\\\\.\\PhysicalDrive%u" fullword wide
	$s2 = "\\\\.\\EPMNTDRV\\%u" fullword wide
	$s3 = "DRV_X64" fullword wide
	$s4 = "DRV_X86" fullword wide
	$s5 = "DRV_XP_X64" fullword wide
	$s6 = "DRV_XP_X86" fullword wide
	$s7 = "$ATTRIBUTE_LIST" fullword wide
	$s8 = "$EA_INFORMATION" fullword wide
	$s9 = "$SECURITY_DESCRIPTOR" fullword wide
	$s10 = "$INDEX_ROOT" fullword wide
	$s11 = "$INDEX_ALLOCATION" fullword wide
	$s12 = "$LOGGED_UTILITY_STREAM" fullword wide
	condition:
	(uint16(0) == 0x5A4D) and (uint32(uint32(0x3C)) == 0x00004550) and (8 of ($s*) or all of ($code*))
	}

NEARMISS	rule M_Hunting_Win32_NEARMISS_1
	{
	meta:
	author = "Mandiant"
	description = "Rule looks for code present in NEARMISS samples. Based on a rule generated by symhunt for symfunc/cef8160083d485a3676d55b3fc5e1c42."
	strings:
	$c = { 55 8B EC 81 EC AC 08 ?? ?? 53 56 57 33 DB 89 4D E0 68 ?? ?? ?? ?? 8D 85 78 FC FF FF C7 45 DC ?? ?? ?? ?? 53 50 C7 45 E4 ?? ?? ?? ?? 89 5D F8 89 5D A4 E8 ?? ?? ?? ?? 83 C4 0C 68 ?? ?? ?? ?? FF ?? ?? ?? ?? ?? 68 ?? ?? ?? ?? 8B F8 8D 85 78 FC FF FF 68 ?? ?? ?? ?? 50 FF ?? ?? ?? ?? ?? 83 C4 0C 89 45 F0 85 FF 74 ?? 8B ?? ?? ?? ?? ?? 68 ?? ?? ?? ?? 57 FF D6 68 ?? ?? ?? ?? 57 8B D8 FF D6 68 ?? ?? ?? ?? 57 FF D6 8B F0 85 F6 74 ?? 8D 45 F8 50 FF }
	condition:
	filesize < 15MB and uint16(0) == 0x5a4d and uint32(uint32(0x3C)) == 0x00004550 and any of them
	}

NEARMISS	rule M_Hunting_Win32_NEARMISS_2
	{
	meta:
	author = "Mandiant"
	description = "Rule looks for a specific stackstring - mangled SeShutdownPrivilege - found in NEARMISS samples."
	strings:
	$s1 = { 53 00 65 00 [4] 53 00 68 00 [4] 75 00 74 00 [4] 64 00 6F 00 [4] 9A 02 00 00 [4] 00 00 00 00 }
	$s2 = { 77 00 6E 00 [7] 50 00 72 00 }
	condition:
	filesize < 15MB and uint16(0) == 0x5a4d and uint32(uint32(0x3C)) == 0x00004550 and all of them
	}

NEARMISS	rule M_Hunting_Win_WiperPaths_1
	{
	meta:
	author = "Mandiant"
	description = "Detects notable wiper strings"
	reference = "https://twitter.com/ESETresearch/status/1496581903205511181"
	strings:
	$w1 = "\\\\.\\EPMNTDRV" wide fullword
	$w2 = "\\\\.\\PhysicalDrive" wide fullword
	$w3 = "Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced" wide fullword
	$w4 = "\\\\?\\C:\\Windows\\System32\\winevt\\Logs" wide fullword
	$w5 = "\\\\?\\C:\\Documents and Settings" wide fullword
	$w6 = "<<Obsolete>>" wide fullword
	condition:
	uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and (all of them)
	}

WEEVELY	rule M_Webshell_PHP_WEEVELY_1
	{
	meta:
	author = "Mandiant"
	description = "Weevely3 open source webshell detection from https://artikrh.github.io/posts/weevely-backdoor-analysis -- Webshell source code: https://github.com/epinna/weevely3/tree/master/core"
	strings:
	$php = "<?php" ascii
	$rf1 = "$k" ascii
	$rf2 = "$kh" ascii
	$rf3 = "$kf" ascii
	$rf4 = "$p" ascii
	$rf5 = "$o" ascii
	$rf6 = /\$\w{1,4}=str_replace\('\w{1,}','',/ ascii
	condition:
	$php at 0 and all of ($rf*) and filesize > 500 and filesize < 1000
	}

DARKCRYSTALRAT

```
import "pe"

rule M_Backdoor_DARKCRYSTALRAT_1
{
   meta:
      author = "Mandiant"
      description = "Detection for DARKCRYSTAL RAT's C2 checkin and CSharp compiling code"
   strings:
      $c1 = {72?????????a2251907a2251a72????????a2251b11??28????????72????????28????????28????????a2251c72????????a2251d11??28????????28????????a2251e72????????a2251f??11??28????????72????????28????????28????????a2251f??72????????a2251f??7e????????28????????28????????1628????????28????????28????????a2251f??72????????a2251f??07a228????????}
      $c2 = {0228????????0000000428????????0a0428????????0b0672????????6f????????72????????28????????2d??0672????????6f????????72??????28????????2b??170c0839????????00140d0672????????6f????????72????????28????????13??11??2c??73????????0d2b??73????????0d73????????25176f????????0025166f????????0013??11??6f????????0672????????6f????????178d????????25161f??9d6f????????6f????????000911??178d????????25160672????????6f????????a26f????????13??11??6f????????6f????????13??11??39????????0072????????13??0011??6f????????6f??????13???2b??11??6f????????74????????13??1e8d????????25161l??a2251772????????a2251811??6f????????a2251972????????a2251a11??6f????????13??12??28????????a2251b72????????a2251a11??6f????????13??1e8d????????25161l??a2251772????????a2251811??6f????????a2251972????????a2251a11??6f????????13??12??28????????a2251b72????????a2251c11??6f????????a2251d72????????dc0211??7d????????02177d????????dd????????11??6f????????72????????6f????????13??0011??6f????????6f????????13??1613???2b??11??11??9a13??0011??6f????????72????????28????????13??11??2c??0011??11??146f????????262b??0011??175813??11??11??8e6932??0038????????0672????????6f????????72????????28????????13??11??39????????0028????????72????????1f??28????????72????????28????????13??11??0672????????6f????????28????????000772????????6f????????a5????????13??11??2c??0073????????25176f????????002528????????2d??72????????2b??72????????6f????????0025176f????????002511??6f????????0013??11??28????????6f????????00002b??0073????????25166f????????002528????????2d??72????????2b??72????????6f????????002572????????6f????????002572????????11??72????????28????????6f????????0013??11??28????????6f????????000000011??28????????0000de??260000de??0038????????0672????????6f????????72????????28????????13??11??39????????0028????????72????????1f??28????????72????????28????????13??11??0672????????6f????????28????????0073????????25176f????????002528????????2d??72????????2b??72????????6f????????002572????????6f????????002572????????11??72????????28????????6f????????0013??11??28????????6f????????000011??28????????0000de??260000de??0038????????0672????????6f????????72????????28????????13??11??2c??0073????????2517 6f????????002528????????2d??72????????2b??72????????6f????????002572????????6f????????002572????????0672????????6f????????72????????28????????6f????????0013??11??28????????6f????????000002167d????????00de??13??000211??6f????????7d????????02177d????????00de??2a}
      $c3 = {73????????0d2b??73????????0d73????????25176f????????0025166f????????0013??11??6f????????0672????????6f????????178d????????25161f??9d6f????????6f????????000911??178d????????25160672????????6f????????a26f????????13??11??6f????????6f????????13??11??39????????0072????????13??0011??6f????????6f????????13??2b??11??6f????????74????????13??1e8d????????25161l??a2251772????????a2251811??6f????????a2251972????????a2251a11??6f????????13??12??28????????a2251b72????????a2251c11??6f????????a2251d72????????dc0211??7d????????02177d????????dd????????11??6f????????72????????6f????????13??0011??6f????????6f????????13??1613???2b??11??11??9a13??0011??6f????????72????????28????????13??11??2c??0011??11??146f????????262b??0011??175813??11??11??8e6932??0038????????0672????????6f????????72????????28????????13??11??0672????????6f????????28????????0073????????25176f????????002528????????2d??72????????2b??72????????6f????????002572????????6f????????002572????????11??72????????28????????6f????????0013??11??28????????6f????????0000de??260000de??0038????????0672????????6f????????72????????28????????13??11??2c??0073????????2517 6f????????002528????????2d??72????????2b??72????????6f????????002572????????6f????????002572????????0672????????6f????????72????????28????????6f????????0013??11??28????????6f????????000002167d????????00de??13??000211??6f????????7d????????02177d????????00de??2a}

      /*
         0c245b2700e9417c0e1cbfd0f8d1aa70
         DCRatBuild.Managers.DCRat.CreatorAuthenticationTask.ReflectGetter() : void @06000E8D
      */
      $c4 = {0073????????2572????????72????????6f????????0a0673????????0b73????????25176f????????25166f????????256f????????72????????6f??????26256f????????72????????6f????????26256f????????72????????6f????????26256f????????72????????6f????????26256f????????72????????6f????????26256f????????72????????6f????????26256f????????72????????6f????????26256f????????72????????6f????????260c0708178d????????2516027b????????28????????a26f????????0d096f????????6f????????2c??1f??8d????????25167e????????a2251772????????a225187e????????28????????28????????a2251972????????a2251a7e????????a2251b72????????a2251c72????????7e????????28????????28????????a2251d72????????a2251e72????????28????????a228??????28????????262b??1f??8d????????25167e????????a2251772????????a225187e????????28????????28????????a2251972????????a2251a7e????????a2251b72????????a2251c72????????7e????????28????????28????????a2251d72????????a2251e72????????28????????a228????????28????????26096f????????72????????6f????????13??11??6f????????72????????6f????????13??11??11??146f????????26de??261f??8d????????25167e????????a2251772????????a225187e????????28????????28????????a2251972????????a2251a7e????????a2251b72????????a2251c72????????7e????????28????????28????????a2251d72????????a2251e72????????28????????a228????????26de??2a}
   condition:
      uint16(0) == 0x5a4d
      and pe.data_directories[pe.IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR].virtual_address != 0
      and 1 of ($c*)
}
```

DARKCRYSTALRAT	rule M_Backdoor_Win_DARKCRYSTALRAT_Config_1
	{
	meta:
	author = "Mandiant"
	description = "This rule looks for PE files containing part of DARKCRYSTALRAT configuration string. Configuration JSON is stored as base64 encoded, reversed, gzip compressed and again bas64 encoded string."
	strings:
	$s = { 48 00 34 00 73 00 49 00 41 00 41 00 41 00 41 00 41 00 41 00 41 00 45 00 41 00 46 00 32 00 54 00 58 00 58 00 75 00 69 00 4D 00 42 00 43 00 46 00 66 00 39 00 }
	condition:
	filesize < 15MB and uint16(0) == 0x5a4d and uint32(uint32(0x3C)) == 0x00004550 and $s
	}

SMOKELOADER	rule M_Downloader_SMOKELOADER_1
	{
	meta:
	author = " Mandiant"
	description = "This rule is designed to detect on events related to smokeloader. SMOKELOADER is a downloader that retrieves additional payloads via HTTP. Retrieved payloads are mapped into memory and may include plugins that expand SMOKELOADER's functionality. Capabilities added via plugins include keylogging, credential theft, and DDoS. Retrieved payloads may also include additional malware such as AZORULT, FORMBOOK, REMCOS, URSNIF, SILENTNIGHT, TRICKBOT, and SYSTEMBC."
	strings:
	$part_of_winmain = {81 3D [4] 00 04 00 00 5? 5? 75 ?? 8D 45 ?? E8 [4] 8D 75 ?? E8 [4] 8B 3D [4] 5? 8B 5D ?? 33 F6 FF D7 81 FE [4] 7E ?? 81 FB ?? ?? ?? 78 75 ?? 4? 81 FE ?? 1D 00 00 7C ?? 8B 3D [4] 8B 1D [4] 33 F6 8D A4 24 00 00 00 00 6A 00 FF D7 FF D3 FF 15 [4] 81 FE 47 6D 20 00 7F ?? 46 81 FE A4 F6 04 00 7C ?? 8B 3D [4] 33 F6 5B}
	$part_of_alloc_memeory = {8B [5] 05 4B 13 01 00 50 6A 00 89 [5] A3 [4] FF 15 [4] A3 [4] E8 [4] 33 F6 39 35}
	$part_of_stackstring_virtualprotect = {68 38 2B 42 00 FF 15 [4] B1 74 B2 72 68 80 97 42 00 50 A3 [4] C6 05 [4] 56 C6 05 [4] 69 88 15 [4] C6 05 [4] 50 88 0D [4] C6 05 [4] 00 88 0D [4] C6 05 [4] 63 C6 05 [4] 75 C6 05 [4] 61 C6 05 [4] 6C 88 15 [4] C6 05 [4] 6F 88 0D [4] C6 05 [4] 65 FF 15 [4] A3}
	condition:
	uint16(0) == 0x5a4d and uint32(uint32(0x3C)) == 0x00004550 and 2 of the
	}

SHARPCOFFEE	rule M_Downloader_SHARPCOFFEE_1
	{
	meta:
	author = "Mandiant"
	strings:
	$str1 = "ActiveXObject(\"WScript.Shell\").Run(\"powershell.exe" nocase
	$str2 = "new-object net.webclient;" nocase
	$str3 = ".downloaddata('http" nocase
	$str4 = ".uploaddata('http" nocase
	$str5 = "[System.Net.Dns]" nocase
	condition:
	all of them
	}

SHARPCOFFEE.VBS	rule M_APT_Downloader_SHARPCOFFEEVBS_2
	{
	meta:
	author = "Mandiant"
	description = "Detects SHARPCOFFE.VBS variant, a VBS script used to download and run a secondary payload, and upload the output of the secondary payload during the same script execution."
	strings:
	$vbs = "dim" ascii wide nocase
	$a1 = /\$\w{1,20}\.uploaddata\('http:\/\/.{1,20}\/page\d{1,3}',\$\w{1,10}\);/
	$a2 = /=\$\w{1,20}\.downloaddata\('http:\/\/.{1,50}\/page\d{1,3}\/upgrade\.txt'\);if\(/
	condition:
	filesize < 1MB and $vbs at 0 and any of ($a*)
	}

COLDWELL	rule M_Dropper_COLDWELL_Permission_Arch_Check_1
	{
	meta:
	author = "Mandiant"
	
	strings:
	$ = {C7 45 F? 00 05 50 C7 45 F? }
	$ = {C7 45 F? 00 00 00 00 C7 45 F? }
	$ = {0F 95 C3 6A 04 83 C3 [7] F7 D8 6A 0A}
	
	condition:
	all of them
	}

ROARBAT	rule M_Disrupt_ROARBAT_1
	{
	meta:
	author = "Mandiant"
	
	strings:
	$ = "takeown /a /f \"%%"
	$ = "in (C:\\Users,"
	$ = "a -df %"
	$ = "\" & del %"
	
	condition:
	all of them
	}

WILDDIME	rule M_Hunting_Backdoor_PowerShell_WILDDIME_Strings_1
	{
	meta:
	author = "Mandiant"
	description = "Searching for PowerShell scripts with strings associated with WILDDIME."
	
	strings:
	$s1 = "GetEnviron" nocase ascii wide
	$s2 = "R64Encoder" nocase ascii wide
	$s3 = "R64Decoder" nocase ascii wide
	$s4 = "Send-HttpRequest" nocase ascii wide
	$s5 = "JVBERi0xLjcNCiW1tb" nocase ascii wide
	
	condition:
	filesize < 200KB and
	all of them
	}

SHARPENTRY	rule M_Hunting_Downloader_SHARPENTRY_1
	{
	meta:
	author= "Mandiant"
	description="Detects code fragments connected to the payload decoding and mining routines found within SHARPENTRY."
	
	strings:
	$decode_routine = { 0F B6 ?? ?? 0F B6 ?? ?? 33 C2 88 ?? ?? 0F B6 ?? ?? 83 ?? 4D }
	$payload_mine = { 8B ?? ?? 03 ?? ?? 81 ?? 89 C3 81 C3 }
	
	condition:
	uint16(0) == 0x5A4D
	and $decode_routine
	and $payload_mine
	}

SHARPIVORY	rule M_Hunting_Dropper_SHARPIVORY_Strings_1
	{
	meta:
	author = "Mandiant"
	description = "Searching for executables containing strings references to the SHARPIVORY code family."
	
	strings:
	$s1 = "WriteAllBytes"
	$s2 = "FromBase64String"
	$w1 = "schtasks.exe" wide
	$w2 = "kernel32.dll" wide
	$w3 = "/create /tn" wide
	$w4 = "/sc minute /mo 20 /f" wide
	
	condition:
	filesize < 5MB and
	uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and
	all of them
	}

Mandiant Security Validation Actions

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

VID	Name
A101-165	Application Vulnerability - APT44, CVE-2019-10149, Remote Code Execution, Benign Payload
A101-166	Application Vulnerability - APT44, CVE-2019-10149, Remote Code Execution, Malicious Payload
A102-517	Command and Control - APT44, AXETERROR, Beacon, Variant #1
A107-038	Command and Control - APT44, BLACKENERGY, Beacon, Variant #1
A106-188	Command and Control - APT44, BRUSHPASS, DNS Query, Variant #1
A107-010	Command and Control - APT44, DARKCRYSTALRAT, C2 Communication, Variant #2
A105-312	Command and Control - APT44, DARKCRYSTALRAT, DNS Query, Variant #1
A105-407	Command and Control - APT44, DNS Query, Variant #1
A105-408	Command and Control - APT44, DNS Query, Variant #2
A107-026	Command and Control - APT44, FELIXROOT, Beacon, Variant #1
A106-106	Command and Control - APT44, GOGETTER, DNS Query, Variant #1
A107-024	Command and Control - APT44, PASWEB, Download File
A107-027	Command and Control - APT44, PASWEB, Establish Connection
A107-033	Command and Control - APT44, PASWEB, Execute phpinfo() Command
A107-013	Command and Control - APT44, PASWEB, Execute Version Command
A107-016	Command and Control - APT44, PASWEB, File Search
A107-031	Command and Control - APT44, PASWEB, Upload File
A106-103	Command and Control - APT44, QUICKTOW, C2 Communication, HTTP Post, Variant #1
A106-102	Command and Control - APT44, QUICKTOW, DNS Query, Variant #1
A106-008	Command and Control - APT44, SPAREPART, Beaconing, Variant #1
A107-001	Command and Control - APT44, TRICKSHOW, Beacon, Variant #1
A106-994	Command and Control - APT44, TRICKSHOW, Beacon, Variant #2
A106-996	Command and Control - UNC1908, STRAYKEY, Check-in
A106-998	Command and Control - UNC1908, STRAYKEY, Command Response
A106-999	Command and Control - UNC1908, STRAYKEY, Startup Communication
A104-850	Host CLI - APT44, Add New Local User mysql_db, Linux
A106-193	Host CLI - APT44, BRUSHPASS, Modifying Firewall Rules
A106-439	Host CLI - APT44, CADDYWIPER, Scheduled Task, Variant #1

A106-446	Host CLI - APT44, CADDYWIPER, Scheduled Task, Variant #2
A106-438	Host CLI - APT44, GOGETTER, Systemd Service
A104-623	Host CLI - Mshta/Schtasks Persistence via HTA
A106-993	Host CLI - UNC4209, SWEETJADE, Create Mutex, Variant #1
A103-029	Malicious File Transfer - APT44, ARGUEPATCH, Download, Variant #1
A103-873	Malicious File Transfer - APT44, ARGUEPATCH, Download, Variant #4
A102-519	Malicious File Transfer - APT44, AXETERROR, Download, Variant #1
A102-518	Malicious File Transfer - APT44, AXETERROR, Download, Variant #2
A107-000	Malicious File Transfer - APT44, BACKORDER, Download, Variant #1
A102-582	Malicious File Transfer - APT44, BLACKENERGY, Download, Variant #1
A102-583	Malicious File Transfer - APT44, BLACKENERGY, Download, Variant #2
A102-584	Malicious File Transfer - APT44, BLACKENERGY, Download, Variant #3
A102-585	Malicious File Transfer - APT44, BLACKENERGY, Download, Variant #4
A106-190	Malicious File Transfer - APT44, BRUSHPASS, Download, Variant #1
A106-189	Malicious File Transfer - APT44, BRUSHPASS, Download, Variant #2
A103-030	Malicious File Transfer - APT44, CADDYWIPER, Download, Encrypted Variant #1
A102-784	Malicious File Transfer - APT44, CADDYWIPER, Download, Variant #1
A103-615	Malicious File Transfer - APT44, CADDYWIPER, Download, Variant #5
A106-440	Malicious File Transfer - APT44, CADDYWIPER, Download, Variant #6
A107-008	Malicious File Transfer - APT44, COLIBRI Dropper, Download, Variant #1
A102-576	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #1
A102-796	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #10
A102-797	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #11
A102-798	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #12
A102-788	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #2
A102-789	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #3
A102-790	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #4
A102-791	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #5
A102-792	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #6
A102-793	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #7
A102-794	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #8
A102-795	Malicious File Transfer - APT44, CYCLOPSBLINK, Download, Variant #9
A105-313	Malicious File Transfer - APT44, DARKCRYSTALRAT Downloader, Download, Variant #1
A103-624	Malicious File Transfer - APT44, EARLYBLOOM Downloader, Download, Variant #1
A103-613	Malicious File Transfer - APT44, EARLYBLOOM, Download, Variant #1
A101-388	Malicious File Transfer - APT44, EMPIRE, Download
A101-389	Malicious File Transfer - APT44, EMPYRE, Download
A106-990	Malicious File Transfer - APT44, EXARAMEL Dropper, Download, Variant #1
A107-018	Malicious File Transfer - APT44, FAIRROOT, Download, Variant #1
A107-029	Malicious File Transfer - APT44, FELIXROOT Dropper, Download, Variant #1
A107-043	Malicious File Transfer - APT44, FREETOW, Download, Variant #1
A103-166	Malicious File Transfer - APT44, GOGETTER, Download, Variant #1
A103-167	Malicious File Transfer - APT44, GOGETTER, Download, Variant #2
A103-168	Malicious File Transfer - APT44, GOGETTER, Download, Variant #3
A103-169	Malicious File Transfer - APT44, GOGETTER, Download, Variant #4
A106-442	Malicious File Transfer - APT44, GOGETTER, Download, Variant #5
A107-009	Malicious File Transfer - APT44, ILLICITORDER, DARKCRYSTALRAT, Download, Variant #1

A102-993	Malicious File Transfer - APT44, INDUSTROYER, Download, Variant #2
A107-012	Malicious File Transfer - APT44, ITCHYSPARK.SMB, Download, Variant #1
A107-011	Malicious File Transfer - APT44, ITCHYSPARK.WMI, Download, Variant #1
A101-887	Malicious File Transfer - APT44, Malicious Bash Script , Download, Variant #2
A101-390	Malicious File Transfer - APT44, METERPRETER, Download
A102-573	Malicious File Transfer - APT44, NEARMISS, Download, Variant #1
A102-574	Malicious File Transfer - APT44, NEARMISS, Download, Variant #2
A102-662	Malicious File Transfer - APT44, NEARTWIST, Download, Variant #1
A106-995	Malicious File Transfer - APT44, NEWRETURN Dropper, Download, Variant #1
A105-426	Malicious File Transfer - APT44, OWA Credential Harvesting Page, Download, Variant #1
A105-427	Malicious File Transfer - APT44, OWA Credential Harvesting Page, Download, Variant #2
A105-428	Malicious File Transfer - APT44, OWA Credential Harvesting Page, Download, Variant #3
A105-429	Malicious File Transfer - APT44, OWA Credential Harvesting Page, Download, Variant #4
A102-663	Malicious File Transfer - APT44, PARTYTICKET, Download, Variant #1
A103-614	Malicious File Transfer - APT44, POWERDISCO, Download, Variant #1
A107-023	Malicious File Transfer - APT44, REGEORG.NEO, Download, Variant #1
A107-019	Malicious File Transfer - APT44, REGEORG.NEO, Upload, Variant #1
A106-009	Malicious File Transfer - APT44, SPAREPART, Download, Variant #1
A106-546	Malicious File Transfer - APT44, STOWAWAY, Download, Variant #1
A106-547	Malicious File Transfer - APT44, STOWAWAY, Download, Variant #2
A107-030	Malicious File Transfer - APT44, TANKTRAP, Download, Variant #1
A102-579	Malicious File Transfer - APT44, VPNFILTER, Download, Variant #1
A102-580	Malicious File Transfer - APT44, VPNFILTER, Download, Variant #2
A102-581	Malicious File Transfer - APT44, VPNFILTER, Download, Variant #3
A101-287	Malicious File Transfer - CVE-2019-10149, APT44, Malicious Bash Script, Download
A106-997	Malicious File Transfer - UNC4209, SWEETJADE, Download, Variant #1
A106-988	Malicious File Transfer - WSO, Upload, Variant #1
A107-035	Phishing Email - APT44, Malicious Attachment, FAIRROOT, Variant #1
A107-034	Phishing Email - APT44, Malicious Attachment, FELIXROOT, Variant #1
A107-039	Phishing Email - APT44, Malicious Attachment, HEXCHAMBER, EMPIRE, Variant #1
A107-022	Phishing Email - APT44, Malicious Attachment, HEXCHAMBER, Variant #1
A107-032	Phishing Email - APT44, Malicious Attachment, HEXCHAMBER, Variant #2
A107-044	Phishing Email - APT44, Malicious Attachment, PENNYBAG, BLACKENERGY, Variant #1
A107-014	Phishing Email - APT44, Malicious Attachment, PENNYBAG, BLACKENERGY, Variant #2
A107-042	Phishing Email - APT44, Malicious Attachment, PENNYBAG, BLACKENERGY, Variant #3
A107-036	Phishing Email - APT44, Malicious Attachment, PENNYBAG, BLACKENERGY, Variant #4
A101-158	Phishing Email - Malicious Attachment, APT44, Doc Lure
A103-626	Phishing Email - Malicious Attachment, APT44, EARLYBLOOM, HTML Downloader
A107-041	Phishing Email - Malicious Attachment, APT44, TRICKSHOW Dropper, Variant #1
A107-007	Protected Theater - APT44, ILLICITORDER, Execution, Variant #1
A107-020	Protected Theater - APT44, BLACKENERGY, Execution, Variant #1
A105-030	Protected Theater - APT44, CADDYWIPER, Execution, Variant #1
A106-437	Protected Theater - APT44, CADDYWIPER, Execution, Variant #2
A107-002	Protected Theater - APT44, COLIBRI Dropper, Mount ISO, Variant #1
A106-989	Protected Theater - APT44, COLIBRI, Execution, Variant #1
A107-004	Protected Theater - APT44, COLIBRI, Execution, Variant #2
A106-107	Protected Theater - APT44, Create GOGETTER Scheduled Task
A107-005	Protected Theater - APT44, DARKCRYSTALRAT, Execution, Variant #1
A105-346	Protected Theater - APT44, EARLYBLOOM, CVE-2022-30190, HTML Downloader, Execution

A105-143	Protected Theater - APT44, EARLYBLOOM, Execution, Variant #1
A105-349	Protected Theater - APT44, EARLYBLOOM, Malicious Document, Execution, Variant #1
A105-350	Protected Theater - APT44, EARLYBLOOM, Persistence with Scheduled Task
A107-028	Protected Theater - APT44, EMPIRE Stager, Execution, Variant #1
A107-021	Protected Theater - APT44, EMPIRE, Execution, Variant #1
A107-003	Protected Theater - APT44, EXARAMEL Dropper, Deliver EXARAMEL, Variant #1
A107-040	Protected Theater - APT44, FAIRROOT, Execution, Variant #1
A107-015	Protected Theater - APT44, FELIXROOT Downloader, Execution, Variant #1
A107-025	Protected Theater - APT44, FREETOW, Execution, Variant #1
A106-104	Protected Theater - APT44, GOGETTER, Execution, Variant #1
A107-017	Protected Theater - APT44, HEXCHAMBER, Execution, Variant #1
A107-037	Protected Theater - APT44, Install SWEETTREAT as a Service, Variant #1
A104-979	Protected Theater - APT44, NEARMISS, Execution, Variant #1
A106-987	Protected Theater - APT44, NEWRETURN Dropper, Execution, Variant #1
A105-015	Protected Theater - APT44, PARTYTICKET, Execution, Variant #1
A106-105	Protected Theater - APT44, QUICKTOW, Execution, Variant #1
A106-010	Protected Theater - APT44, SPAREPART, Establish Persistence, Variant #1
A107-006	Protected Theater - APT44, WARZONE, Execution via COLIBRI, Variant #1
A106-992	Protected Theater - APT44, WARZONE, Execution via COLIBRI, Variant #2
A106-991	Protected Theater - UNC4209, SWEETJADE, Execution, Variant #1
A106-371	Web Server Activity - APT44, BRUSHPASS, Webshell Command Activity
A106-372	Web Server Activity - APT44, BRUSHPASS, Webshell File Upload Activity