Cyberwarfare: A deep dive into the latest Gamaredon Espionage Campaign

blog.yoroi.company/research/cyberwarfare-a-deep-dive-into-the-latest-gamaredon-espionage-campaign/

February 17, 2020

02/17/2020

Introduction

Gamaredon Group is a Cyber Espionage persistent operation attributed to Russians FSB (*Federal Security Service*) in a long-term military and geo-political confrontation against the Ukrainian government and more in general against the Ukrainian military power.

Gamaredon has been active since 2014, and during this time, the modus operandi has remained almost the same. The most used malware implant is dubbed Pteranodon or Pterodo and consists of a multistage backdoor designed to collect sensitive information or maintaining access on compromised machines. It is distributed in a spear phishing campaign with a weaponized office document that appears to be designed to lure military personnel.

In the recent months, Ukrainian CERT (*CERT-UA*) reported an intensification of Gamaredon Cyberattacks against military targets. The new wave dates back to the end of November 2019 and was first analyzed by Vitali Kremez. Starting from those findings, Cybaze-Yoroi ZLab team decided to deep dive into a technical analysis of the latest Pterodo implant.

Technical Analysis

The complex infection chain begins with a weaponized Office document named “f.doc”. In the following table the initial malware information is provided.

Hash	76ea98e1861c1264b340cf3748c3ec74473b04d042cd6bfda9ce51d086cb5a1a
Threat	Gamaredon Pteranodon weaponized document
Brief Description	Doc file weaponized with Exploit
Ssdeep	768:u0foGtYZKQ5QZJQ6hKVsEEIHNDxpy3TI3dU4DKfLX9Eir:uG1aKQ5OwCrItq3TgGfLt9r

Table 1. Information about initial dropper

The decoy document is written using the ukrainian language mixed to many special chars aimed to lure the target to click on it, and, once opened, it appears as in the following figure.

Figure 1. Overview of the document
The document leverages the common exploit aka template injection and tries to download a second stage from “hxxp://win-apu.]ddns.]net/apu.]dot”.

Figure 2. URL used by document to download the second stage

Thanks to this exploit (Remote Code Execution exploit) the user interaction is not required, in fact the “*enable macro*” button is not shown. The downloaded document has a “.dot” extension, used by Microsoft Office to save templates for different documents with similar formats. Basic Information on the “.dot” file are provided:

Hash	e2cb06e0a5c14b4c5f58d0e56a1dc10b6a1007cf56c77ae6cb07946c3dfe82d8
Threat	Gamaredon Pteranodon loader dot file
Brief Description	Dot file enabling the infection of the Gamaredon Pteranodon
Ssdeep	768:5KCB8tnh7oferuHpC0xw+hnF4J7EyKfJ:oI8XoWruHpp/P4

Table 2. Information about second stage

If we decide to open the document, we see that the document is empty, but it requires the enabling of the macro.

Figure 3. Overview of the second stage document
The body of the macro can be logically divided into two distinct parts:

- The first one is the setting of the registry key "*HKEY_CURRENT_USER\Software\Microsoft\Office\*" & *Application.Version & _"\Word\Security*\" and the declaration of some other variables, such as the dropurl “get-icons.]ddns.net”;
- The second one is the setting of the persistence mechanism through the writing of the vbs code in the Startup folder with name “templates.vbs”. This vbs is properly the macro executed by the macro engine of word

Figure 4. Code of the “template.vbs” stored in the Startup folder
The evidence of the written file in the Startup folder:

Figure 5. Evidence of the “template.vbs” file in the Startup folder
Analyzing the content of “*templates.vbs*” it is possible to notice that it define a variable containing a URL like “*hxxp://get-icons.]ddns.]net/ADMIN-PC_E42CAF54//autoindex.]php*” obtained from "hxp://get-icons.]ddns.]net/" & NlnQCJG & "_" & uRDEJCn & "//autoindex.]php", where “NlnQCJG” is the name that identifies the computer on the network and “uRDEJCn” is the serial number of drive in hexadecimal encoding. From this URL it tries to download another stage then storing it into “C:\Users\admin\AppData\Roaming\” path with random name. At the end, “templates.vbs” script will force the machine to reboot.

Figure 6. Function used to force machine reboot
The dropped sample is an SFX archive, like the tradition of Gamaredon implants.

Hash	c1524a4573bc6acbe59e559c2596975c657ae6bbc0b64f943fffca663b98a95f
Threat	Gamaredon Pteranodon implant SFX archive
Brief Description	SFX Archive First Stage
Ssdeep	24576:zXwOrRsTQlIIIIwIEuCRqKlF8kmh/ZGg4kAL/WUKN7UMOtcv:zgwR/lIIIIwI6RqoukmhxGgZ+WUKZUMv

Table 3. Information about first SFX archive

By simply opening the SFX archive, it is possible to notice two different files that are shown below and named respectively “8957.cmd” and “28847”.

Figure 7. Content of the Gamaredon Pteranodon SFX archive
When executed, the SFX archive will be extracted and the “8957.cmd” will be run. The batch script looks like the following screen:

Figure 8. Bat script source code (with junk instructions)
It contains several junk instructions with the attemption to make the analysis harder. Cleaning the script we obtain:

Figure 9. Batch script source code (cleaned)
At this point, the batch script renames the “28847” file in “28847.exe”, opens it using “pfljk,fkbcerbgblfhs” as password and the file contained inside the “28847.exe” file will be renamed in “WuaucltIC.exe”. Finally, it will be run using “-post.php” as argument.

The fact that the “28847.exe” file can be opened makes us understand that the “28847” file is another SFX file. Some static information about SFX are:

Hash	3dfadf9f23b4c5d17a0c5f5e89715d239c832dbe78551da67815e41e2000fdf1
Threat	Gamaredon Pteranodon implant SFX archive
Brief Description	SFX Archive Second Stage
Ssdeep	24576:vmoO8itbaZiW+qJnmCcpv5lKbbJAiUqKXM:OoZwxVvfoaPu

Table 4. Information about the second SFX archive

Exploring it, it is possible to see several files inside of it, as well as the 6323 file. The following figure shows a complete list.

Figure 10. Content of the second SFX archive
In this case, the SFX archive contains 8 files: five of them are legit DLLs used by the “6323” executable to interoperate with the OLE format defined and used by Microsoft Office. The “ExcelMyMacros.txt” and “wordMacros.txt” files contain further macro script, described next. So, static analysis on the “6323” file shown as its nature: it is written using Microsoft Visual Studio .NET, therefore easily to reverse. Before reversing the executable, it is possible to clean it allowing the size reduction and the junk instruction reduction inside the code. The below image shows the information about the sample before and after the cleaning.

Figure 11. Static information about .NET sample before and after the cleaning
The source code looks as follows.

Figure 12. Part of .NET sample source code
The first check performed is on the arguments: if the arguments length is equal to zero, the malware terminates the execution. After that, the malware checks if the existence of the files “ExcelMyMacros.txt” and “wordMacros.txt” in the same path where it is executed: if true then it reads their contents otherwise it will exit.

Figure 13. Function used by .NET sample to check the presence of the “WordMacros.txt” and the “ExcelMyMacros.txt” files”
Part of the content of the variable “xVGIMEP”:

Figure 14.Piece of the “WordMacros.txt” code
There is a thin difference between the two files.

Figure 15. Difference between “WordMacros.txt” and “ExcelMyMacros.txt” files”
As visible in the previous figure, the only difference between the files are in the variable, registry key and path used by Word rather than by Excel. Finally the macros are executed using the Office engine like in the following figure.

Figure 16. Winword with malicious macro
So let’s start to dissect the macros. For a better comprehension we will be considering only one macro and in the specific case we will analyze “wordMacros.txt” ones. First of all the macro will set the registry key "HKEY_CURRENT_USER\Software\Microsoft\Office\" & Application.Version & _"\Word\Security\" and then will set up two scheduled tasks that will start respectively every 12 and 15 minutes: the first one will run a “IndexOffice.vbs” in the path “%APPDATA%\Microsoft\Office\” and the second one will run “IndexOffice.exe” in the same path.

Figure 17. Registry keys and Scheduled tasks set by malware
Finally, the malware will write the “IndexOffice.txt” file in the “%APPDATA%\Microsoft\Office\” path. The following figure shows what has been previously described:

Figure 18. Part of “IndexOffice.txt” file
The script will check the presence of the “IndexOffice.exe” artifact: if true then it will delete it and it will download a new file/script from “hxxp://masseffect.]space/<PC_Name>_<Hex_Drive_SN>/post.]php”.

Figure 19. Domain “masseffect.]space” declaration and use of the Encode function
The malware tries to save the C2 response and encoding it using Encode function. This function accepts three parameters: the input file, the output file and the arrKey; arrKey is calculated thanks to GetKey function that accepts as input the Hexadecimal value of the Driver SN installed on the machine and returns the key as results. Part of Encode function and complete code of GetKey function are shown below.

Figure 20. Encode function
Figure 21. Function GetKey
Visiting the web page relative to C2, it shows a “Forbidden message” so this means that the domain is still active but refuses incoming requests.

Figure 22. Browser view of the URL “masseffect.]space”

Conclusion

Gamaredon cyberwarfare operations against Ukraine are still active. This technical analysis reveals that the modus operandi of the Group has remained almost identical over the years.

The massive use of weaponized Office documents, Office template injection, sfx archives, wmi and some VBA macro stages that dinamically changes, make the Pterodon attack chain very malleable and adaptive. However, the introduction of a .Net component is a novelty compared to previous Pterodon samples.

Indicator of Compromise

Hashes

- 76ea98e1861c1264b340cf3748c3ec74473b04d042cd6bfda9ce51d086cb5a1a
- e2cb06e0a5c14b4c5f58d0e56a1dc10b6a1007cf56c77ae6cb07946c3dfe82d8
- def13f94cdf793df3e9b42b168550a09ee906f07f61a3f5c9d25ceca44e8068c
- c1524a4573bc6acbe59e559c2596975c657ae6bbc0b64f943fffca663b98a95f
- 86977a785f361d4f26eb3e189293c0e30871de3c93b19653c26a31dd4ed068cc
- 3dfadf9f23b4c5d17a0c5f5e89715d239c832dbe78551da67815e41e2000fdf1
- 2f310c5b16620d9f6e5d93db52607f21040b4829aa6110e22ac55fab659e9fa1
- 145a61a14ec6d32b105a6279cd943317b41f1d27f21ac64df61bcdd464868edd
- ad61df516fb038e806d13d9cc968abaf55eae3b52780d20976ed4e0db440d87b
- f66e820de46bc0d2053c7d24169deb9424f5fdc6973935b108030b03184fcba5
- 40cd2384824ae960a85fc540a763c342c4dc5c9226308d9eb690c98a302fa7a2

Persistence

> %AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\templates.vbs

URL

- hxxp://win-apu.]ddns.]net/apu.]dot/
- hxxp://get-icons.]ddns.]net/apu.]dot/

C2

> hxxp://masseffect.]space/

Yara Rule

```
rule Gamaredon_Campaign_Genuary_2020_Initial_Dropper {
        meta:
        description = "Yara Rule for Gamaredon_f_doc"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2020-02-14"
        tlp = "white"
        category = "informational"

        strings:
         $a1 = { 4B 03 }
         $a2 = { 8E DA 30 14 DD 57 EA 3F }
         $a3 = { 3B 93 46 0F AF B0 2B 33 }
         $a4 = { 50 4B 03 04 14 00 06 00 08 }

    condition:
         all of them
}
rule Gamaredon_Campaign_Genuary_2020_Second_Stage {
        meta:
        description = "Yara Rule for Gamaredon_apu_dot"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2020-02-14"
        tlp = "white"
        category = "informational"

        strings:
         $a1 = "Menu\\Programs\\Startup\\\""
         $a2 = "RandStrinh"
         $a3 = ".txt"
         $a4 = "templates.vbs"
         $a5 = "GET"
         $a6 = "Encode = 1032"
         $a7 = "WShell=CreateObject(\"WScript.Shell\")"
         $a8 = "Security"
         $a9 = "AtEndOfStream"
         $a10 = "GenRandom"
         $a11 = "SaveToFile"
         $a12 = "Sleep"
         $a13 = "WinMgmts:{(Shutdown,RemoteShutdown)}!"
         $a14 = "Scripting"
         $a15 = "//autoindex.php"

    condition:
         11 of ($a*)
}
rule Gamaredon_Campaign_Genuary_2020_SFX_Stage_1 {
        meta:
        description = "Yara Rule for Gamaredon SFX stage 1"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2020-02-14"
        tlp = "white"
        category = "informational"

        strings:
         $a1 = { 4D 5A }
         $a2 = { FF 75 FC E8 F2 22 01 00 }
         $a3 = { FE DE DB DB FE D5 D5 D6 F8 }
         $a4 = { 22 C6 24 A8 BE 81 DE 63 }
         $a5 = { CF 4F D0 C3 C0 91 B0 0D }

    condition:
         all of them
}
rule Gamaredon_Campaign_Genuary_2020_SFX_Stage_2 {
        meta:
        description = "Yara Rule for Gamaredon SFX stage 2"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2020-02-14"
        tlp = "white"
        category = "informational"

        strings:
         $a1 = { 4D 5A }
         $a2 = { 00 E9 07 D4 FD FF 8B 4D F0 81 }
         $a3 = { B7 AB FE B2 B1 B5 FA 9B 11 80 }
         $a4 = { 81 21 25 E0 38 03 FA F0 AF 11 }
         $a5 = { 0A 39 DF F7 40 8D 7B 44 52 }

    condition:
         all of them
}
rule Gamaredon_Campaign_Genuary_2020_dot_NET_stage {
        meta:
        description = "Yara Rule for Gamaredon dot NET stage"
        author = "Cybaze Zlab_Yoroi"
        last_updated = "2020-02-14"
        tlp = "white"
        category = "informational"

        strings:
         $a1 = { 4D 5A }
         $a2 = "AssemblyCompanyAttribute"
         $a3 = "GetDrives"
         $a4 = "Aversome"
         $a5 = "TotalMilliseconds"
         $s1 = { 31 01 C6 01 F2 00 29 01 5C 03 76 }
         $s2 = { 79 02 38 03 93 03 B5 03 }
         $s3 = { 00 07 00 00 11 00 00 72 01 }
         $s4 = { CD DF A6 EF 66 0E 44 D7 }

    condition:
         all of ($a*) and 2 of ($s*)
}
```

This blog post was authored by Davide Testa, Luigi Martire and Antonio Pirozzi of Cybaze-Yoroi ZLAB.