Decoy Microsoft Word document delivers malware through a RAT

By Jerome Segura

Published: 2017-10-13 · Archived: 2026-04-05 20:32:12 UTC

October 13, 2017

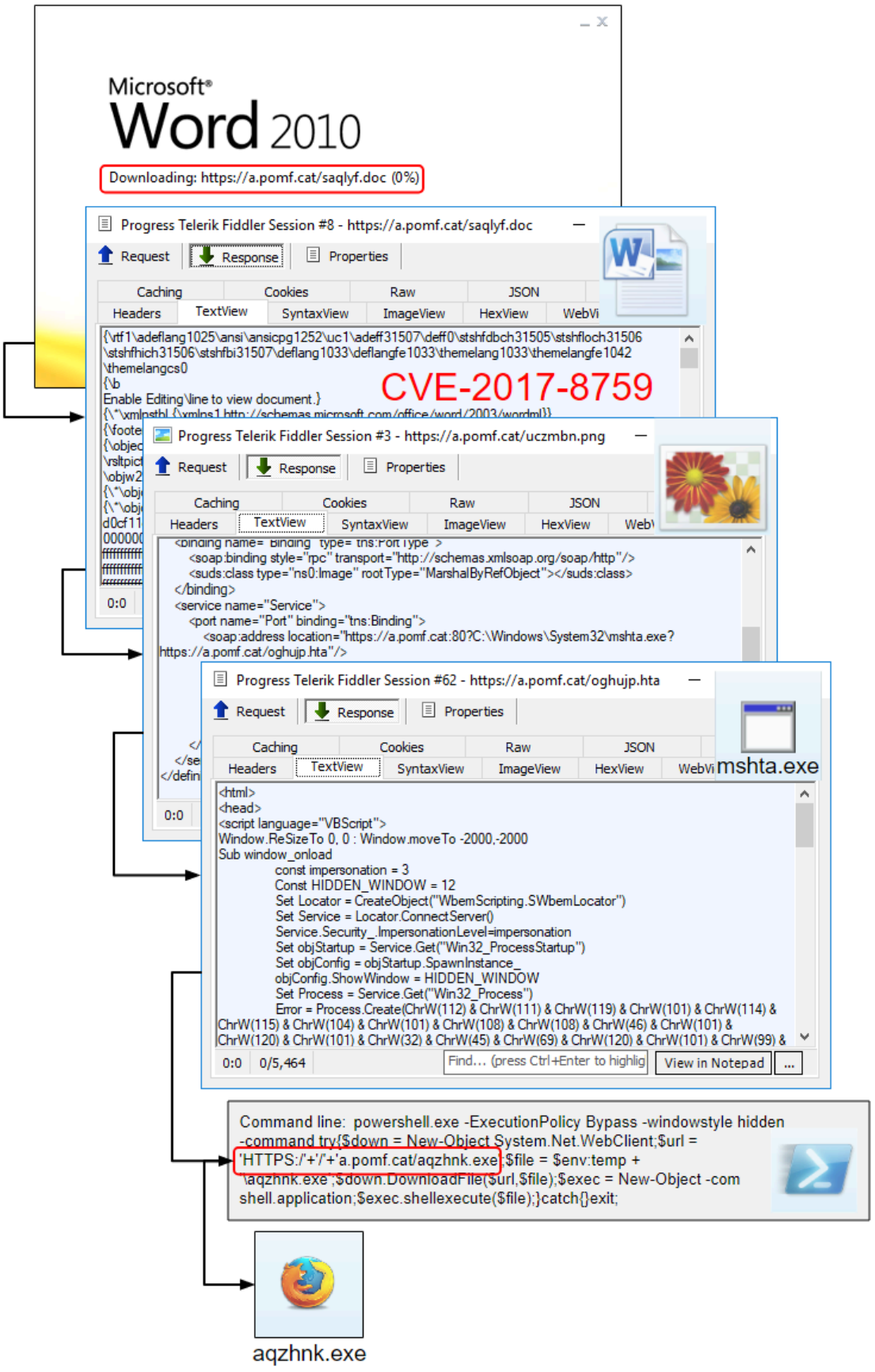
In this post, we take a look at a Microsoft Word document which itself is somewhat clean, but is used to launch a multi-stage attack that relies on the hyperlink feature in the OpenXML format. This then loads another document that contains an exploit.

Most malicious Microsoft Office documents involve either macros, embedded scripts, or exploits and are typically delivered via email. In this case, the unsuspecting user opening the decoy Word document will trigger an automatic (no click or interaction required) download of a malicious RTF file that deploys an exploit (CVE-2017-8759), which ends up distributing the final malware payload.

The several-step removed payload is a commercial Remote Administration Tool that, in this case, is used for nefarious purposes. Victims will be none-the-wiser as the infection process happens in the background, while their Word document finally loads what looks like legitimate content.

While attackers could have sent the exploit-laced document first, that might have triggered detection and quarantine at the email gateway. Instead, the benign document acted as a kind of Trojan horse that made its way to the end user’s desktop, where it would finally show its real intent.

The diagram below summarizes the different steps that this attack takes, from the original document all the way to the malware payload.



Initial package

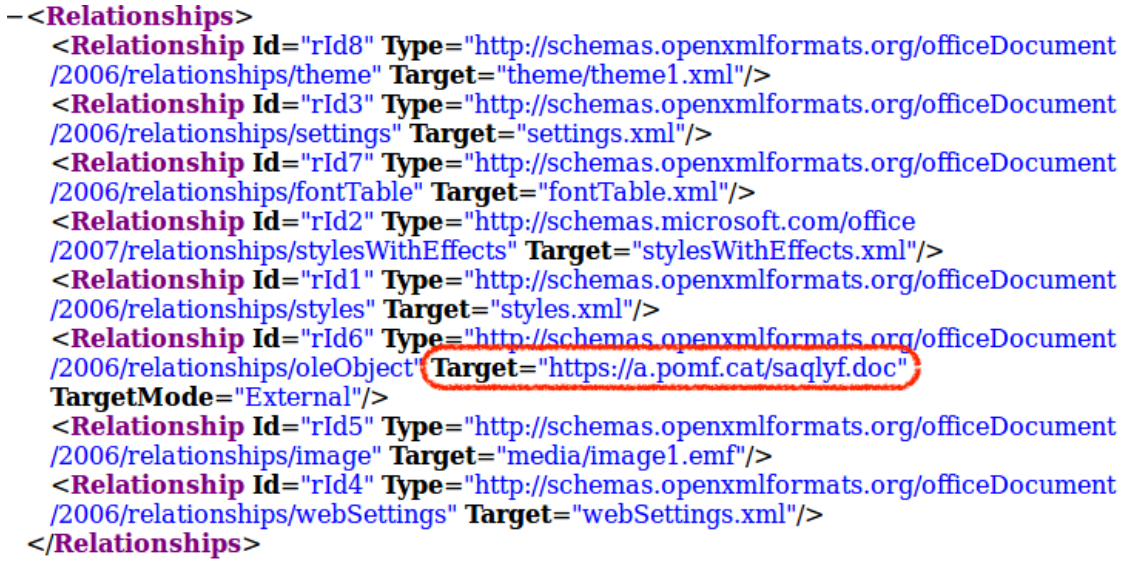
The initial document was reported by @xme on Twitter. A quick check using oletools indicates that the file has the OpenXML format and no macros.

```
FILE: Product Description.docx Type: OpenXML No VBA macros found.
```

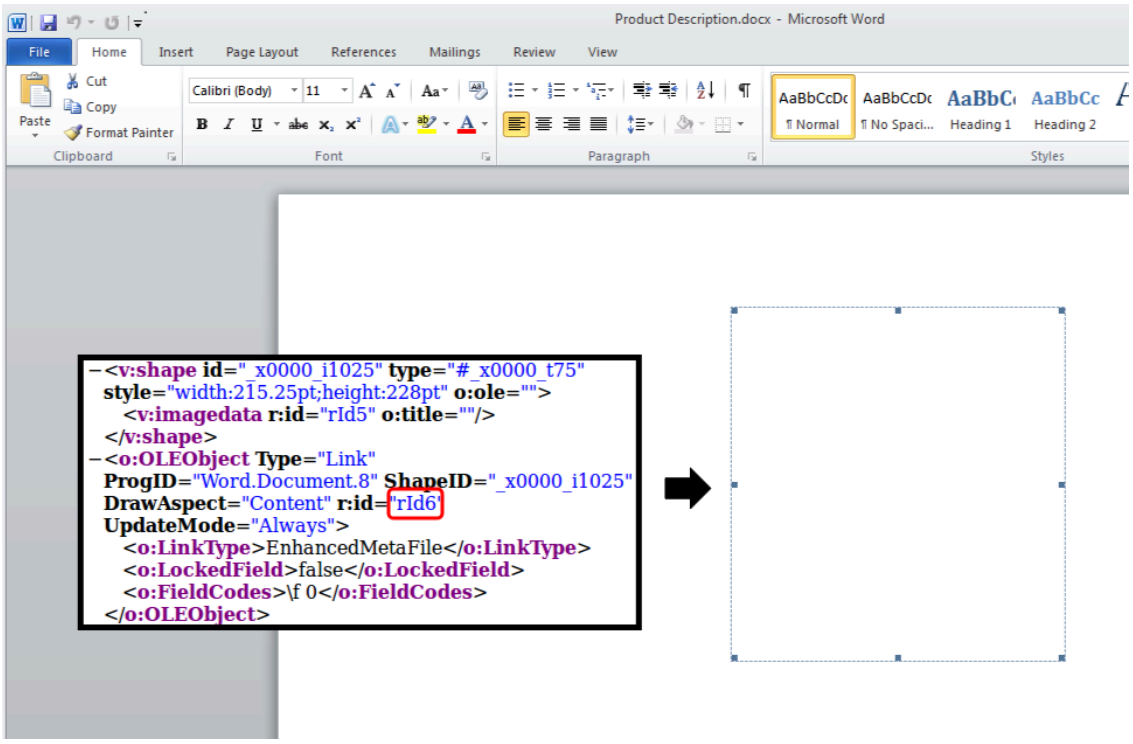
Since OpenXML files are archives, they can be decompressed to reveal their content.

```
[CONTENT_TYPES].XML _RELS/.RELS WORD/_RELS/DOCUMENT.XML.RELS WORD/DOCUMENT.XML WORD/MEDIA/IMAGE1.EMF
```

Opening *document.xml.rels* reveals an interesting external URL, pointing to another document.



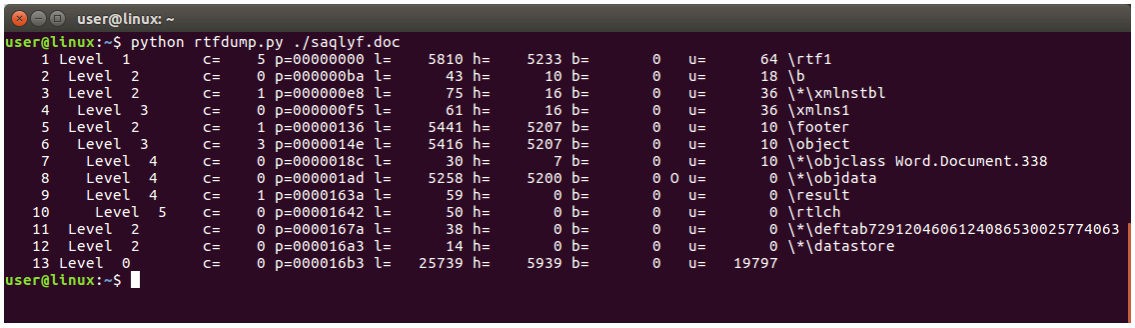
The relationship with *Id=”rID6"* is loaded by the main *document.xml* file. If we open the document without network connectivity (to prevent the automatic execution), we can spot where this object is located.



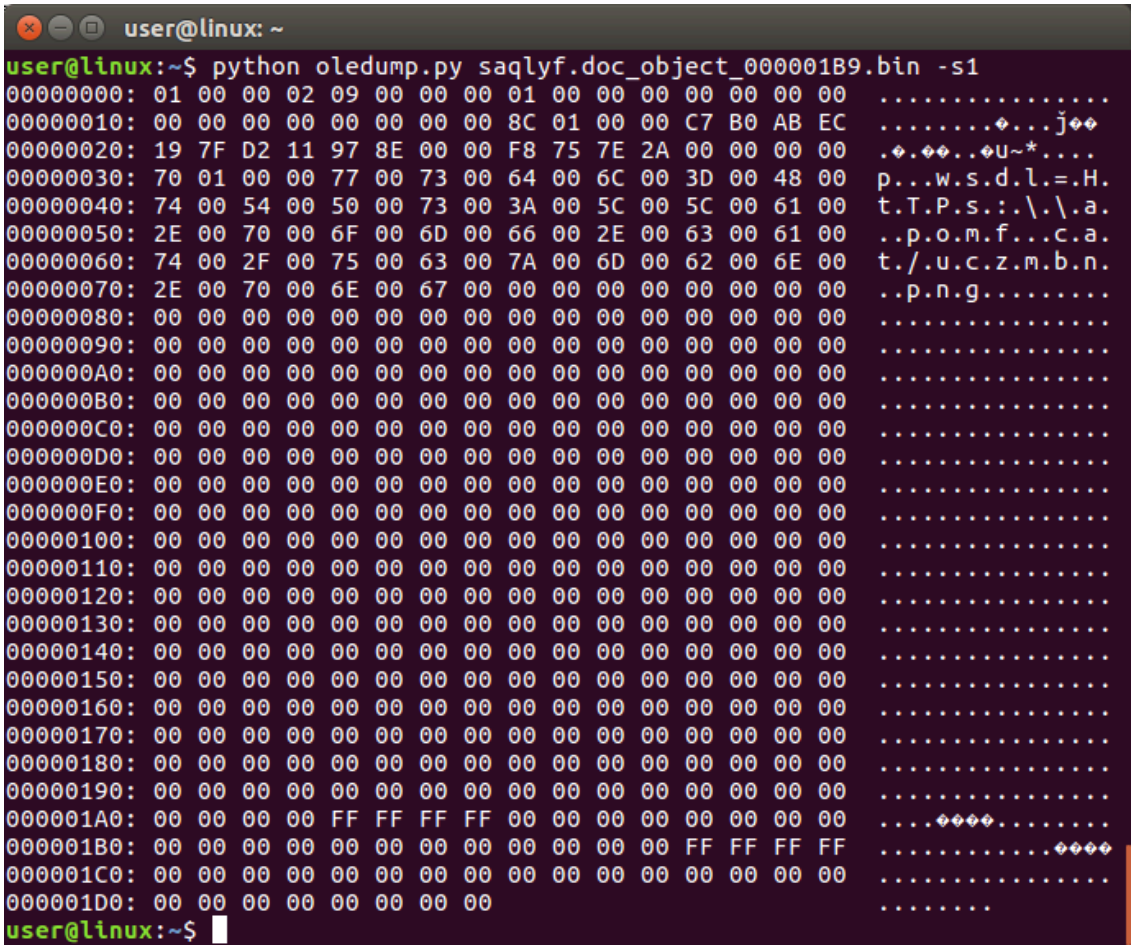
The actual exploit: CVE-2017-8759

The remote file *saqlyf.doc* is downloaded and opened by *Product Description.docx* into the Temporary Internet Files folder.

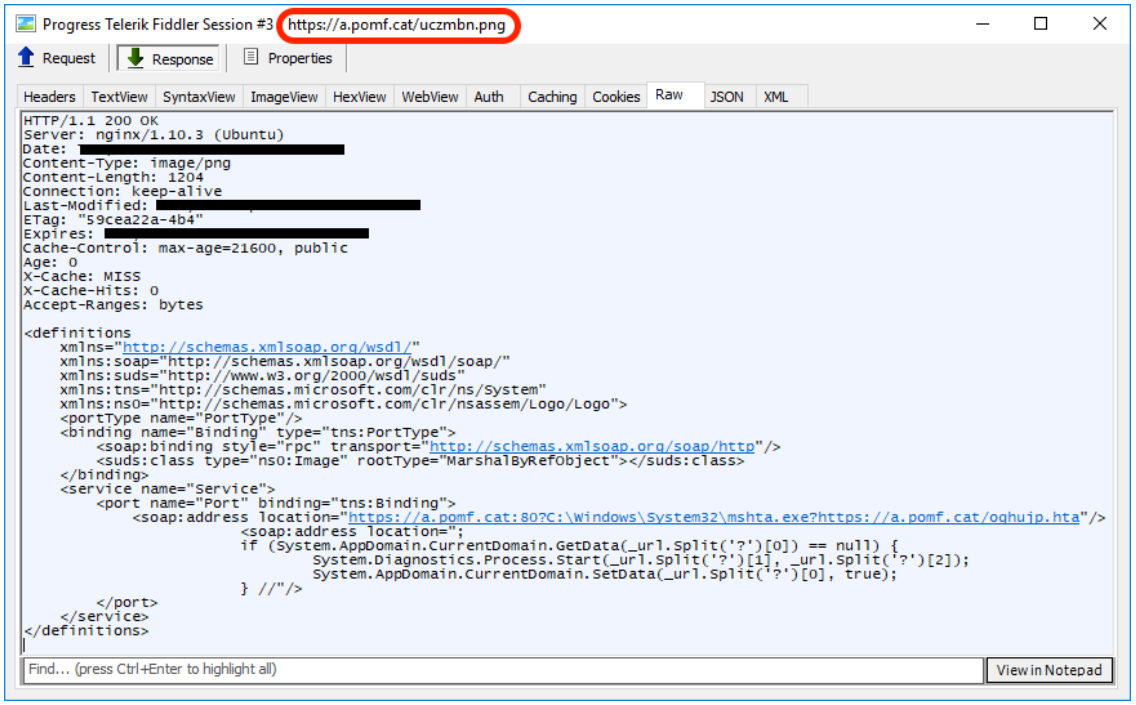
This time, it is an RTF file.



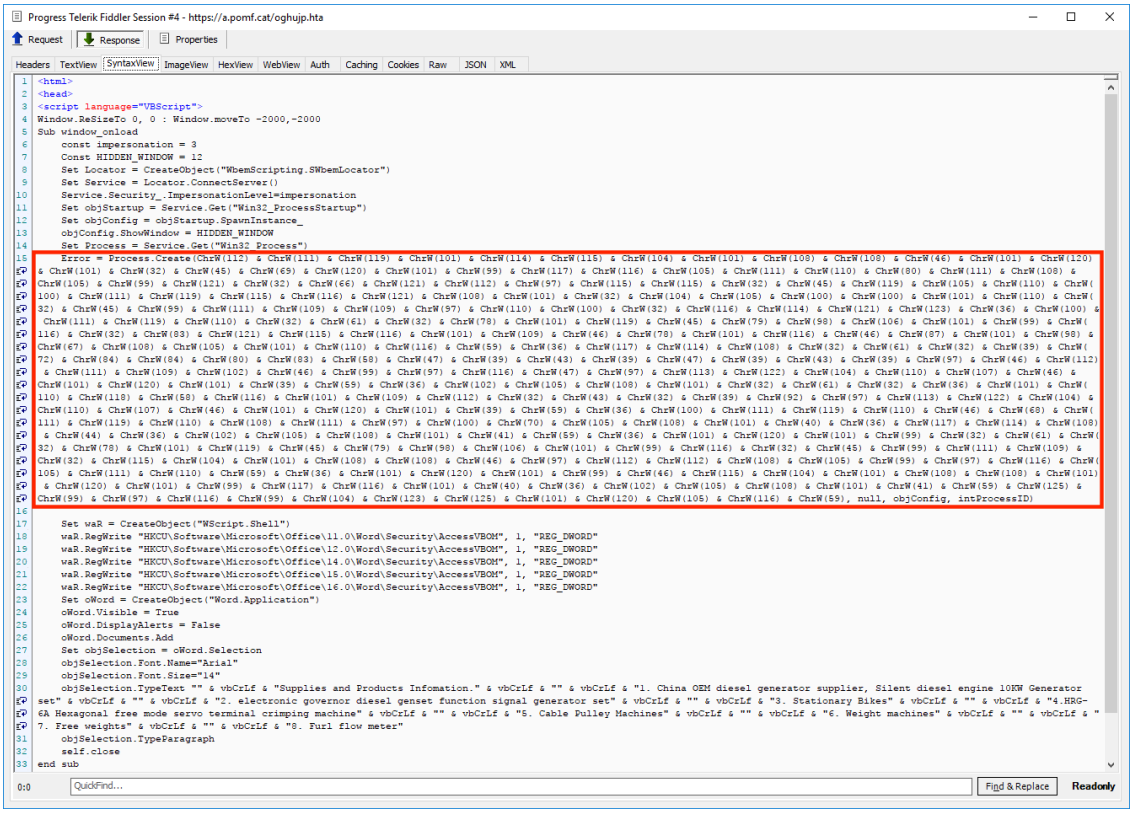
After we convert the hexadecimal encoding to binary (oledump), we can spot another interesting URL.



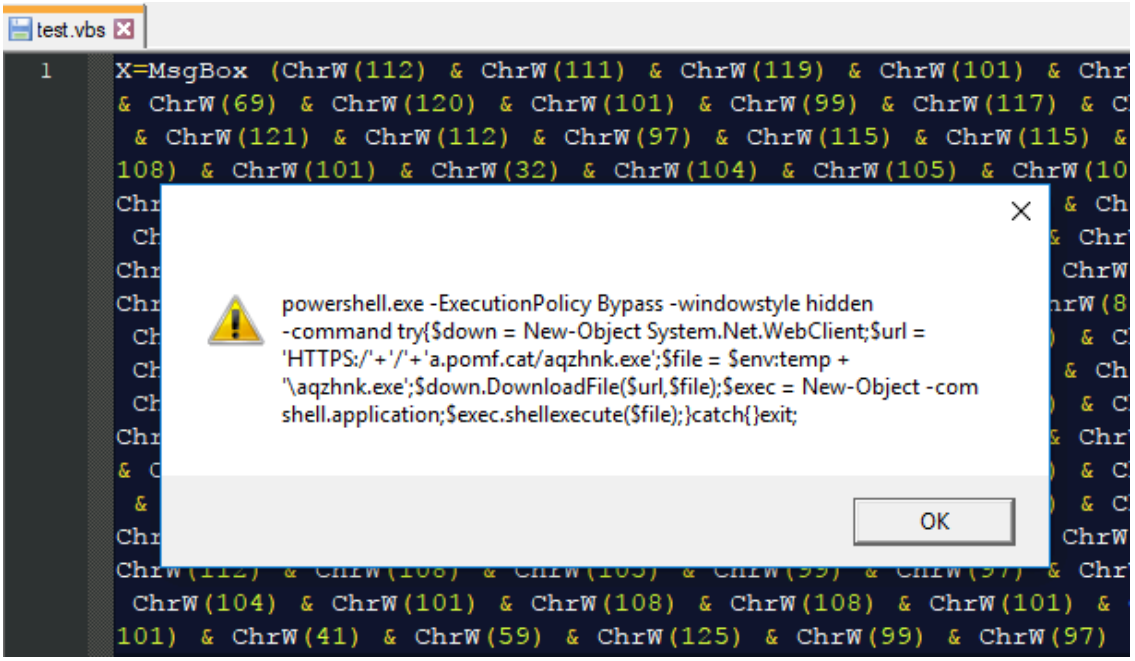
At this point, we could be looking at CVE-2017-0199 if the server provided a MIME type response of application/hta. But in this case, we have something different, and we can quickly spot the SOAP-related bug associated with CVE-2017-8759.



The above code will parse and execute the content of the *oghujp.hta* file pictured below.



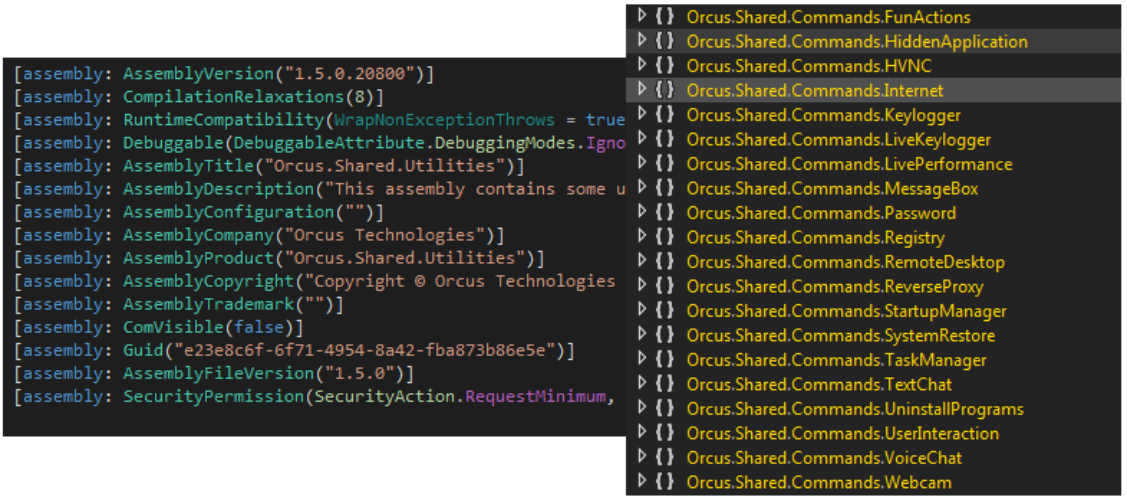
The nasty bit is encoded with ChrW but we can let VBScript do the work and output what it is in human, readable terms.



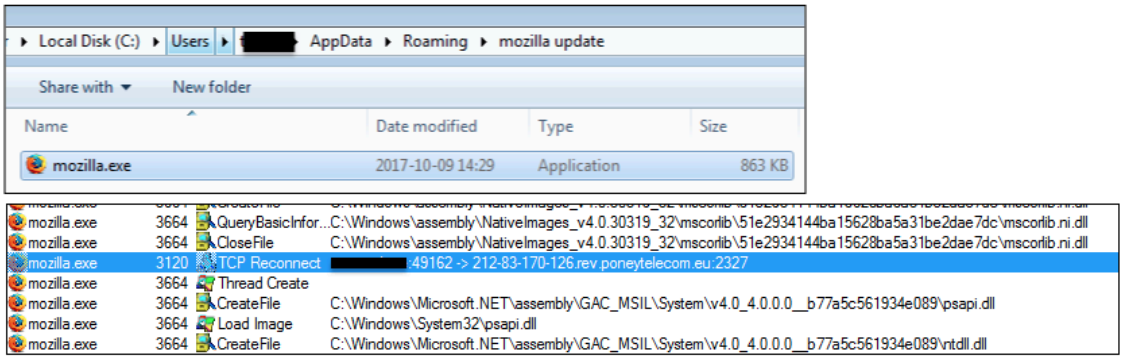
This is the final part of the exploitation phase, and it involves running PowerShell to download and run a binary.

Attack payload: a RAT

This attack was meant to install a commercial Remote Administration Tool known as Orcus Rat, which as seen previously was also hosted on the same server containing the exploit. The program is written in .NET and contains functions such as keylogging, remote desktop, or access to the webcam.



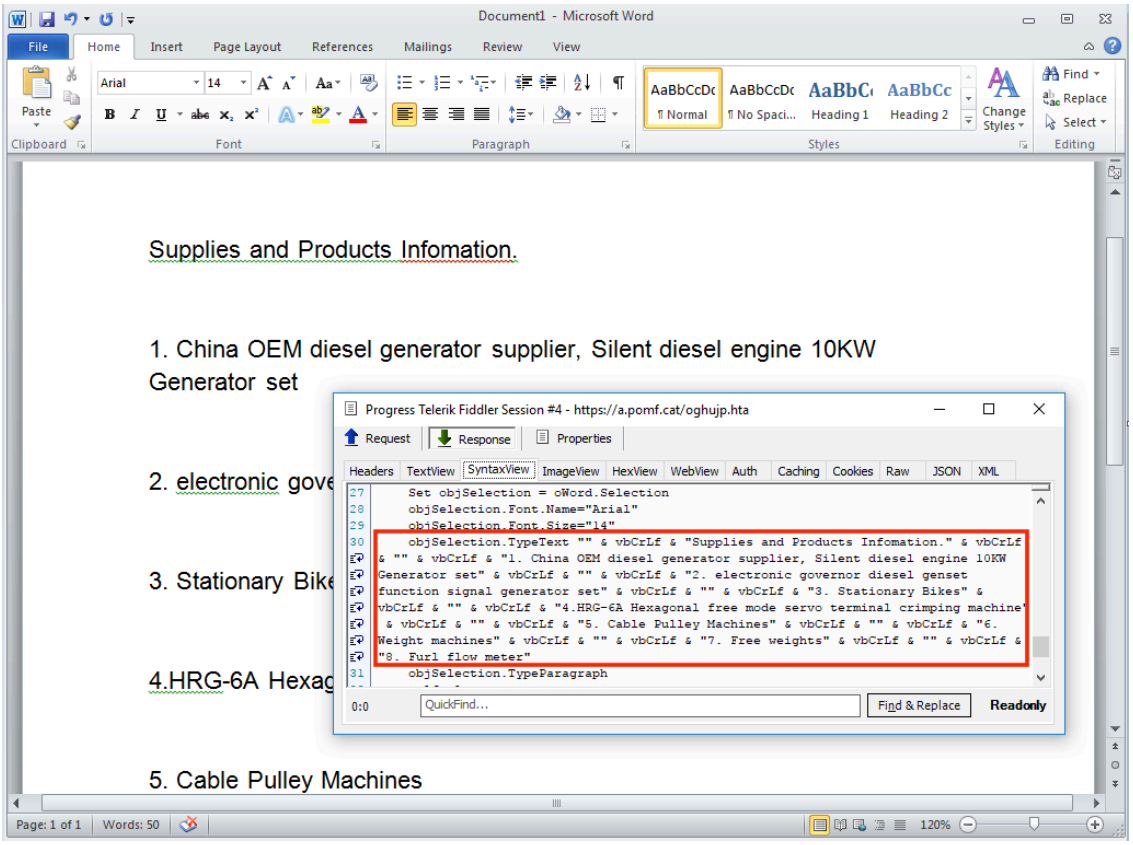
The file is concealed as *mozilla.exe* and periodically checks with its command and control infrastructure.



While commercial RATs can be used for legitimate purposes, malicious actors often abuse them for their own sinister goals.

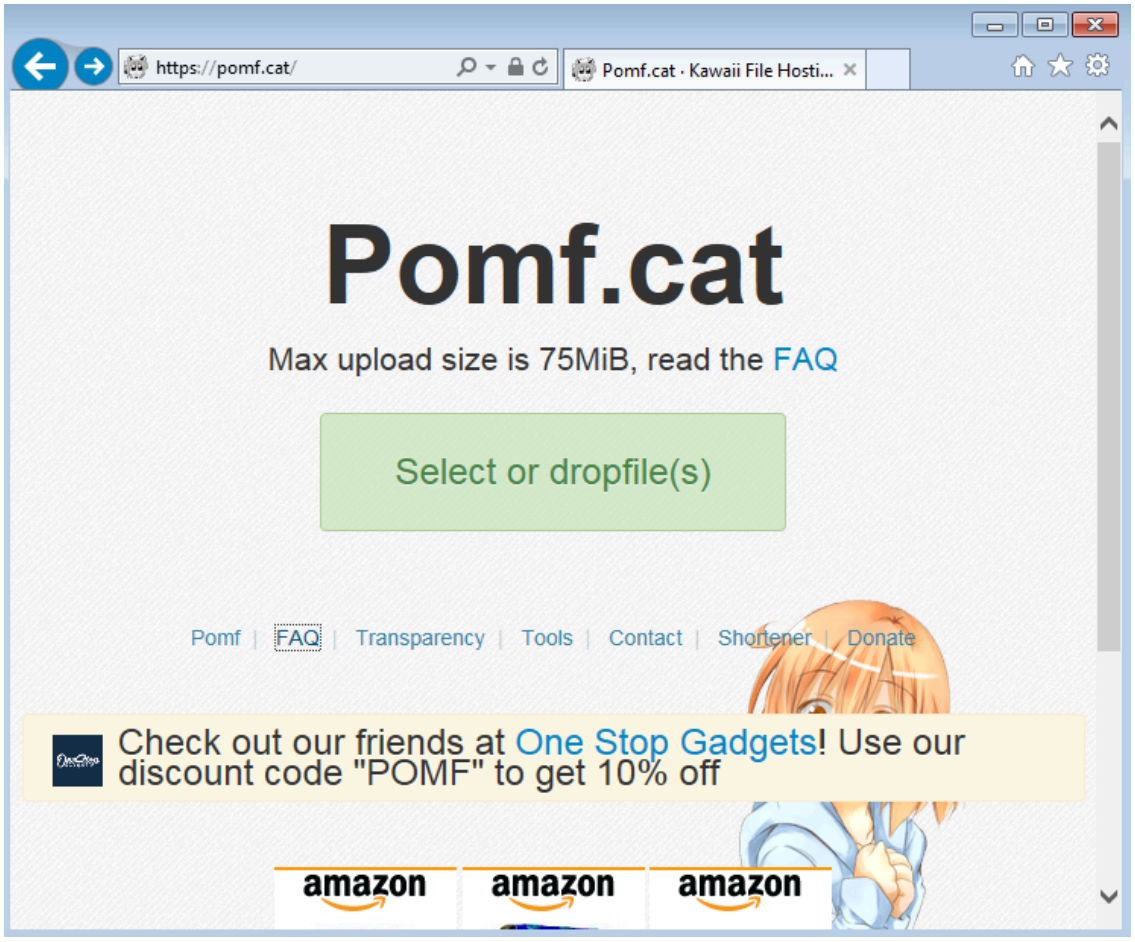
Diversion

Part of the malicious VBScript creates a fake document on the fly that is displayed to the user. If you look carefully, you will notice that the file is called *Document1*, therefore it's an additional file to the original *Product Description.docx* one. It also contains too many typos (but that's a debate for another day).



Attack infrastructure

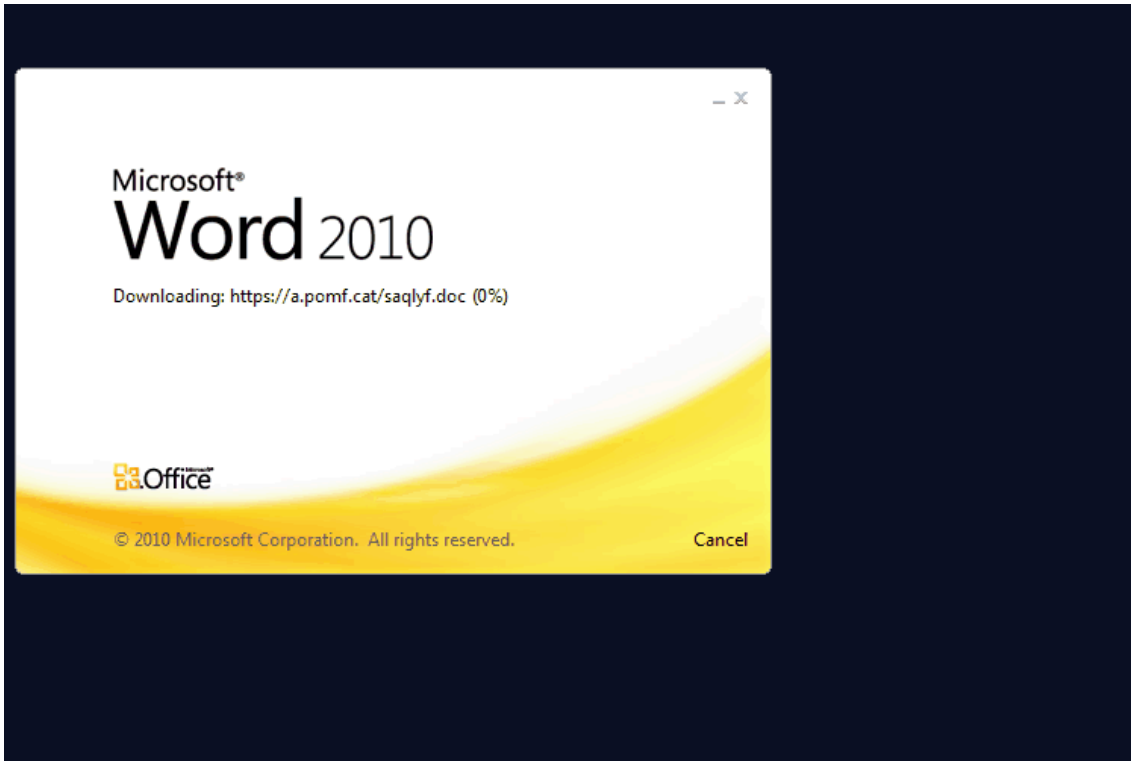
The exploit and payload used in this attack are served from a free file hosting site at *pomf[.]cat.*



A cursory look at the site revealed that many other malicious files are also hosted on this platform. We have reached out and requested a takedown of the offending files.

Scanning for the original document at the gateway may not have returned anything due to its relatively benign nature, and this is why protection at the end point is so important. More and more attacks these days are modular and retrieve payloads on the fly in order to evade detection.

Malwarebytes users are already protected against this exploit. Additionally, we detect the RAT as Backdoor.NanoCore.



Indicators of compromise

Initial document (Product Description.docx)

```
01e45e5647f103ccc99311066d0625f24e79ec8462b131d026b7a557a18d7616
```

RTF (CVE-2017-8759)

```
a.pomf.cat/saqlyf.doc 5758c31928c5f962fbb3ec2d07130e189a8cf4f3fbd0cd606cb1c1d165334a1c
```

PNG (CVE-2017-8759)

```
a.pomf.cat/uczmbn.png 5ed4582313d593a183ab0b8889dc3833c382ce9ca810287d0fcf982275b55e60
```

HTA (CVE-2017-8759)

```
a.pomf.cat/oghujp.hta b048a2d2ea3bb552ac6e79e37fc74576a50c79b4d8c9fd73b1276baabc465ebf
```

Payload (RAT)

```
a.pomf.cat/aqzhnk.exe 72041b65777a527667e73ccc5df95296f182e4787f4a349fcbe0220961dd0ed2
```

Source: https://blog.malwarebytes.com/threat-analysis/2017/10/decoy-microsoft-word-document-delivers-malware-through-rat/