THREAT INTELLIGENCE LAB (CS-5202)

2021-2022

**YARA RULE FOR ARTRA DOWNLOADER**

**Arjun Anil (MT20ACS499)**

**Prof. Debashis Sengupta Dr. Ashu Sharma Professor and Area Director Assistant Professor**

Contents

[Artra Downloader Yara Rule 3](#_Toc81249480)

[**Artra Downloader Malware Family** 3](#_Toc81249481)

[**Activities** 3](#_Toc81249482)

[**Working** 3](#_Toc81249483)

[**Details Collected from Virustotal** 4](#_Toc81249484)

[**Describing Yara Patterns** 5](#_Toc81249485)

[**Strings Explained** 7](#_Toc81249486)

[**References** 7](#_Toc81249487)

# Artra Downloader Yara Rule

## **Artra Downloader Malware Family**

In total, roughly 80 unique instances of the Artra Downloader 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 malware family was named Artra Downloader based on the PDB string discovered on the samples by Palo Alto Network’s Unit 42 team. This downloader has frequently been observed downloading the Remote Access Trojan (RAT) BitterRAT which is associated with BITTER threat operations.

### **Activities**

Artra downloader was used by a South Asian threat grouping known as BITTER for targeting Pakistan and Saudi Arabian organizations. This downloader, leveraging unique custom obfuscation routines downloads and executes the BitterRAT malware family via HTTP.

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

### **Working**

Variant-1

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.

### **Details Collected from Virustotal**

File Type : Win32 EXE

Vhash : 034056655d15155048z18hz22z145zc2z3a1z

Authentihash : 809e1b4e7f0bcb931b25c3762fe00bb0c5427a5498ef106f1eb13f14bffba520

Imphash : 2dfea0cb99dfd178aefd42a0ebb0ea98

Threat Name : Artemis!Trojan (Trojan Downloader)

Target Machine: Intel 386 or later processors and compatible processors

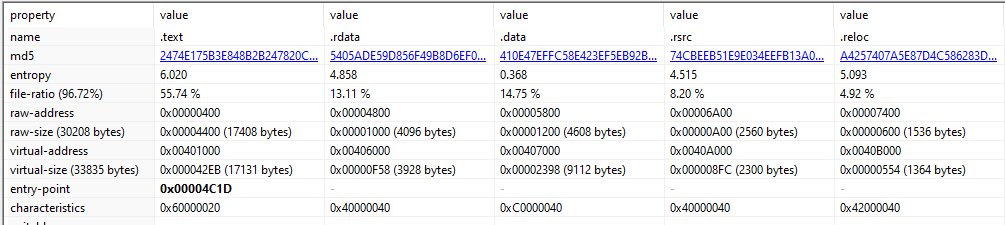
Entry Point : 19484

File Size : 30.5 kB

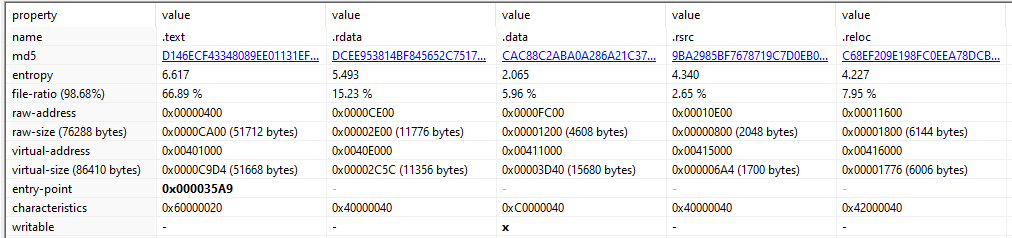
IP Traffic : 162.222.215.90:80

**From Pestudio:**

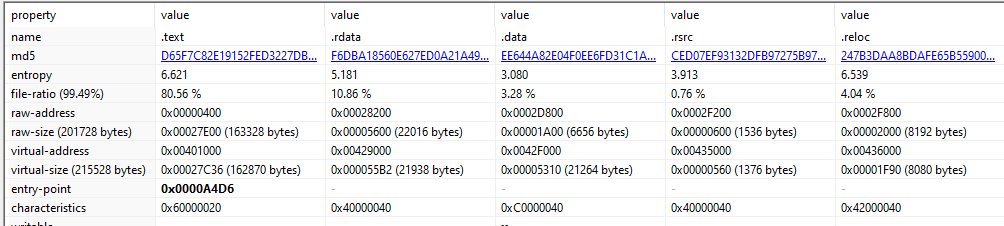
File: 0b2a794bac4bf650b6ba537137504162520b67266449be979679afbb14e8e5c0



File: ef0cb0a1a29bcdf2b36622f72734aec8d38326fc8f7270f78bd956e706a5fd57

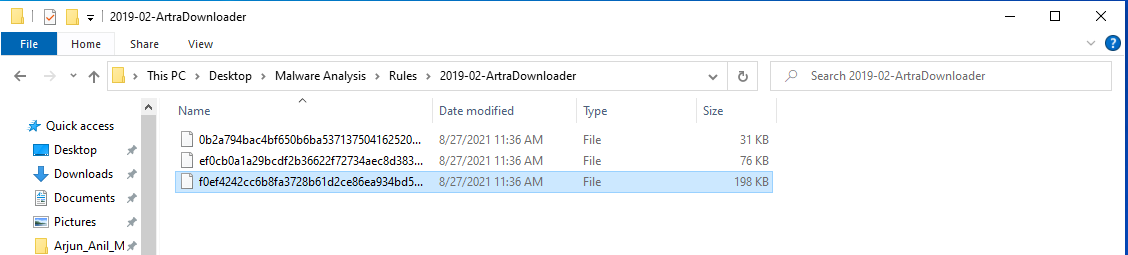


File: f0ef4242cc6b8fa3728b61d2ce86ea934bd59f550de9167afbca0b0aaa3b2c22

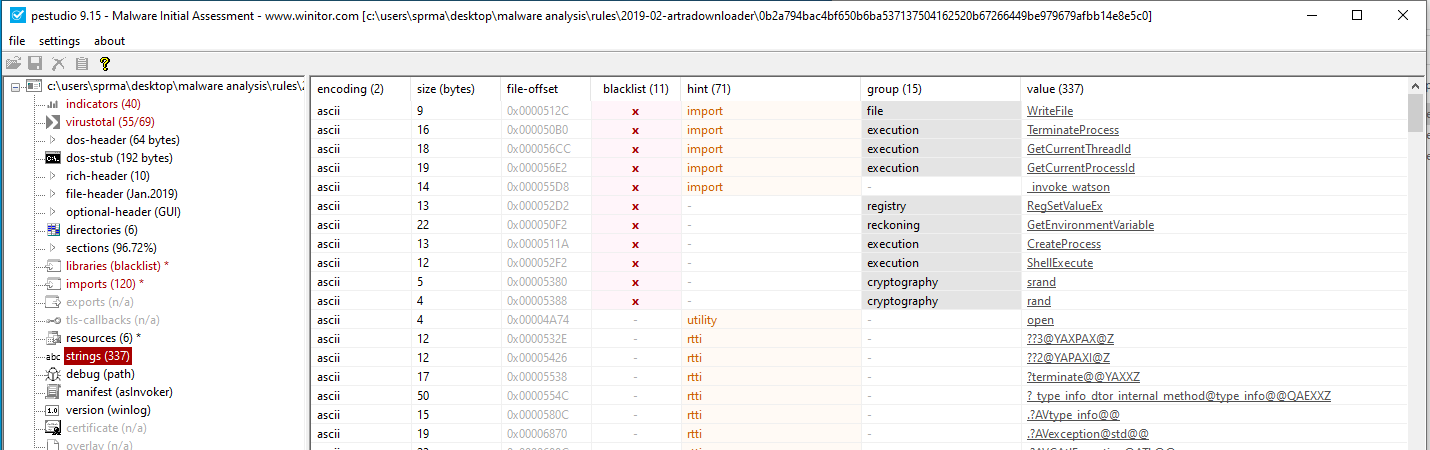


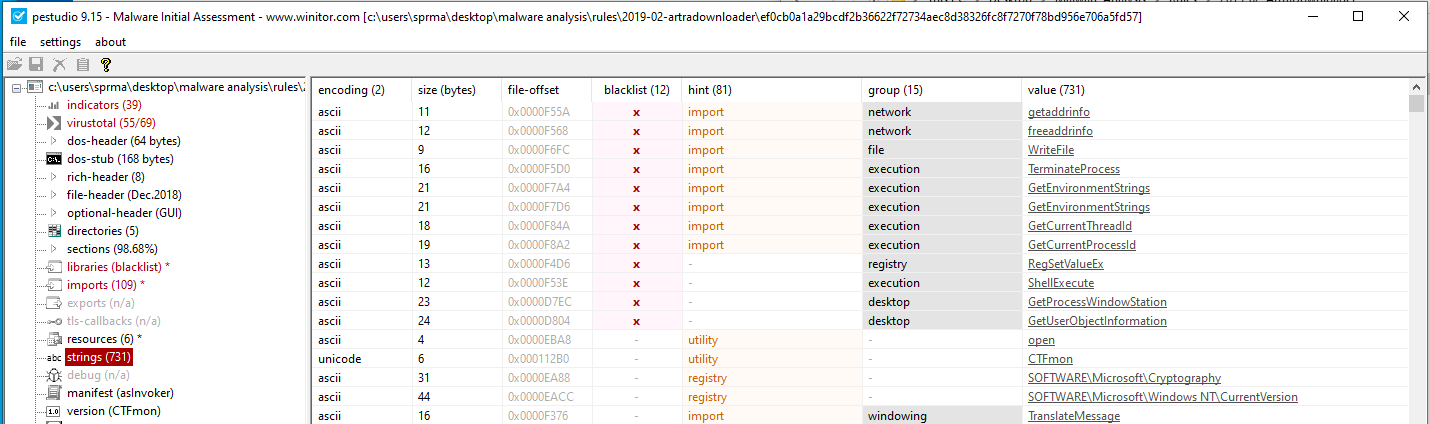
## **Describing Yara Patterns**

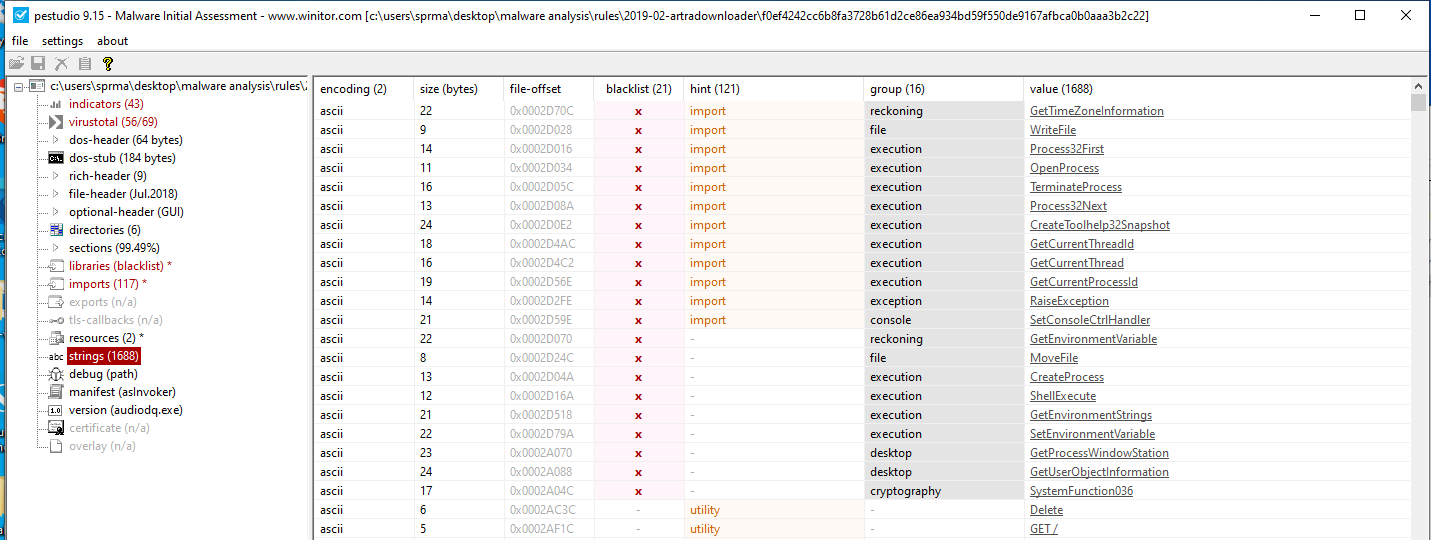
In the Artra Downloader malware family, below were the list of files for analysis.



Opened the files in pestudio and below are the list of strings on each file. Sorted the strings and selected the blacklisted strings from each malware file.







From these blacklisted strings, identified the list of strings which are appearing on all the malware files, they are : TerminateProcess, WriteFile, GetCurrentThreadId, GetCurrentProcessId and ShellExecute.

With the common strings identified, created a yara rule with condition satisfying all these strings present. The yara rule is as below:

rule search\_exe

{

meta:

author = "Arjun Anil"

description = "To find malwares coming under Artra Downloader"

strings:

$a = "TerminateProcess"

$b = "WriteFile"

$c = "GetCurrentThreadId"

$d = "GetCurrentProcessId"

$e = "ShellExecute"

condition:

$a and $b and $c and $d and $e

}

### **Strings Explained**

* "TerminateProcess" : Not meant to be used in 'Release' applications. They are reserved for debugging tools and exceptional applications such as process managers. Clearly, they cause great damage to a system.
* "WriteFile"
* "GetCurrentThreadId" : Retrieves the thread identifier of the calling thread.
* "GetCurrentProcessId" : Retrieves the process identifier of the calling process.
* "ShellExecute" : Performs an operation on a specified file.

## **References**

1. <https://raw-data.gitlab.io/post/ghidra_artra/>
2. <https://unit42.paloaltonetworks.com/multiple-artradownloader-variants-used-by-bitter-to-target-pakistan/>