URSNIF CAMPAIGN

OVERVIEW

JASK security research department has been able to confirm both by third party sources and by its own research, an ongoing campaign using a new variant of URSNIF. This type of malicious code is being used as a trojan, data stealer, and targeting different industries. This malicious payload is being delivered via email with a MS Word file attached to it.

INDICATORS OF URSNIF

Delivery of this malicious code is usually done via email, in the form of an attached word document. Malicious actors will target non technical personnel and use crafted emails with misleading messages. As the majority of enterprises will automatically block zip files or exes, malicious actors must lean more on social engineering techniques. The target and timing suggest attackers did some pre-texting. Further on, sending a password protected zip file can bypass automatic blocking protection controls. Also a password added to a word document serves as leverage to force the user to bypass code execution controls.

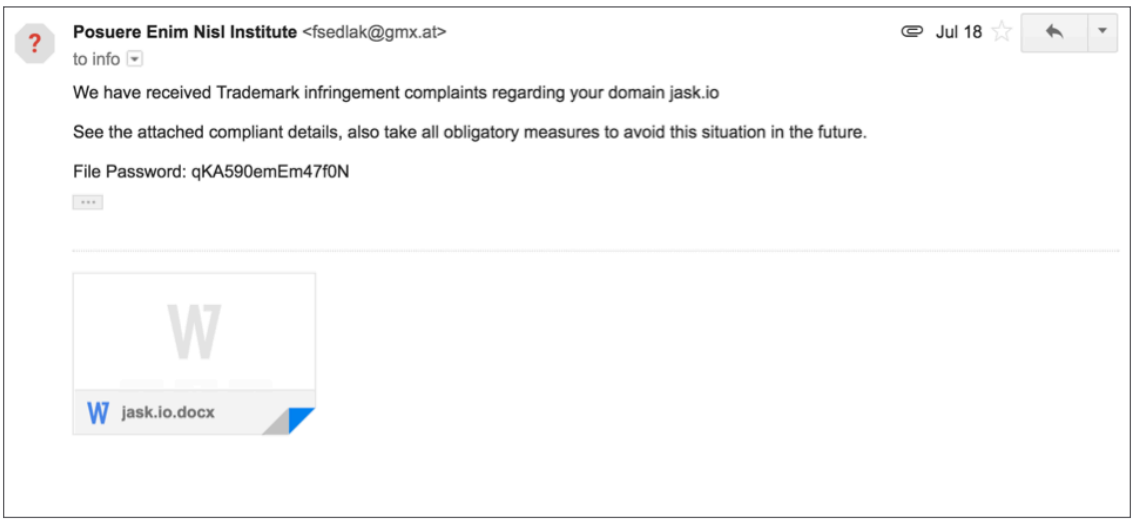

Fig 1 Shows malicious email with attached word document

Some of the known capabilities of URSNIF include*:

- Capture screenshots
- Steal cookies
- Clear cookies
- Steal certificates
- Reboot machine
- Start a SOCK proxy
- Upload a log file that contains user information
- Get a list of active running processes
- Terminate process
- Download and install a new executable
- Steals system and information at rest or in transit. Looks for clear text and protected information

*Further URSNIF indicators can be found at this Trend Micro report as well.

Once this email was spotted by JASK employees, they proceeded to notify the security research department to address this threat.

LAB STUDY

Once the document was placed inside a sandbox several IOC were measured on this document. It is important to highlight that all cloud sandboxes came back negative. A quick entropy check via Didier Steven's tool Oledump revealed the presence of obfuscated code. URSNIF is also known to have anti-sandbox features.

```
jaskrod:oledump_V0_0_28 rodsoto$ python oledump.py jask.io.docx -p plugin_vba_su
mmary.py
  1:       128 '\x05DocumentSummaryInformation'
  2:       164 '\x05SummaryInformation'
  3:        64 '\x06DataSpaces/DataSpaceInfo/StrongEncryptionDataSpace'
  4:       112 '\x06DataSpaces/DataSpaceMap'
  5:       200 '\x06DataSpaces/TransformInfo/StrongEncryptionTransform/\x06Prima
ry'
  6:        76 '\x06DataSpaces/Version'
  7:     76968 'EncryptedPackage'
  8:      1289 'EncryptionInfo'
jaskrod:oledump_V0_0_28 rodsoto$ python oledump.py jask.io.docx -p plugin_vba_dc
o.py
  1:       128 '\x05DocumentSummaryInformation'
  2:       164 '\x05SummaryInformation'
  3:        64 '\x06DataSpaces/DataSpaceInfo/StrongEncryptionDataSpace'
  4:       112 '\x06DataSpaces/DataSpaceMap'
  5:       200 '\x06DataSpaces/TransformInfo/StrongEncryptionTransform/\x06Prima
ry'
  6:        76 '\x06DataSpaces/Version'
  7:     76968 'EncryptedPackage'
  8:      1289 'EncryptionInfo'
```

Fig 2 Shows Oledump tool results

The following figure shows the result of opening the file where a password prompt is shown. This is followed by a request to enable editing mode which effectively allows code execution (Macros, Vbs, Vba) and successfully bypasses security controls. This enables the hostile file to download subsequent payloads.

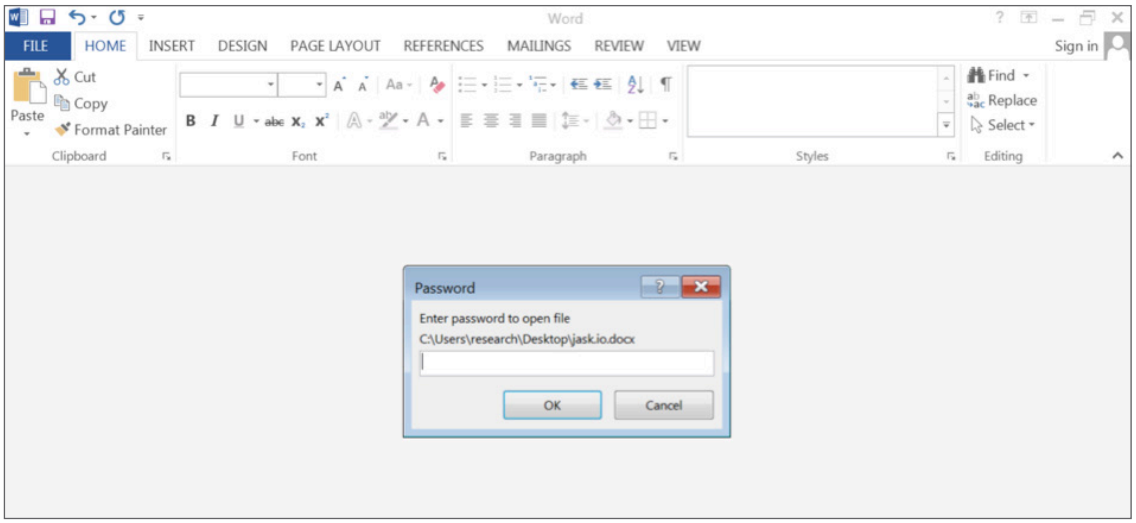

Fig 3 Shows password request

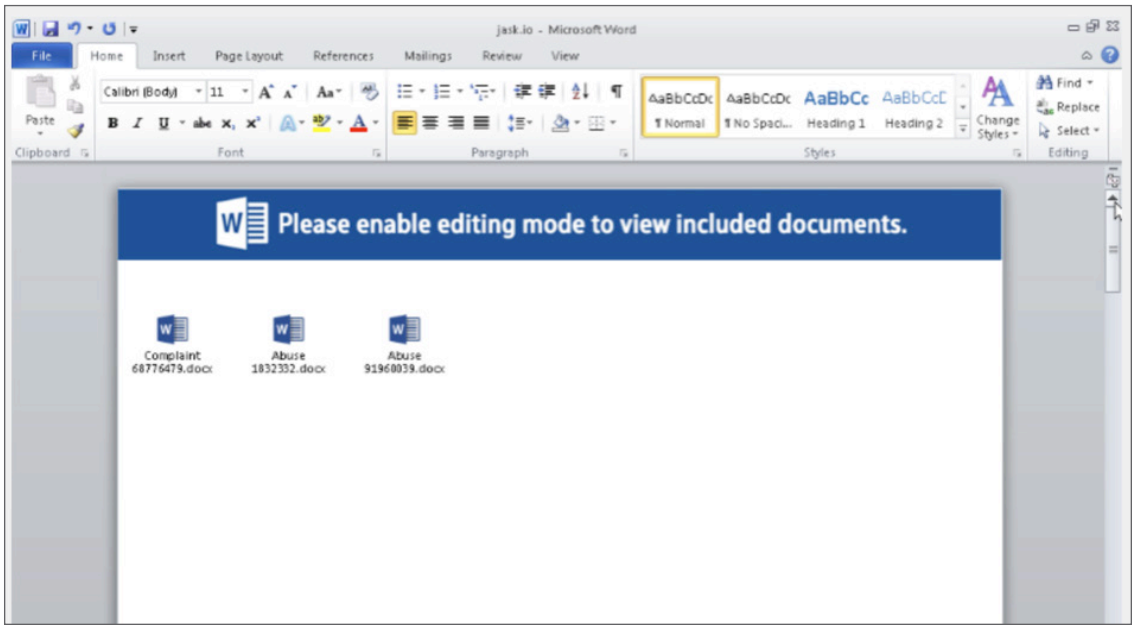

Fig 4 Shows embeded documents

Once editing mode is enabled and the documents are opened, packet analysis shows an outbound connection to an IP in Moscow Russia. Specifically a HTTP GET request to a file named changelog.txt

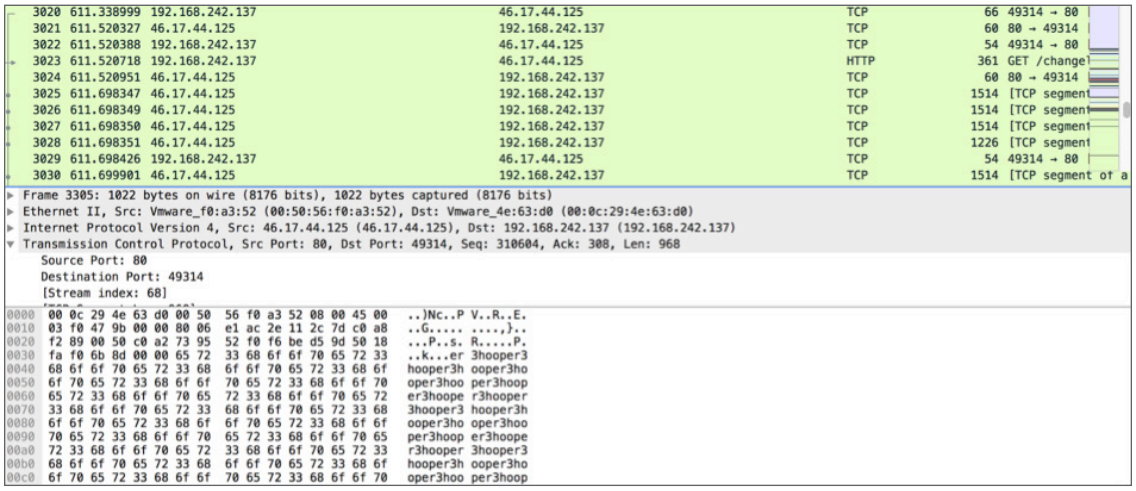

Fig 5 Shows outbound connection to 46.17.44.125

Fig 6 Shows TCP Stream

Most cloud sandbox services were not able to come up with an accurate detection of this threat. Further de-obfuscation and analysis was required. After such analysis was performed the following indicators were found as well:

- Malicious code downloads multiple payloads, specifically
- changelog.txt.exe & and second get request to http://boxerphotography.com.au/d.rtf.exe drops these:
  - scs32.tmp -- md5: 4a587187d760161311010b03417b3c3f, type: ASCII text, with CRLF line terminators
  - scs33.tmp -- md5: 71f4b39c5eb73df738ad3e0dacd89057, type: DOS batch file, ASCII text, with CRLF line terminators

The second GET request to the .AU shows a multi-stage payload used by code. Here is a link to a deeper sandbox analysis of boxerphotography.com.au/d.rtf. Sandbox analysis found suspicious indicators including gathering of system information, contacting external domains, spawning new processes, modifies, proxy settings, hooks/patches running processes among others.

It is common within the infosec community, to share and contribute in the research of threats. The indicators found in the payloads targeting JASK matched those specified by Wapack Labs in their URSNIF campaign advisory. Further analysis by performing PDB debugging, found traces similar in previous URSNIF samples as well.

IDENTIFIED MICRO BEHAVIORS

JASK Trident already performs analysis of this type of attack and payload delivery. By defining and dissecting the micro behaviors present in this attacks we can find the following:

- Malicious file download: changelog.txt.exe
- Unusual download to Non TLD IP **46.17.44.125**
- C2—traffic pattern to **46.17.44.125**
- Subsequent executable download: d.rtf.xe
- Outbound communication to an unusual port 80 > 49314
- Entropy, binary, obfuscation like in TCP stream

JASK Trident detects the above Micro behaviors and produces alerts on exploit delivery condition. Further on Trident “Enrichments” can add on to this exploit delivery scenario by adding, Threat Intelligence indicators (VT, Payload Security, Malwr, AlienVault, etc), along with firewall or endpoint information (PAN, Cylance, Carbon Black).

Below an example of JASK trident detection of this exploit delivery.

```
%sql
select filename
from file
where
((filename like "%changelog%"
or filename like "%rtf%")
and dst_port rlike '^([1-4][0-9][3-9][0-9][0-4]|[1-9][0-9][0-9][0-9]|[8-9][0-9]|[1-9][0-9][0-9])$')
and dst_ip.address = "46.17.44.125"
```

Fig 7 JASK Trident

RECOMMENDED MITIGATION

1. Security awareness in organizations is the first line of defense against this type of threats. JASK employees were able to identify the suspicious nature of this communication and referred it to security department.
2. Refer or notify of suspicious communications with attachments. These emails, sms, pop ups or unsolicited chat messages can be deleted as well, however remember your organization might be under attack and it is important to let others know, specially the security department.
3. Use common sense when looking at suspicious emails, certain positions cannot do their jobs without opening emails (HR, Estimation, Architecture, Marketing, etc).
4. The use of AV even though is passive and easily bypass measure can help at times to prevent these type of threats.
5. If possible DISABLE Macros in Microsoft Office or apply Microsoft suggested macro control procedure**.
6. Enforce principle of minimum privilege.
7. Segmenting network, can prevent further infestation.
8. Block macros in files originating from the Internet and external email systems (Office 2016).
9. When viewing attachments use Microsoft User viewers as they enable document viewing without enabling Macros.

JASK

CONTRIBUTORS

Wapack Labs
Robert Simmons
Hackmiami

ABOUT JASK.AI

JASK monitors networks end to end, surfacing, triaging and mapping the most relevant attacks at unprecedented speed, using advanced AI. Analysts are empowered to make informed decisions faster and with more precision.

www.jask.ai