New Techniques to Uncover and Attribute Cobalt Gang Commodity Builders and Infrastructure Revealed

unit42.paloaltonetworks.com/unit42-new-techniques-uncover-attribute-cobalt-gang-commodity-builders-infrastructure-revealed/

Unit 42 October 25, 2018

By Unit 42

October 25, 2018 at 6:00 AM

Category: Unit 42

Tags: Cobalt

This post is also available in: 日本語 (Japanese)

Nowadays, it's very easy for an advanced attacker to use commodity tools and malware along with very simple initial delivery methods to keep a low profile and stay away from possible attribution. One of the most common approaches is the use of spear phishing emails employing social engineering or commonly used exploits (such as CVE-2017-0199 or the ThreadKit builder) to trick the employees of organizations of interest. Once the initial infection has occurred is when the attacker becomes more sophisticated, deploying advanced custom pieces of malware, more advanced tools, and/or using living-off-the land tools (such as the use of PowerShell, or tools like CMSTP or Regsvr32).

This approach makes it more difficult for threat hunters and defenders to find those needles in the haystack necessary to identify a campaign and its objectives. However even if an attacker uses commodity builders and tools, there is always a chance to find specific signals or characteristics that help to identify and track an actor's infrastructure. One of the groups well known for following these TTPs is the Cobalt Gang, which is still active even after the arrest of their alleged leader in Spain this year.

During October 2018, Unit 42 has been investigating ongoing Cobalt Gang campaigns, as well as leveraging the latest information publicly reported in research reports, such as the ones described by Talos or Morphisec, to help discover and tie new infrastructure to this attack group.

As a result, we have been able to identify both the use of a common macro builder as well as specific document metadata which have allowed us to track and cluster new activity and infrastructure associated with the Cobalt Gang.

A Recent Effective Example of Delivery

One of the latest examples related to the campaign under analysis was used in attacks just a few days ago. It shows the simplicity of the attack delivery employed by this group.

The attack reinforces the fact that email is still one of the primary attack vectors we continuously observe. This attack begins by targeting employees at several banking entities across the globe using an email with subject "Confirmations on October 16, 2018".

The sample shown in Figure 1 can already be found in popular public online malware repositories.

(SHA256: 5765ecb239833e5a4b2441e3a2daf3513356d45e1d5c311baeb31f4d503703e).

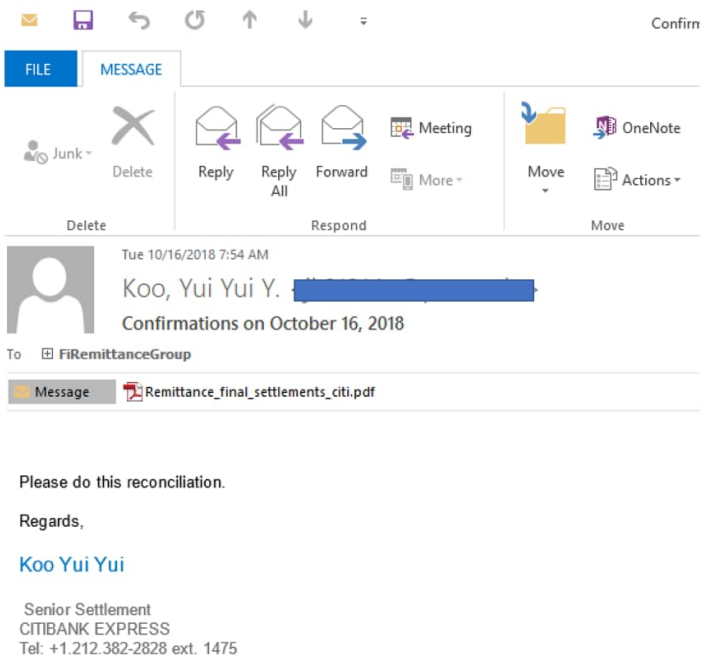

Figure 1. Example of Email delivery

The attachment is just a PDF document without any kind of code or exploit. Instead it seeks to use social engineering to convince the user click a link to download a malicious macro. This is a method used before by the Cobalt Gang and discussed in previous research as for example by Talos.

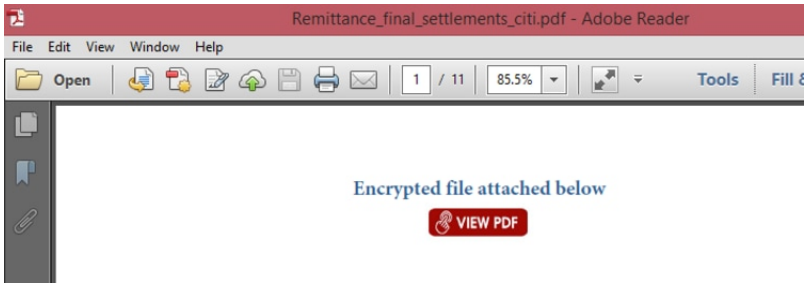

Figure 2. PDF sample with embedded link

The PDF is simple and embeds a link that will open a legitimate Google location, and redirect the browser to a malicious document from there:

Figure 3. Malicious doc browser redirect

In order to be effective against static analysis tools, the PDF that attackers crafted the PDF to seem more authentic: it contains empty pages as well as some text pages that help in not raising red flags during analysis, shown in Figures 4 and 5. Keep in mind that PDFs with low number of pages or high entropy in the content can raise suspicious flags in static analysis.

```
==================================
[+] Analyzing: Remittance_final_settlements_citi.pdf
----------------------------------
[-] Sha256: 57f65ecb239833e5a4b2441e3a2daf3513356d45e1d5c311baeb31f4d503703e
[-] AcroForm.............: 1
[-] Total Entropy........: 7.693554
[-] Entropy inside streams : 7.782326
[-] Entropy outside streams: 5.249718
----------------------------------
[-] Total YARA score.....: 0
[-] Total severity score...: 0
[-] Overall score........: 0
==================================
[-] Scanning didn't determine anything warranting suspicion
```

Figure 4. PDF static analysis

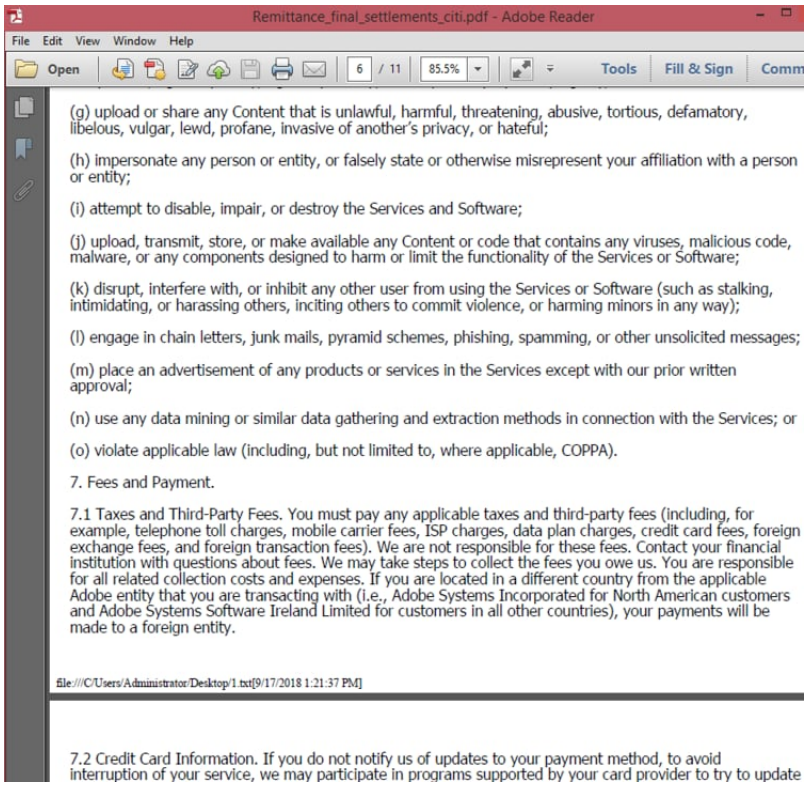

Figure 5. PDF text used to fill pages

By employing these two techniques the PDF avoids almost all traditional AV detection, resulting in a very effective transport of the first stage of the attack via email.

If the attack progresses, the user will be taken to the download of an MS Word document containing malicious macros that has very low detection rate at the moment of this campaign delivery. From a metadata standpoint, the document does not include any specific signal or characteristic that would help us tracking documents from the same author, as shown in Figure 6.

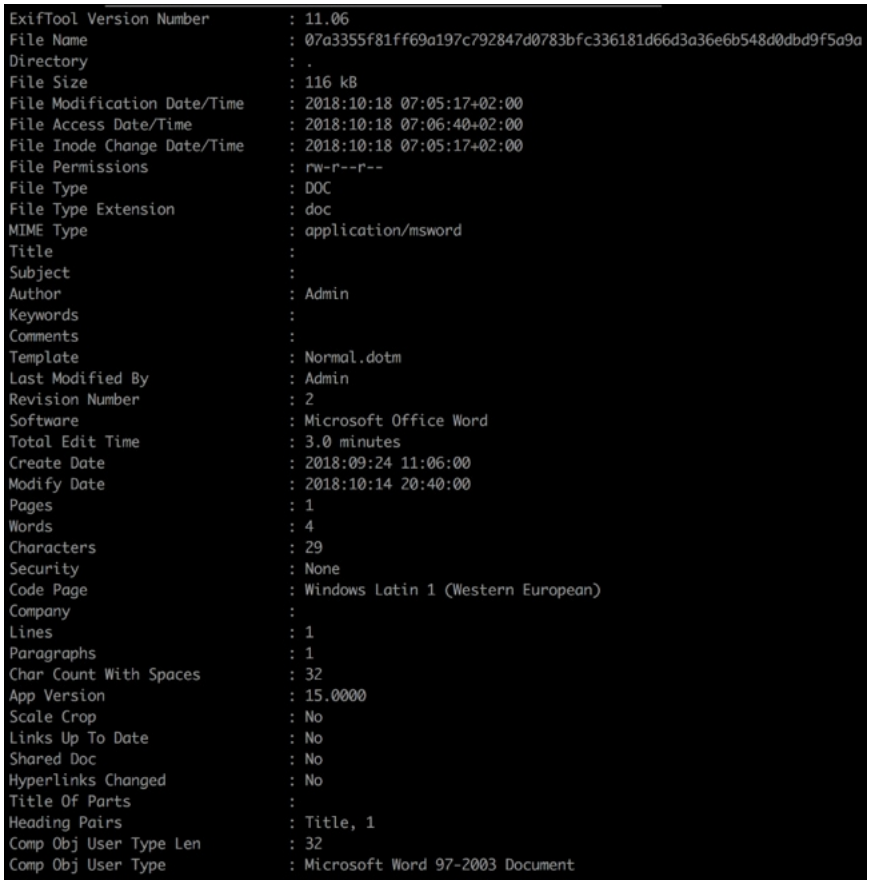

Figure 6. Doc102018.doc metadata

The downloaded malicious macro uses cmstp.exe to run a “scriptlet”, a technique well known to bypass AppLocker, and continues with the next stages of the payload delivery. The objective of this research is not the payload analysis, but to focus on all possible aspects of the attack delivery for further tracking on the actors’ campaign and its associated infrastructure.

So, the question is now… how can this simple delivery method help identify the campaign and objectives?

Macro Builder Identification

The attack also achieves quite low detection results with its macro code, so one of the first focuses of the investigation is the identification of a possible underlying builder. By looking into the macro code for “Doc102018.doc”, we can posit multiple theories.

The macro code is over 1500 lines in length, and starts declaring a set of variables with a very specific nomenclature (in this sample, letXX(num)):

```
1  Dim let52, let1(2) As Byte, let28(9) As Byte, let02(32) As Byte, let39(19) As Byte,
2  let42(13) As Byte, let72(6) As Byte, let81(55) As Byte, let95(1342) As Byte,
3  let23(5) As Byte, let32(59) As Byte, let67(59) As Byte, let01(1 To 255) As Byte
```

Figure 7. Example of format of macro variables

Some of the variables are used in long encoding / decoding routines based on individual character assignations:

```
let95(1055) = let01(89)
let95(281) = let01(106)
let95(283) = let01(123)
let95(217) = let01(110)
let95(442) = let01(193)
let95(1202) = let01(62)
let95(484) = let01(70)
let95(508) = let01(10)
let95(892) = let01(138)
let95(631) = let01(227)
let95(359) = let01(191)
let95(1049) = let01(125)
let95(1092) = let01(159)
let95(214) = let01(31)
let95(267) = let01(86)
let95(840) = let01(127)
```

Figure 8. Using specific variable format in decoding routines

Procedures and functions are also defined using the same nomenclature (in this sample, letXX()):

```
Private Sub let24()
let23(4) = let01(73)
let23(1) = let01(166)
let23(0) = let01(28)
let23(3) = let01(145)
let23(2) = let01(205)
let23(5) = let01(181)
End Sub
Private Function let14(let33() As Byte, let36)
Dim let37, let18
On Error GoTo let11
While let37 <= let36
let18 = let33(let37)
If let18 = 0 Then
Exit Function
```

Figure 9. Procedures and Functions in VBA code

And it makes use of the API call “CallByName” to invoke methods at runtime:

```
Private Function let5(let26, let88, let47, let4, let21, let66)
On Error GoTo let2
Set let5 = CallByName(let26, let88, let47, let21)
let2:
End Function
```

Figure 10. Use of CallByName in VBA code

If we analyze some previous samples linked to Cobalt Gang, such as the ones depicted by Morphisec, this pattern is also observable (in this case, using PkXX instead of letXX):

```
Dim Pk7, Pk22(2) As Byte, Pk36(9) As Byte, Pk529(32) As Byte, Pk5(19) As Byte,
Pk90(13) As Byte, Pk61(5) As Byte, Pk671(55) As Byte, Pk65(805) As Byte, Pk52(5) As Byte
Private Function Pk449(Pk753)
Set Pk449 = GetObject(Pk753)
End Function

Private Sub Pk79()
Pk529(9) = 193
Pk529(15) = 129
Pk529(27) = 142
Pk529(0) = 248
Pk529(16) = 75
Pk529(12) = 73
Pk529(17) = 193
Pk529(5) = 21
Pk529(30) = 126
Pk529(25) = 205
Pk529(28) = 168
Pk529(3) = 146
Pk529(32) = 79
Pk529(18) = 156
Pk529(31) = 107
Pk529(6) = 162
Pk529(20) = 131
Pk529(8) = 252
Pk529(14) = 229
Pk529(23) = 42
Pk529(7) = 136
Pk529(22) = 144
Pk529(26) = 109
```

```
On Error GoTo Pk0
If Pk1 = 1 Then
CallByName Pk2, Pk348, Pk509, Pk173(0), Null, Pk173(1)
Else
Set Pk228 = CallByName(Pk2, Pk348, Pk509, Pk173(0))
```

Figure 11. VBA pattern in other documents

One initial approach to hunt for the pattern can be based on the following regular expressions for the different areas:

Variable definitions **/[A-Za-z]k[0-9]{2}([0-9]{1})/**
Function definitions **/Private Function [A-Za-z]{2,5}[0-9]{2,3}\(/**

Procedure definitions **/Sub [A-Za-z]{2,5}[0-9]{2,5}\(/**

In order to test our hypothesis for the builder we created the following Yara rules:

```
rule cmstp_macro_builder_rev_a
{
    meta:
        description="CMSTP macro builder based on variable names and runtime invoke"
        author="Palo Alto Networks Unit42"
    strings:
        $method="CallByName"
        $varexp=/[A-Za-z]k[0-9]{2}([0-9]{1})/
    condition:
        $method and
        #method == 2 and
        #varexp > 10
}
rule cmstp_macro_builder_rev_b {
    meta:
        description="CMSTP macro builder based on routines and functions names and runtime invoke"
        author="Palo Alto Networks Unit42"
    strings:
        $func=/Private Function [A-Za-z]{1,5}[0-9]{2,3}\(/
        $sub=/Sub [A-Za-z]{1,5}[0-9]{2,5}\(/
        $call="CallByName"
    condition:
        $call and
        #func > 1 and
        #sub > 1
}
```

Hunting with these Yara rules leads to very positive results identifying this builder as well as a set of malicious documents using it. But the documents identified are not always targeting the finance or banking industries, and so, we cannot guarantee that this builder is only used by this specific Cobalt Gang group and its campaigns against those industries.

However, using this in combination with other aspects such as the target, payload, or dropper characteristics, becomes very useful in tracking this group's campaigns, as we will see in the following sections.

Let's focus then on the first stage of the delivery, the PDF documents.

Common Signals in PDF Documents

As we have seen, the use of a commodity PDF file with an embedded Google redirect link results in a very effective social engineering artifact. As there is no exploit or code executed, our research will now be focused on the metadata information from the document for further analysis.

```
File Name                       : Remittance_final_settlements_citi.pdf
Directory                       : .
File Size                       : 39 kB
File Modification Date/Time     : 2018:10:17 22:54:56+02:00
File Access Date/Time           : 2018:10:17 22:55:08+02:00
File Inode Change Date/Time     : 2018:10:17 22:55:07+02:00
File Permissions                : rw-r--r--
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.6
Linearized                      : Yes
Create Date                     : 2018:05:27 17:56:11+02:00
Creator                         : Adobe Acrobat 18.0
Modify Date                     : 2018:10:14 22:42:41+02:00
Has XFA                         : No
XMP Toolkit                     : Adobe XMP Core 5.6-c015 84.159810, 2016/09/10-02:41:30
Metadata Date                   : 2018:10:14 22:42:41+02:00
Creator Tool                    : Adobe Acrobat 18.0
Format                          : application/pdf
Title                           :
Document ID                     : uuid:31ac3688-619c-4fd4-8e3f-e59d0354a338
Instance ID                     : uuid:05a8e817-f2cf-4c93-8652-b3dc746521e8
Producer                        : Acrobat Web Capture 15.0
Page Count                      : 11
```

Figure 12. PDF Exiftool metadata

Our next hypothesis would be to check if the PDF documents could have been created based on a template document, where the author modifies the embedded link in the PDF and saves different document versions over time.

Based on the XMP specification, we will pay attention to the values of the “DocumentID” and “InstanceID” Media Management Properties:

Name	Type	Property content
xmpMM:DerivedFrom	ResourceRef	A reference to the resource from which this one is derived. This should be a minimal reference, in which missing components can be assumed to be unchanged. See definitions of *rendition* (3.7) and *version* (3.9). NOTE A rendition might need to specify only the **xmpMM:InstanceID** and **xmpMM:RenditionClass** of the original.
xmpMM:DocumentID	GUID	The common identifier for all versions and renditions of a resource. See Annex A, "(informative) Document and instance IDs" and definitions of *rendition* (3.7) and *version* (3.9).
xmpMM:InstanceID	GUID	An identifier for a specific incarnation of a resource, updated each time a file is saved. See Annex A, "(informative) Document and instance IDs".

Figure 13. XMP Media Management Properties

In order to confirm this hypothesis, let’s focus on the Document ID metadata field. Basically, saving the same template twice with 2 different links would produce the same Document ID but multiple Instance ID values (one per saved document).

Searching our telemetry data for this metadata content produces interesting results.

In order to help hunting for the content, the following Yara rule also could be used:

```
rule cobaltgang_pdf_metadata_rev_a{
    meta:
        description="Find documents saved from the same potential Cobalt Gang PDF template"
        author="Palo Alto Networks Unit 42"
    strings:
            $ = "<xmpMM:DocumentID>uuid:31ac3688-619c-4fd4-8e3f-e59d0354a338" ascii wide
    condition:
            any of them
}
```

The results confirm our hypothesis (see Appendix for IOCs), and we have been able to find multiple PDF files that the attacker has saved with different contents but starting on the same “template”, all of them sharing the same characteristics.

(different content)

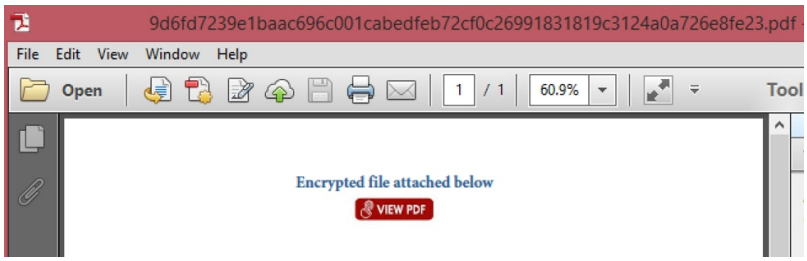

Figure 14. Example PDF document

Further analysis on the subsequent stages of the attack will allow us to confirm if the samples are related to Cobalt Gang campaigns.

For example, let's analyze the following document:

Observed File Name	SHA256
REMITTER REFERENCE PMT.pdf	1d0aae6cff1f7a772fac67b74a39904b8b9da46484b4ae8b621a6566f7761d16

The document was delivered by email, with the subject "Fund Transfer 08-October-2018", targeting banking customers:

```
From Benoît Filion <benoit.filion.2@ulaval.ca>  Wed Oct 10 03:02:13 2018
Date: Mon, 8 Oct 2018 17:35:04 -0400
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="15391333331.40EF41.14329"
Content-Transfer-Encoding: 7bit
Subject: Fund Transfer 08-October-2018
From: =?utf-8?Q?Beno=C3=AEt_Filion?= <benoit.filion.2@ulaval.ca>
To: "Armando Antonio Medrano Duran" <amedrano@bancosol.com.bo>
Message-Id: <15390327126c10f38df42e7920b0ba46b1b9daed80_3012@ulaval.ca>
Received: from SN1NAM02FT024.eop-nam02.prod.protection.outlook.com
 (2a01:111:f400:7e44::209) by BYAPR11CA0051.outlook.office365.com
 (2603:10b6:a03:80::28) with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384) id 15.20.1207.21 via Frontend
 Transport; Tue, 9 Oct 2018 01:08:42 +0000
Received: from ul-exc-pr-as02.ulaval.ca (132.203.244.30) by
```

Figure 15. Email data associated to REMITTER REFERENCE PMT.pdf

And it contains the embedded link redirecting to the following URL:

hxxps://fundswp[.]com/Document082018.doc

Which downloads the document:

Observed File Name	SHA256
Document082018.doc	020ba5a273c0992d62faa05144aed7f174af64c836bf82009ada46f1ce3b6eee

By extracting the macro code, we can validate how it matches the macro builder described in the previous section. The following output shows how running the Yara rule searching for the macro builder against the extracted VBA contents of the document produces the expected match in its contents:

```
❯ yara cmstp_macro_builder.yar 020ba5a273c0992d62faa05144aed7f174af64c836bf82009ada46f1ce3b6eee_subfiles

cmstp_macro_builder_2
020ba5a273c0992d62faa05144aed7f174af64c836bf82009ada46f1ce3b6eee_subfiles/e657fe761effbe7e11e3cc343ba6845c2c9a6c989e7b80

cmstp_macro_builder_2
020ba5a273c0992d62faa05144aed7f174af64c836bf82009ada46f1ce3b6eee_subfiles/8a6d2cccb6f2007cb7fa29d3f009f9fbe305bffc45dc35d3
```

The pieces of our puzzle start to match for this campaign if we now put things in perspective:

1. Hunting for PDF files that are created with the same "DocumentID" management metadata field result in a set of files that have been used in email delivery against banking entities.
2. All of the PDF files embed a link based on a Google redirect, leading to the download of a Microsoft Office document file.
3. The Microsoft Office document files contain macros for code execution. Those macros match the characteristics of the builder that we have characterized.

Discovering the Attacker's Infrastructure

With these results obtained, we can start to move towards finding attacker infrastructure pieces based on multiple aspects, such as the hunting rules defined in previous sections, session data obtained by our telemetry, or public WHOIS registrar data.

Using our "hunting rules"

Based on the metadata and builder characteristics, we have tracked a set of malicious PDF and Office files (see Appendix) that provide us with domains and Office files in use by the attacker.

Some examples of the PDF and embedded C2 links and document names are below in Table 1:

SHA256	Embedded Link
1fd9ba8eb97bf03cd4d3cbaac867595c920f1f36ebfbe9c1fc76558ea5e0ece5	hxxp://www[.]pedidoslalacteo[.]com[.]ar/Proof-of-payment-19.09.2018.doc
5ac1612535b6981259cfac95efe84c5608cf51e3a49b9c1e00c5d374f90d10b2	hxxps://s3[.]sovereigncars[.]org[.]uk/inv005189.pdf
07f60611836c0a679c0fb2e25f5caeb4d29cd970919d47f715666b80be46f45c	hxxps://alotile[.]biz/Document092018.doc
9d6fd7239e1baac696c001cabedfeb72cf0c26991831819c3124a0a726e8fe23	hxxps://goo[.]gl/mn7iGj Which redirects to: hxxps://document[.]cdn-one[.]biz/doc000512.pdf
444c63bb794abe3d2b524e0cb2c8dcc174279b23b1bce949a7125df9fab25c1c	hxxps://safesecurefiles[.]com/doc041791.pdf
a5f2ad08b5afdbd5317b51d0d2dd8f781903522844c786a11a0957a81abfd29e	hxxp://www[.]mky[.]com/Proof-of-payment-19.09.2018.doc
df18e997a2f755159f0753c4e69a45764f746657b782f6d3c878afb8befe2b69	hxxps://mail[.]halcyonih[.]com/uploads/doc004718538.pdf

Table 1. Example PDF and embedded links

The PDF documents and URLs allowed us to discover multiple overlaps between this new infrastructure and the existing knowledge about Cobalt Gang attributed activity in previous research, corroborating new infrastructure belonging to the same attacker.

Let's see a couple of examples of PDF documents from the list which belong to the same Document ID.

Observed File Name	SHA256
inv005189.pdf	5ac1612535b6981259cfac95efe84c5608cf51e3a49b9c1e00c5d374f90d10b2

This sample has been already documented in previous campaigns, being related to s3[.]sovereigncars[.]org[.]uk domain. See the Talos blog.

Observed File Name	SHA256
doc000512.pdf	9d6fd7239e1baac696c001cabedfeb72cf0c26991831819c3124a0a726e8fe23

The sample embeds the URL hxxps://goo[.]gl/mn7iGj which is actually a shortened URL resolving to hxxps://document[.]cdn-one[.]biz/doc000512.pdf.

Domain cdn-one[.]biz is a well-known Cobalt Gang attributed domain in previous analysis.

The complete list of domains used by the PDF identified can be found in the Appendix section.

Pivoting on Email Sender Telemetry

Based on email delivery data, our telemetry helps us collecting samples related and indicators related to the campaign.

Let's put a simple example of how tracking session data lead us to new infrastructure, by using some of the email sender data that is identified in malicious email sessions sending the PDF documents.

For example, the following senders belong to the recent campaign and are spoofing both legitimate email domains and senders:

dominique.denis-berube.1@ulaval.ca
billb@verticalwebmedia.combillb@verticalwebmedia.com
benoit.filion.2@ulaval.cabenoit.filion.2@ulaval.ca
dominique.denis-berube.1@ulaval.ca

Some of the samples delivered by these senders and their embedded links are shown below in Table 2:

SHA256	Embedded link
1c1a6bb0937c454eb397495eea034e00d1f7cf4e77481a04439afbc5b3503396	hxxps://alotile.biz/Document092018.doc
187e0d911cd0393caad1364ded1c394257cd149898b31f9718c7c6319af79818	hxxps://alotile.biz/Document042018.doc
988d430ce0e9f19634cf7955eac6eb03e3b7774b788010c2a9742b38016d1ebf	hxxps://fundsxe.com/Document09202018.doc
852f11e5131d3dab9812fd8ce3cd94c1333904f38713ff959f980a168ef0d4ce	hxxps://fundsxe.com/Document09222018.doc

Table 2. Email sender associated PDFs and embedded links

These sample are delivered under the following file names:

REMITTER REFERENCE PMT.pdf
Aml_S0680260A79301.pdf
CIT180126-000768.pdf
AMENDMENT.pdf
Citi720TEME171440008_Query.pdf
Query _S-170526-005399.pdf

Both the domains and file names correlate with the results of the domains obtained based on hunting for PDFs metadata and macro builder structure, allowing us to keep tracking new activity over time.

WHOIS Registrar Overlaps

Two of the newly discovered domains used by the collected PDF documents have very interesting registrant information, pointing to a public registrant name, “grigoredanbadescu”.

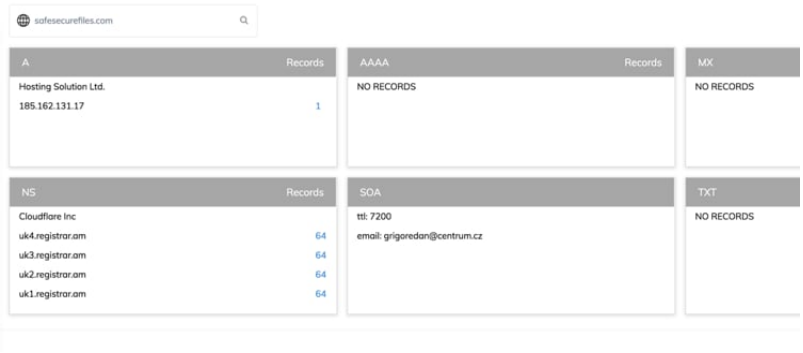

Figure 16. Historical DNS data on safesecurefiles[.]com

Domains:

safesecurefiles[.]com

document[.]cdn-one[.]biz

WHOIS registrar information:

grigoredan@centrum.cz
Grigoredanbadescu

+4001289858474 (Romania)

Brasov

Romania

By pivoting on infrastructure related to the same registrant data we can obtain a very interesting set of domains:

arubrabank[.]com

outlook-368[.]com

usasecurefiles[.]com

safesecurefiles[.]com

ms-server838[.]com

msoffice-365[.]com

total-share[.]biz

bank-net[.]biz

cdn-one[.]biz

total-cloud[.]biz

web-share[.]biz

cloud-direct[.]biz

n-document[.]biz

my-documents[.]biz

firstcloud[.]biz

yourdocument[.]biz

xstorage[.]biz

safe-cloud[.]biz

via24[.]biz

zstorage[.]biz

webclient1[.]biz

bnet1[.]biz

firstcloud[.]biz

mycontent[.]biz

total7[.]biz

freecloud[.]biz

contents[.]bz

judgebin[.]bz

Many of the domains listed are already known as malicious domains attributed to other Cobalt Gang campaigns.

As an important note “arubrabank[.]com” is a new domain registered on 2019-09-18 and still not observed in an active campaign.

The domain seems to be intended to mimic the legitimate Arubabank website for further activity:

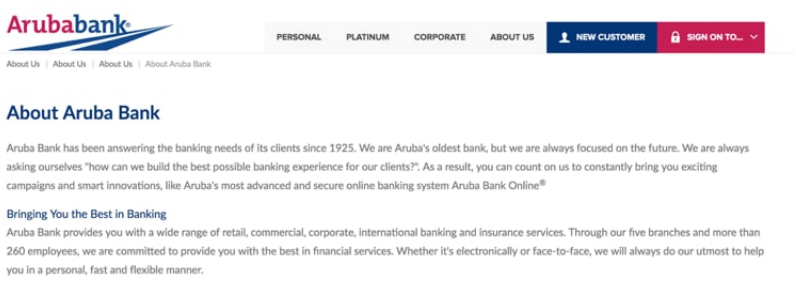

Figure 17. Arubabank legitimate site

Infrastructure Relationships

Let's summarize all the pieces of our puzzle, now that we can put together all the relationships we observed.

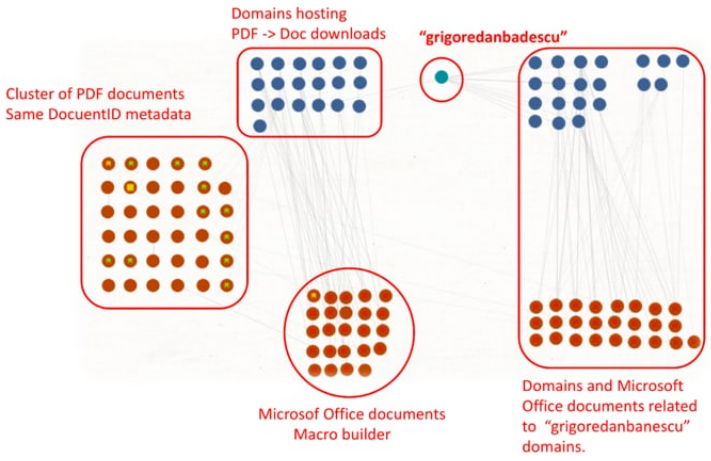

Figure 18. Maltego graph. Summary of relations and overlaps.

As it can be observed in the different clusters of activity:

1. The initial cluster of PDF documents, on the left, links to specific domains for the download of Microsoft Office files using the macro builder structure.
2. Some of the domains in use are publicly registered with the name of "grigoredanbanescu" and allow us to find other related domains, which are already linked to previous Cobalt Gang activity.
3. Some of the initial PDFs have relations with Microsoft Office files linked to "grigoredanbanescu" activity, confirming again the relationships.

Conclusion

Commodity attacks are widely used for both criminal and more targeted attacks, making identification difficult for networks defenders and threat hunters. One actor that makes extensive use of this approach to compromise victims is the Cobalt Gang.

By focusing on specific aspects of the macro builders and metadata the actors left behind we were able to develop new mechanisms to track and hunt Cobalt Gang activity and infrastructure.

Palo Alto Networks customers are protected in the following ways:

- WildFire detects malware samples used by this campaign.
- Traps prevents these attacks at the endpoint.
- All involved malicious domains have been covered by PAN-DB URL Filtering.
- An Autofocus tag has been created for tracking CobaltGang actor group.

Appendix – Indicators of Compromise

Initial example

Type	SHA256
Email	2f74c8b55292d59ab66960f21a4413d4d54f8b7500bb385954e7ffe68d775443
PDF	57f65ecb239833e5a4b2441e3a2daf3513356d45e1d5c311baeb31f4d503703e
Microsoft Office Document	07a3355f81ff69a197c792847d0783bfc336181d66d3a36e6b548d0dbd9f5a9a
Domain	transef[.]biz

Outlook messages

477c432382c97648767ee45c264f0f2aaf8d3d9f9ed547d8418db12b7c140760

e0f1dbc10088b68f772ee73b0785c3d67b8e5f147b687911613d163ad5ebda6d

e6a17617eaa98c49bfb2c9d3d090ffea69bb0c1864c43861bdf8d027339ea847

Microsoft Office Document (Macro builder)

020ba5a273c0992d62faa05144aed7f174af64c836bf82009ada46f1ce3b6eee

8004601c08983420408d2784e2a4aa79de426d41a09726a884edcb21f83ee7f8

d8a2384a51cd59f6390e6a4fcb04b51358cdbd5e04cae5be23daae548c306a73

161ba501b4ea6f7c2c8d224e55e566fef95064e1ed059d8287bc07e790f740e8

62a278119d732e4c839ee074553f087588a9040be027bdf9e617413c6fd2e9af

641d692386dab5ca60f4c6b1da0edecc5c3473c9a7d187dad6098786404780a3

07a3355f81ff69a197c792847d0783bfc336181d66d3a36e6b548d0dbd9f5a9a

161ba501b4ea6f7c2c8d224e55e566fef95064e1ed059d8287bc07e790f740e8

12ecb6b3780cd19ea84f6e84e816a701e8231441bf90145481baa0648139e001

a6f941fcec01fb006fc51df96396aeeb826cdf3864756669e19cb145fe41692f

19dc9b93870ddc3beb7fdeea2980c95edc489040e39381d89d0dfe0a825a1570

cb5644bd670dcd9caf5185ebe396996e514ed1d93982157186611135aea79bd3

a0111977c79f4eb30511f22055b54e4e973c0501240f3ba462691b1b4999d561

PDF Documents

3a7525ffa571775aca45551ebd2c192d9b8ed45db1a61bdd8398d91db885d7a2

1d0aae6cff1f7a772fac67b74a39904b8b9da46484b4ae8b621a6566f7761d16

1c1a6bb0937c454eb397495eea034e00d1f7cf4e77481a04439afbc5b3503396

187e0d911cd0393caad1364ded1c394257cd149898b31f9718c7c6319af79818

988d430ce0e9f19634cf7955eac6eb03e3b7774b788010c2a9742b38016d1ebf

852f11e5131d3dab9812fd8ce3cd94c1333904f38713ff959f980a168ef0d4ce

9d6fd7239e1baac696c001cabedfeb72cf0c26991831819c3124a0a726e8fe23

5ac1612535b6981259cfac95efe84c5608cf51e3a49b9c1e00c5d374f90d10b2

df18e997a2f755159f0753c4e69a45764f746657b782f6d3c878afb8befe2b69

a5f2ad08b5afdbd5317b51d0d2dd8f781903522844c786a11a0957a81abfd29e

66bd5e492531adf675897de5de8aee427b896c9b2c406daff006ce6a4e8aa810

1fd9ba8eb97bf03cd4d3cbaac867595c920f1f36ebfbe9c1fc76558ea5e0ece5

d5328e519daadaf1520619da1f24f6d81d23c84222640058bbb366752be93537

94c9fa812cebb733eda3a4eed33a0a49b60c207bb0f9153c0d08724c8b30f578

07f60611836c0a679c0fb2e25f5caeb4d29cd970919d47f715666b80be46f45c

7b9c183dc40c8d765e98024f8fb6565c69dee2bb97957c5ba754a23d2698bf7a

195580b78e144f66ac1f9be2b927d7828ed1dc3974dc1897e0ed59a96ac8f4e1

444c63bb794abe3d2b524e0cb2c8dcc174279b23b1bce949a7125df9fab25c1c

07f60611836c0a679c0fb2e25f5caeb4d29cd970919d47f715666b80be46f45c

7629dfcc9345578626a250afb67027955c6f78dd80b771c2968c5be0d4b11c59

195580b78e144f66ac1f9be2b927d7828ed1dc3974dc1897e0ed59a96ac8f4e1

b92707ebfaa15225064ff3a1a7d279b3dde1e70200e37d0074e9acc160cb16a7

ebf309ecd6c7a0911e1252d9e90fd302bfbd3e1d2679772025bdb9cc38bca141

57f65ecb239833e5a4b2441e3a2daf3513356d45e1d5c311baeb31f4d503703e

Domains

alotile[.]biz

fundsxe[.]com

s3[.]sovereigncars[.]org[.]uk

safesecurefiles[.]com

document[.]cdn-one[.]biz

mail[.]halcyonih[.]com

transef[.]biz

Domains registered by “grigoredanbanescu”

arubrabank[.]com

outlook-368[.]com

usasecurefiles[.]com

safesecurefiles[.]com

ms-server838[.]com

msoffice-365[.]com

total-share[.]biz

bank-net[.]biz

cdn-one[.]biz

total-cloud[.]biz

web-share[.]biz

cloud-direct[.]biz

n-document[.]biz

my-documents[.]biz

firstcloud[.]biz

yourdocument[.]biz

xstorage[.]biz

safe-cloud[.]biz

via24[.]biz

zstorage[.]biz

webclient1[.]biz

bnet1[.]biz

firstcloud[.]biz

mycontent[.]biz

total7[.]biz

freecloud[.]biz

contents[.]bz

judgebin[.]bz

URLs

hxxp://www[.]pedidoslalacteo[.]com[.]ar/Proof-of-payment-19.09.2018.doc

hxxps://s3[.]sovereigncars[.]org[.]uk/inv005189.pdf

hxxps://alotile[.]biz/Document092018.doc

hxxps://goo[.]gl/mn7iGj

hxxps://document[.]cdn-one[.]biz/doc000512.pdf

hxxps://safesecurefiles[.]com/doc041791.pdf

hxxp://www[.]mky[.]com/Proof-of-payment-19.09.2018.doc

hxxps://mail[.]halcyonih[.]com/uploads/doc004718538.pdf

hxxps://e-dropbox[.]biz/doc058915654e.pdf

hxxp://www[.]bit[.]do/etaYk

hxxps://cloud-direct[.]biz/doc0047581678.pdf

hxxps://transef[.]biz/Doc102018.doc

Observed File Names

Document082018.doc

REMITTER REFERENCE PMT.pdf

Aml_S0680260A79301.pdf

CIT180126-000768.pdf

AMENDMENT.pdf

Citi720TEME171440008_Query.pdf

Query _S-170526-005399.pdf

Document092018.doc

Proof of payment 19.09.2018.doc

Document092018.doc

doc005681.doc