INTRO starts here

[CONTROL SLIDE]

A tale about Threat Intelligence and magic rabbits

UNbreakable Romania

Dan Demeter, Security Researcher

kaspersky

Nathan
Rothschild
1777-1836

"Who owns the information,
he owns the world."

“If you are unable to see the problem,
It doesn’t mean there is no problem.”

Common Sense


Google
images
WAR

Source: Google Images

Source: Google Images

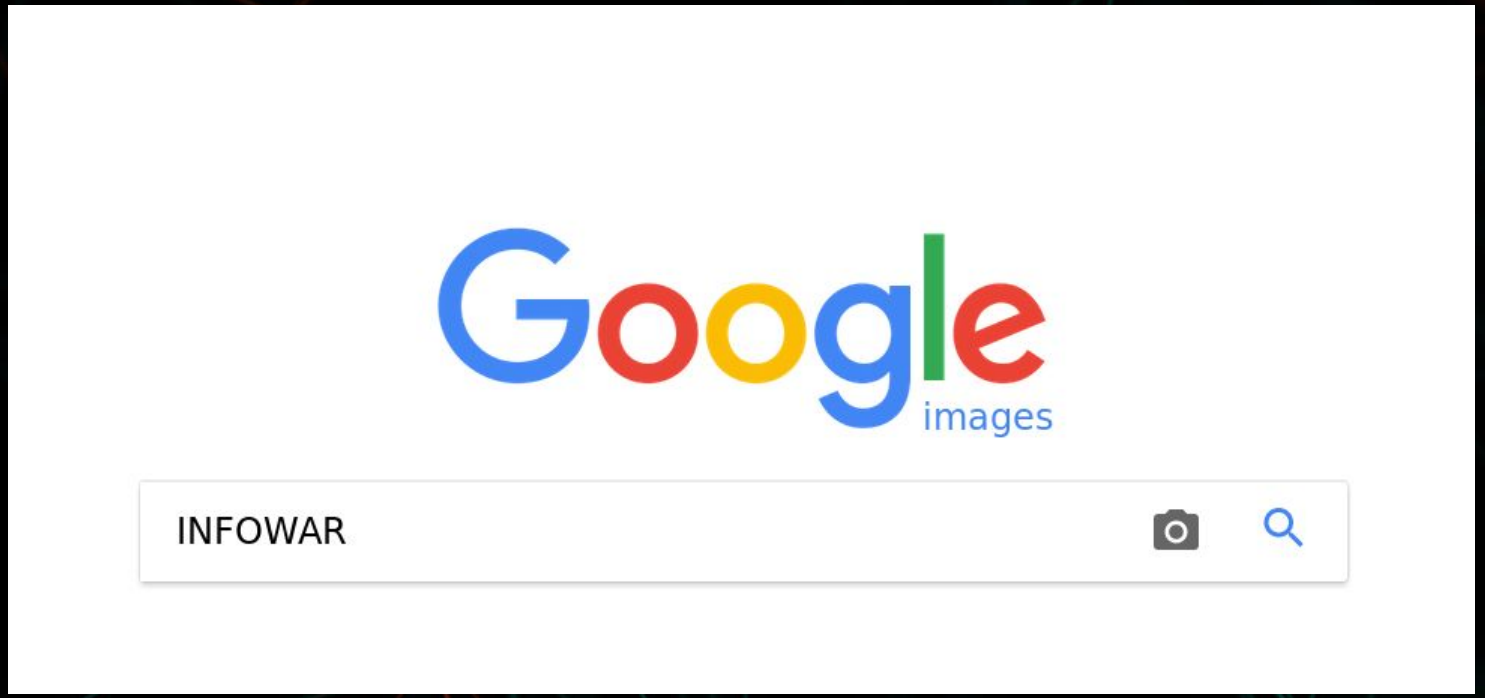
Source: Google Images

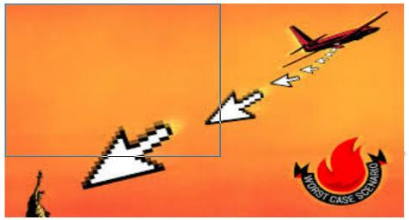
Source: Google Images

Source: Google Images

Source: Google Images

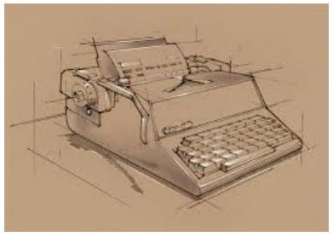

WAR
Source: Google Images


Google
images
INFOWAR


WORST CASE SCENARIO


1300 × 867 - entrepreneur.com


Thanks for watching ● Please Like ▶ Share ▶ Comment ▶ My channel Thanks ♥♥♥




HACKERS

Source: Getty Images


Source: Getty Images

Source: Getty Images

Source: Getty Images

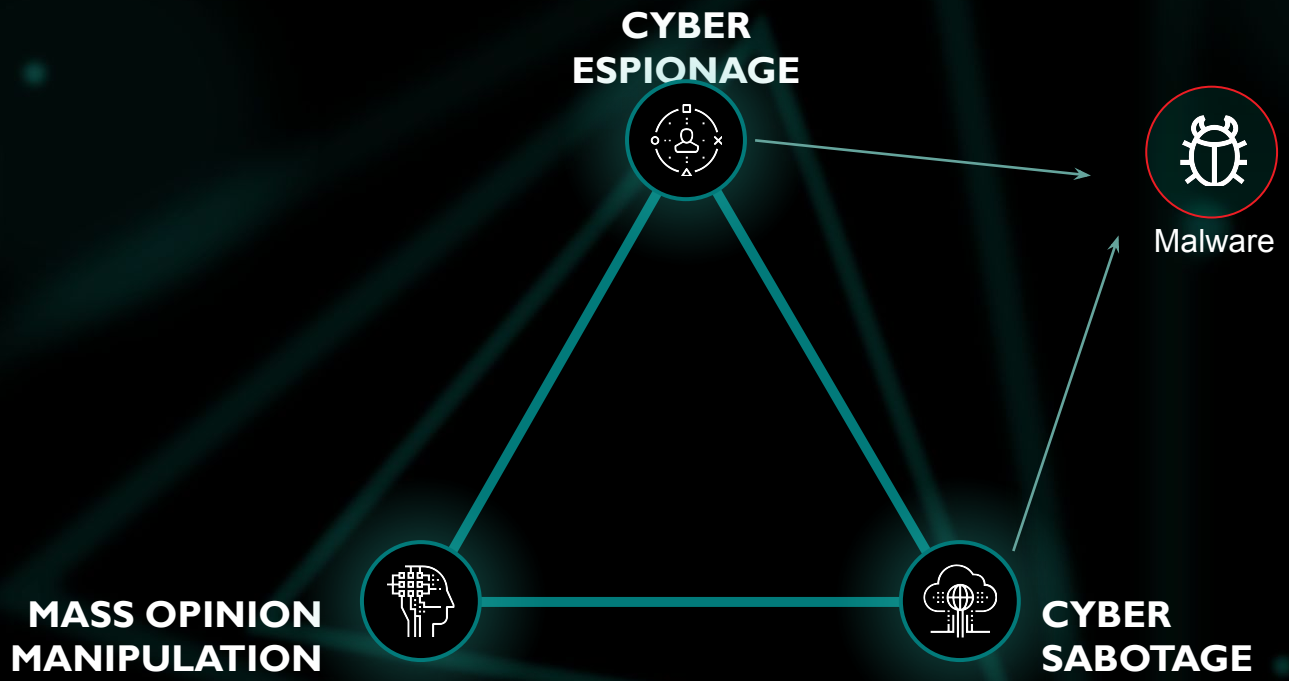
Source: Getty Images

Source: Getty Images

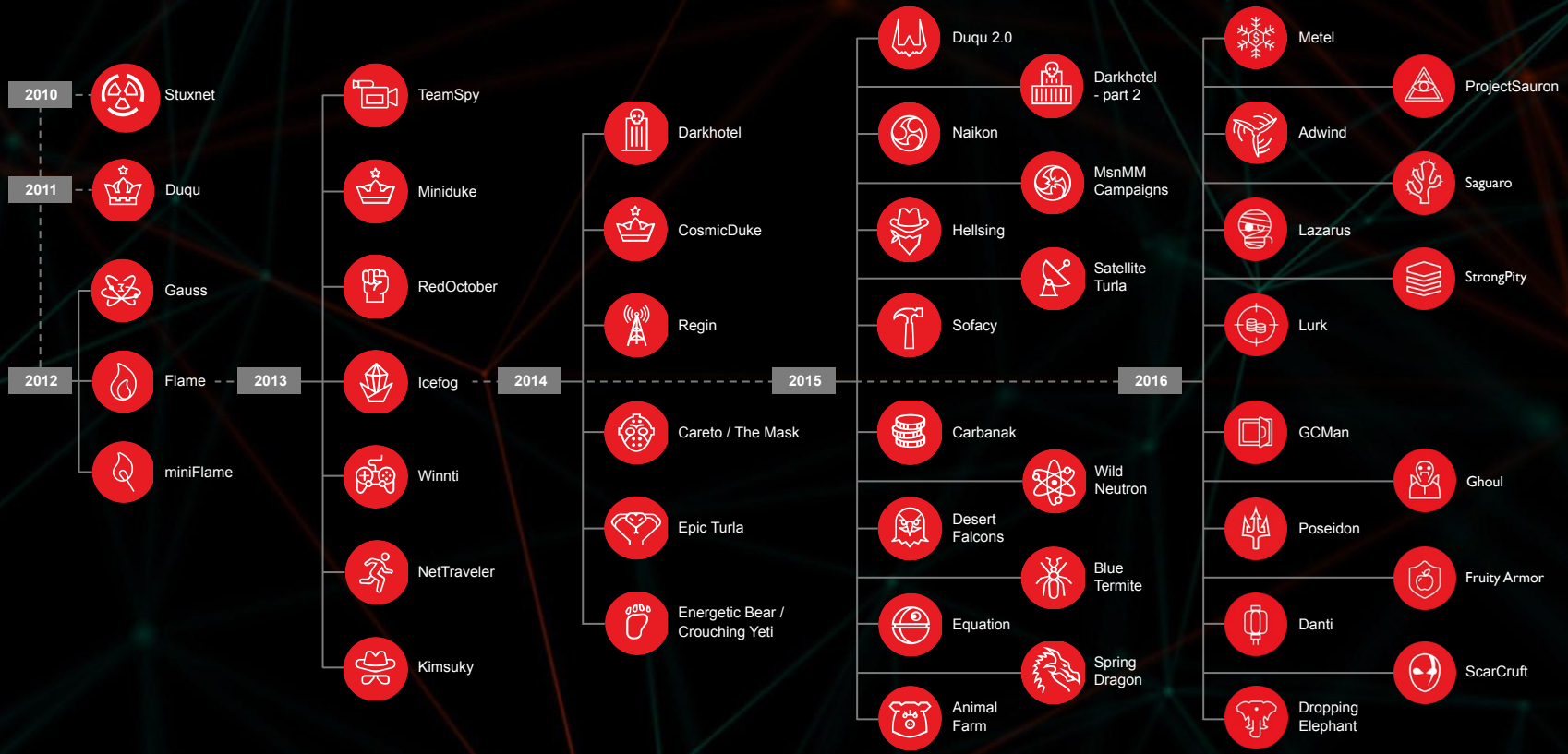

Source: Getty Images
INFOWAR


PENETRATION TESTING
TARGETED ATTACK DISCOVERY
DIGITAL FORENSICS
MALWARE ANALYSIS
INCIDENT RESPONSE
SECURITY TRAININGS

THE INFORMATION WAR



OUR RESEARCH (BEFORE 2017)



About Speaker

Dan Demeter

- Security Researcher at GReAT, Romania
- Like OPSEC, mountain hiking and snowboarding
- Usually builds tools helping smart people do Threat Intel

About Speaker

Date Joined:	January 16 2005 - 13:31:58
Shoutbox Posts:	0
Comments Posted:	7
Forum Posts:	39
Last Visit:	July 03 2018 - 09:59:02
Point Ranking	**2516** of 5001
Community Points	0
Member Status:	Member
Challenges	**Points: 210**
Basic Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
Flash Challenges	1, 2, 3, 4
Javascript Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Realistic Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
Rooting Challenges	1, 2, 3
Application Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
Encryption Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
Logical Challenges	1, 2, 3, 4, 5, 6, 7
Social Challenges	1, 2
Stegano Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
Tracking Challenges	1, 2
Patching Challenges	1, 2, 3, 4
App Patching Challenges	1, 2
Timed Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Other Challenges	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Pen Testing 1	Points accumulated: 0 out of 350
Pen Testing 2	Points accumulated: 0 out of 175
Programming Challenges	

What is GReAT

GREAT
GLOBAL RESEARCH
& ANALYSIS TEAM

Kaspersky's Global Research & Analysis Team

~40 Cybersecurity experts working worldwide, not limited by timezone, geography nor nationality

Investigating APTs

Established 2008

Agenda

TLP: Amber

TLP: Amber

Recipients may only share TLP: Amber information with members of their own organization, and with clients or customers who need to know the information to protect themselves or prevent further harm.

https://www.cisa.gov/tlp

What is Threat Intel
and
why do we need it?

What happened?

Who did it?

Can we fix it?

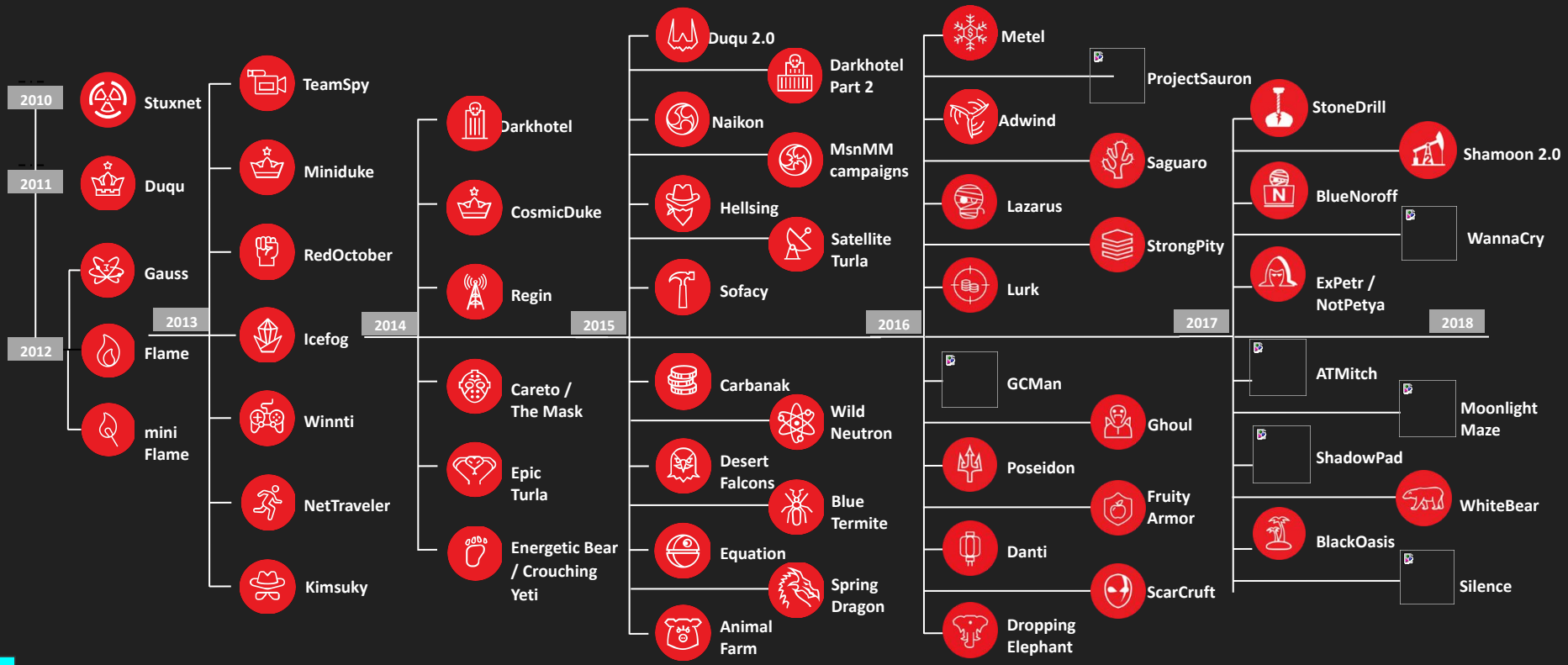
Basics of Threat Intel

- Understanding threats in the wild:
  - Who are they targeting?
  - Why are they attacking, what are they after and how?
  - What are TTPs?
  - Is what I'm seeing commodity malware or not?
- (Practical) Tools of the trade
  - Yara
  - IoCs
  - Sigma
  - etc..
- Resources available
  - Intel platforms
  - Malware sharing platforms
  - Malware Analysis tools

Threat groups

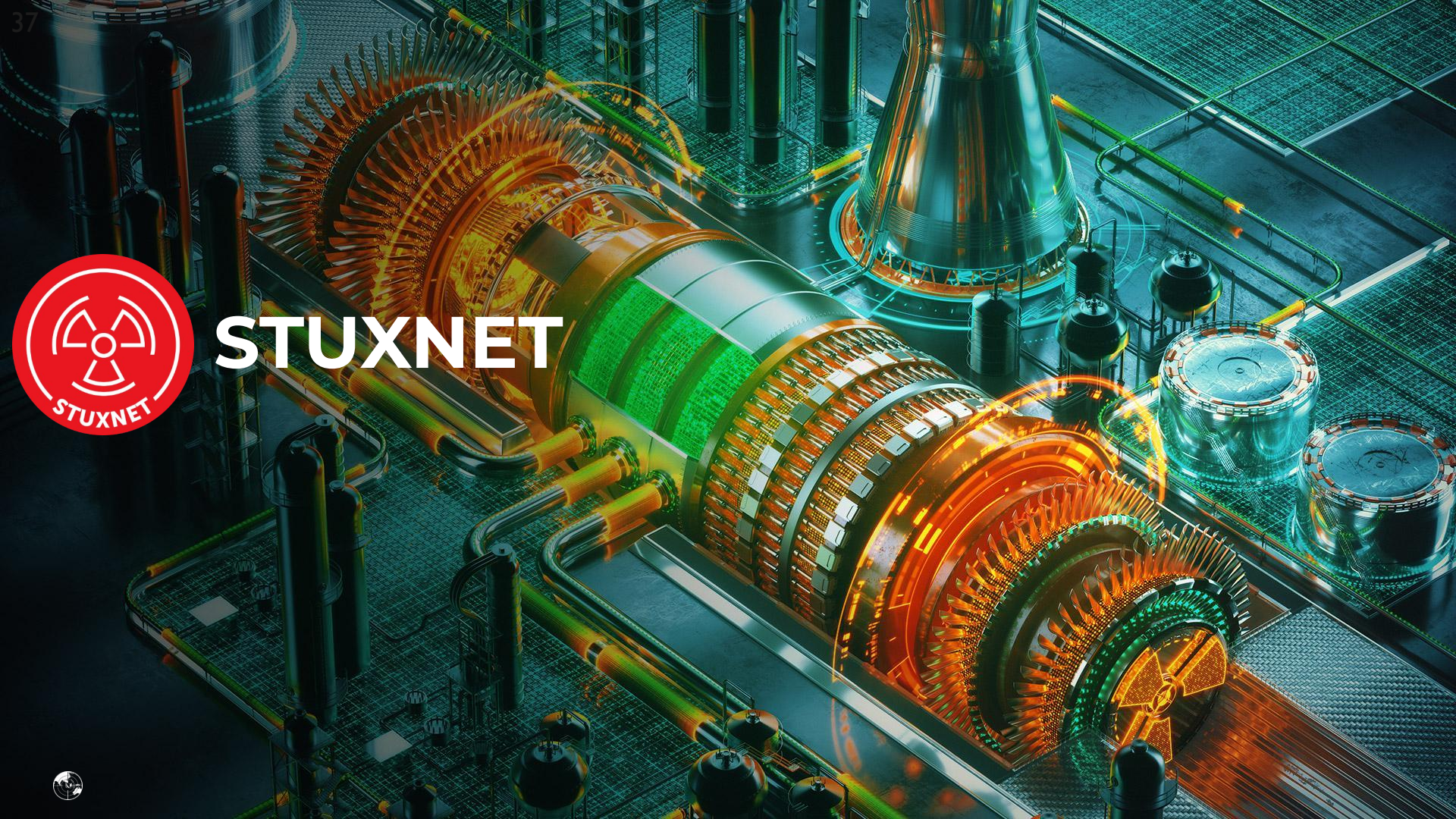
- APT groups
  - As name suggests, targeted, funded, advanced attacks
- FIN groups
  - Monetary-gain motivated
- Opportunistic, Commodity threats (except when it's used by an APT)
  - Ransomware
  - Bots / RATs
  - Stealers

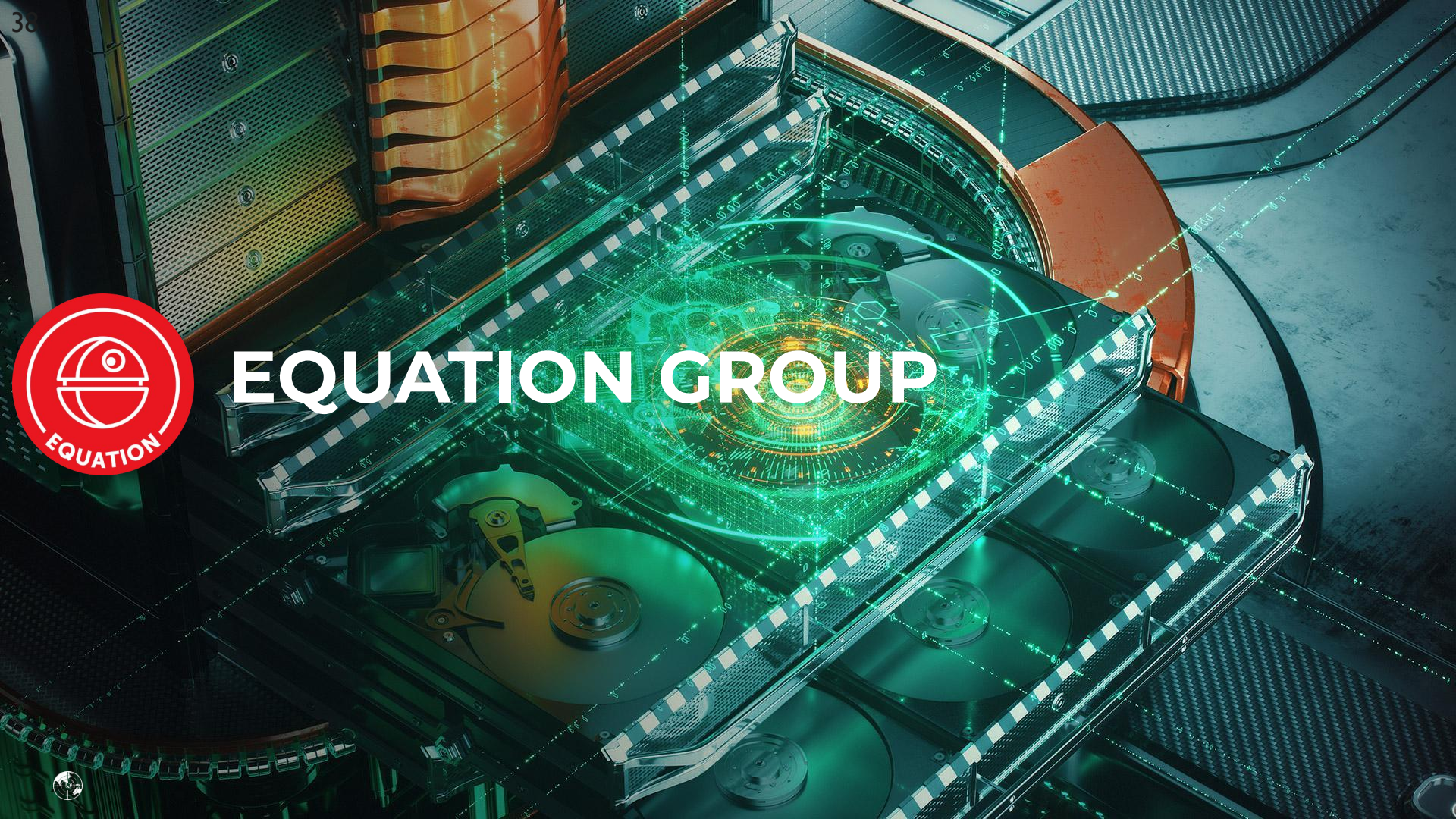
Advanced Threat groups



HIGHLIGHTS


2017
LAZARUS
2015
EQUATION
TURLA
2014
DARK HOTEL
2012
RED OCTOBER
2011
DUQU
2010
STUXNET
2008
GREAT


STUXNET


EQUATION GROUP


TURLA
SATELLITE TURLA

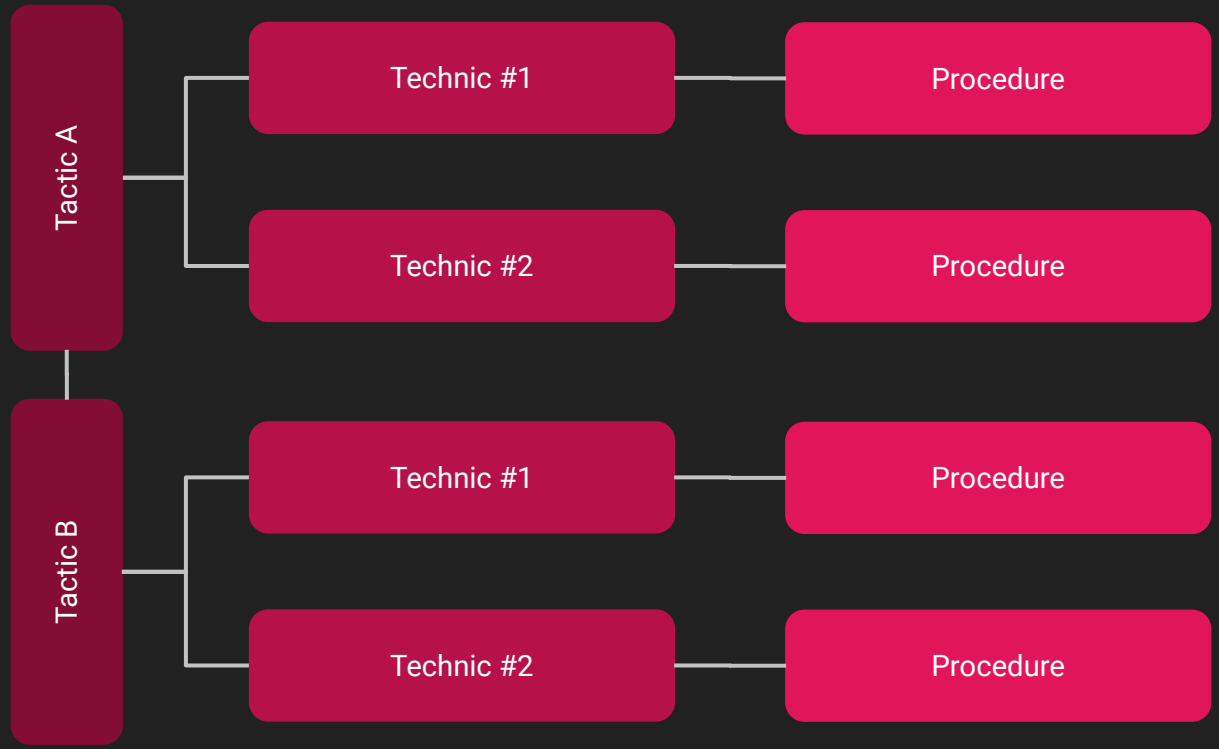
Cyberespionage in 2020

- Multi stage malware (with validators), increasingly sophisticated
- Scripting languages
- Living off the land
- Open-source tools

- Fileless malware
- Networking hardware malware
- Firmware malware

Threat group's imprint - TTPs

- Sets of Tactics, Techniques and Procedures that are associated with threat groups.
- Acts as attribution tokens when analysing an attack.



Sources - ATT&CK Mitre profiles

Sources - ATT&CK Mitre profiles:

OilRig

OilRig is a threat group with suspected Iranian origins that has targeted Middle Eastern and international victims since at least 2014. The group has targeted a variety of industries, including financial, government, energy, chemical, and telecommunications, and has largely focused its operations within the Middle East. It appears the group carries out supply chain attacks, leveraging the trust relationship between organizations to attack their primary targets. FireEye assesses that the group works on behalf of the Iranian government based on infrastructure details that contain references to Iran, use of Iranian infrastructure, and targeting that aligns with nation-state interests. [1] [2] [3] [4] [5] [6] This group was previously tracked under two distinct groups, APT34 and OilRig, but was combined due to additional reporting giving higher confidence about the overlap of the activity.

ID: G0049

Aliases: OilRig, Helix Kitten, APT34

Contributors: Robert Falcone, Bryan Lee

Version: 1.0

Alias Descriptions

Name	Description
OilRig	[1] [2] [3] [4] [5] [11]
Helix Kitten	[11]
APT34	This group was previously tracked under two distinct groups, APT34 and OilRig, but was combined due to additional reporting giving higher confidence about the overlap of the activity. [11] [6]

Techniques Used

Domain	ID	Name	Use
Enterprise	T1087	Account Discovery	OilRig has run `net user`, `net user /domain`, `net group "domain admins" /domain`, and `net group "Exchange Trusted Subsystem" /domain` to get account listings on a victim.[3]
Enterprise	T1119	Automated Collection	OilRig has used automated collection.[5]

Sources - ATT&CK Mitre profiles:

Techniques Used

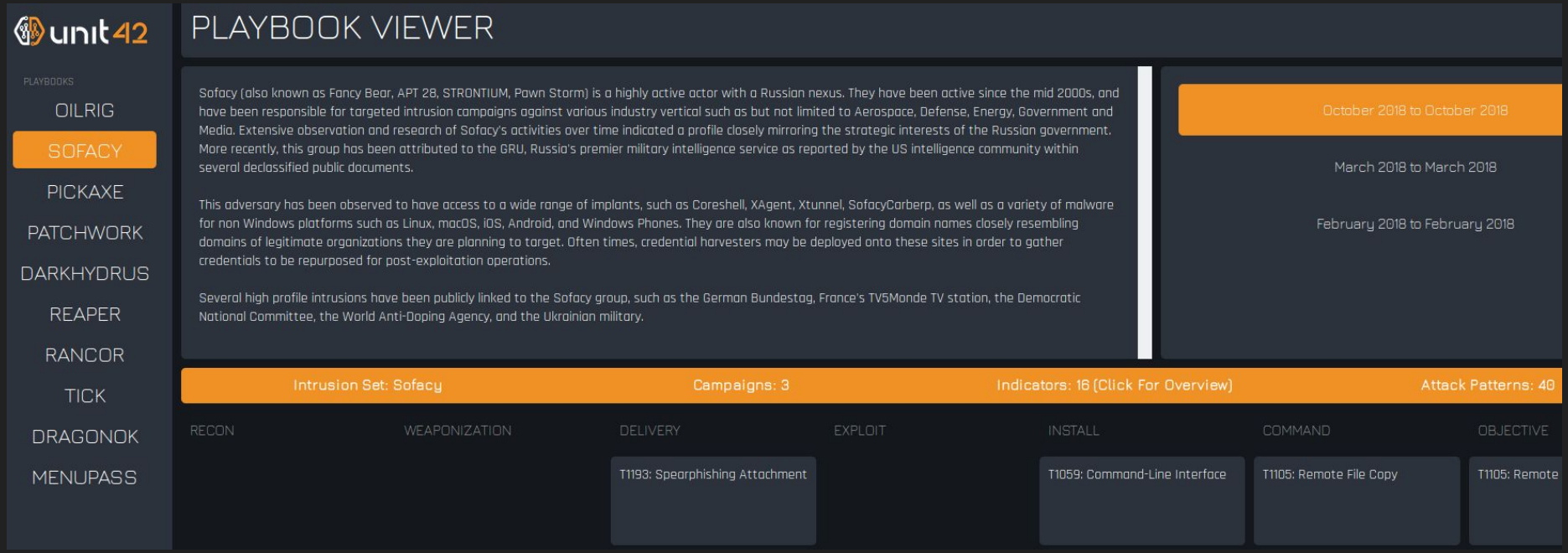
Domain	ID	Name	Use
Enterprise	T1087	Account Discovery	OilRig has run `net user`, `net user /domain`, `net group "domain admins" /domain`, and `net group "Exchange Trusted Subsystem" /domain` to get account listings on a victim.[3]
Enterprise	T1119	Automated Collection	OilRig has used automated collection.[5]
Enterprise	T1110	Brute Force	OilRig has used brute force techniques to obtain credentials.[7]
Enterprise	T1059	Command-Line Interface	OilRig has used the command-line interface for execution.[6][8][5][7]
Enterprise	T1223	Compiled HTML File	OilRig has used a CHM payload to load and execute another malicious file once delivered to a victim.[3]
Enterprise	T1003	Credential Dumping	OilRig has used credential dumping tools such as Mimikatz and Lazagne to steal credentials to accounts logged into the compromised system and to Outlook Web Access.[5][7]
Enterprise	T1094	Custom Command and Control Protocol	OilRig has used custom DNS Tunneling protocols for C2.[5]
Enterprise	T1140	Deobfuscate/Decode Files or Information	A OilRig macro has run a PowerShell command to decode file contents. OilRig has also used certutil to decode base64-encoded files on victims.[6][9][8]
Enterprise	T1048	Exfiltration Over Alternative Protocol	OilRig has exfiltrated data over FTP separately from its primary C2 channel over DNS.[4]
Enterprise	T1133	External Remote Services	OilRig uses remote services such as VPN, Citrix, or OWA to persist in an environment.[7]

Sources - ATT&CK Mitre profiles:

Software

ID	Name	Techniques
S0160	certutil	Deobfuscate/Decode Files or Information, Install Root Certificate, Remote File Copy
S0095	FTP	Commonly Used Port, Exfiltration Over Alternative Protocol
S0170	Helminth	Automated Collection, Clipboard Data, Code Signing, Command-Line Interface, Data Encoding, Data Staged, Data Transfer Size Limits, Input Capture, Obfuscated Files or Information, Permissi Discovery, PowerShell, Process Discovery, Registry Run Keys / Startup Folder, Remote File Copy, Scheduled Task, Scripting, Shortcut Modification, Standard Application Layer Protocol, Standar Cryptographic Protocol
S0100	ipconfig	System Network Configuration Discovery
S0189	ISMInjector	Deobfuscate/Decode Files or Information, Obfuscated Files or Information, Process Hollowing, Scheduled Task
S0002	Mimikatz	Account Manipulation, Credential Dumping, Credentials in Files, DCShadow, Pass the Hash, Pass the Ticket, Private Keys, Security Support Provider, SID-History Injection
S0039	Net	Account Discovery, Create Account, Network Share Connection Removal, Network Share Discovery, Password Policy Discovery, Permission Groups Discovery, Remote System Discovery, Service System Network Connections Discovery, System Service Discovery, System Time Discovery, Windows Admin Shares
S0104	netstat	System Network Connections Discovery
S0264	OopsIE	Command-Line Interface, Data Compressed, Data Encoding, Data Staged, Data Transfer Size Limits, Deobfuscate/Decode Files or Information, Exfiltration Over Command and Control Channel, Obfuscated Files or Information, Remote File Copy, Scheduled Task, Scripting, Security Software Discovery, Software Packing, Standard Application Layer Protocol, System Information Discov Time Discovery, Windows Management Instrumentation
S0184	POWRUNER	Account Discovery, Command-Line Interface, Data Encoding, File and Directory Discovery, Permission Groups Discovery, PowerShell, Process Discovery, Query Registry, Remote File Copy, Sche Screen Capture, Security Software Discovery, Standard Application Layer Protocol, System Information Discovery, System Network Configuration Discovery, System Network Connections Disc Owner/User Discovery, Windows Management Instrumentation
S0029	PsExec	Service Execution, Windows Admin Shares

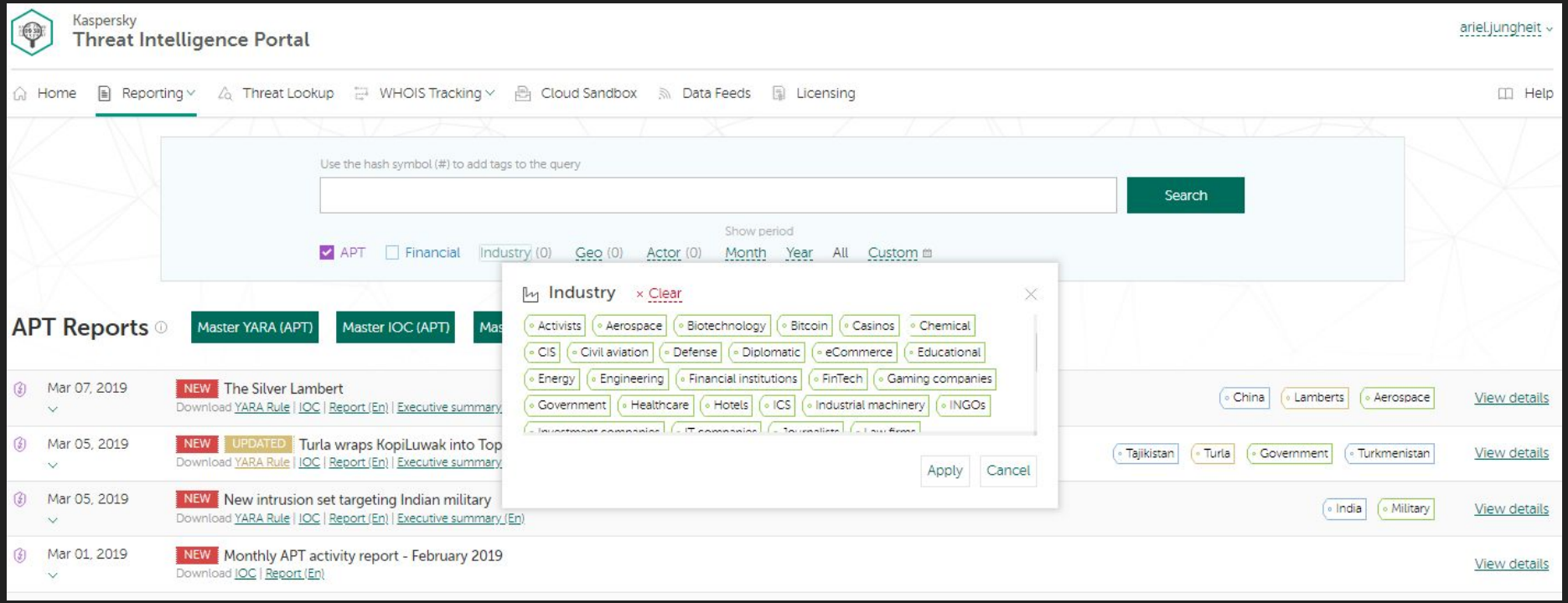
Sources – Palo Alto Unit42 Playbooks:



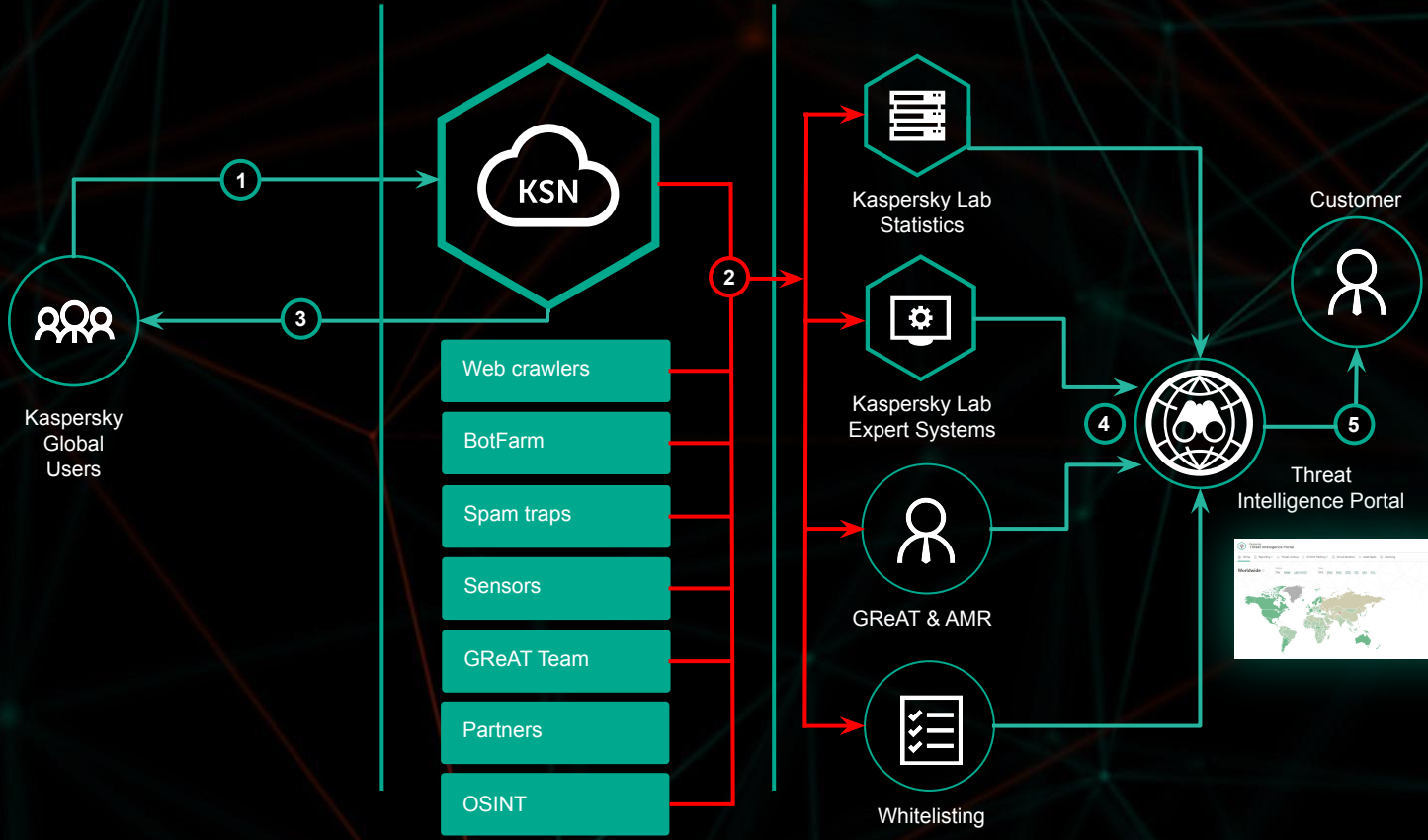
Sources - Threat Intel reports:

- Almost every major Cybersecurity company offer a Threat Intel subscription.
- The quality and content depends on the company's resources and visibility
- There is no standard, they greatly differ
  - Some are more technical
  - Some are more C-level oriented
  - Some could be just wrong
- Best reports caters to both Analysts and C-level employees
  - Analysts learn from and use the technical analysis
  - CISO draws conclusions from the executive summary

Sources - Threat Intel Platform:



TIP Architecture



Other Collection Sources

- Malware sharing platforms:
  - VirusShare
  - Malpedia
  - Malshare
  - Contagio
- Exchanges
  - Alienvault OTX
  - Facebook Threat Exchange
- Pastebin
- Open- and Closed-Forum Scraping

Putting Profiles/Reports to use

- Understand who is being targeted, why? Get context
- Extract IoCs:
  - Hashes
  - IP addresses
  - Domains
- If applicable, create detection:
  - Endpoint
  - Network
- Create Hunting rules, track the threat, anticipate and mitigate future risks

Putting Profiles/Reports to use

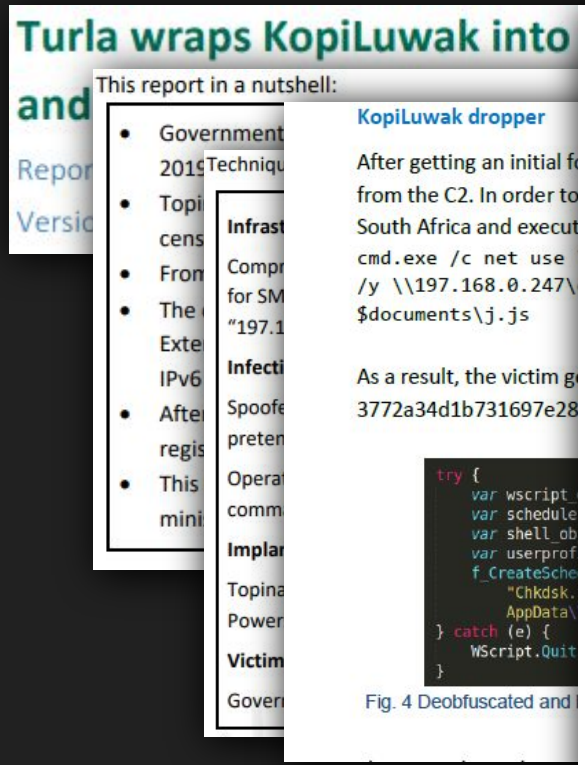


Fig. 4 Deobfuscated and

Appendix I – Indicators of Compromise

Note: *The indicators in this section are valid at the time of publication. Any future changes will be directly updated in the corresponding .ioc file.*

Files

```
%APPDATA%\Roaming\Microsoft\Chkdsk.js
%APPDATA%\Roaming\Microsoft\reportservice.js
%APPDATA%\Roaming\Microsoft\vpngate.js
%APPDATA%\Roaming\Microsoft\microsoftthemesexample.js
%APPDATA%\Microsoft\t235.dat
%LOCALAPPDATA%\VirtualStore\certcheck.exe
```

Registry

Server and port config

```
HKCU\Software\Microsoft\Notepad\lfStrikeOut
```

Payload

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Maintenance\MOfficeMaintenance
HKCU\Software\Microsoft\Windows\CurrentVersion\SoftEther\VPN UI Helper
HKCU\Software\Microsoft\Windows\CurrentVersion\SkydriveSettings\ControllerSettings
HKCU\Software\Microsoft\Windows\\CurrentVersion\ReportSettings\Office16
```

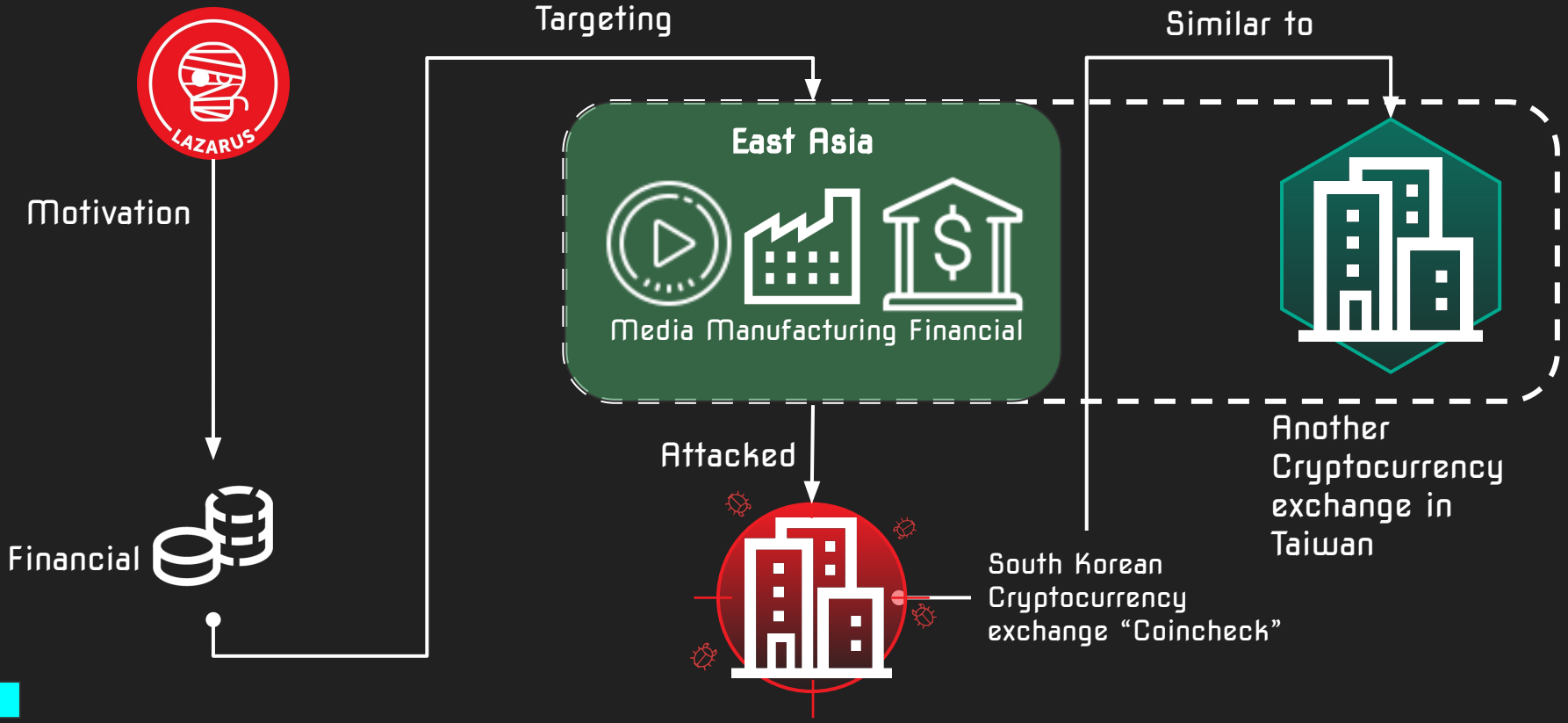
Scheduler task name

```
VerifiedPublisherCertCheck
ProactiveScan
```

WordPress legit compromised sites .php URL templates

```
http://txflasher[.]com/wp-includes/Requests/Socks.php
```

Profiling your company / Victim perspective



Some tools

- Signatures:
  - Yara rules
  - IoCs
  - Network signatures
  - SIEM – Sigma rules
- Malware analysis:
  - Depending on the artifact in question
- Hunting:
  - VirusTotal
  - KLARA

So let’s start it going with the most useful tool we have

YARA starts here

[CONTROL SLIDE]

Yara

Yara - introduction

YARA in a nutshell

YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples. With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns. Each description, a.k.a rule, consists of a set of strings and a boolean expression which determine its logic. Let's see an example:

https://github.com/VirusTotal/yara

Latest version: 4.0.5 (Feb 2021)
Precompiled Windows binaries; source code for Linux and Mac

Documentation: yara.readthedocs.org/en/latest/writingrules.html

Pro-tips

- Yara reference
  - https://yara.readthedocs.io/en/v4.0.5/
- File format memos (by Ange Albertini aka corkami):
  - https://github.com/corkami/pics
- Yara for VirusTotal:
  - https://www.virustotal.com/en/documentation/searching/

What can we do with Yara?

- identify and classify malware
- find new malware samples based on family-specific features
- find new exploits and zero-days
- help speeding up incident response
- increase your defenses by deploying custom rules inside your organization
- classification: identify file formats, archives, packed files, known threats
- filter network traffic (why not)
- build your own private antivirus :-)

```
rule hello_world {
meta:
    description = “My first yara!”

strings:

    $a=”Make Yara GReAT Again!”

condition:
    $a
}
```

Yara rules basic “design” tips

Design tips

- Think of Yara like a programming language, not enhanced regexp's
- Indent the code, comment it nicely
- Quick "hack" rules can be okay now, but what will happen in 1-2 years when you need to reuse it?
- Other people might rely on your rule
- Some rules might get published. You may want to include contact info.
  - Name, e-mail address

The rules of YARA club

- There are no rules for rules
- There is no wrong or right rule
- Every rule is good enough **if it *works***
- We only show how ***We do it***,
  you are free to choose your way

*Note: if this is your first time in YARA club you have to make a kick-ass rule!

Better: use unique and specific (data) strings

- use the Unix strings tool to extract ASCII text strings*
  *to extract Unicode strings use "-e l" option

- look at the results manually (+Google):
  - mutex or event names (eg. “WerTyQ34C”)
  - rare user-agents (eg. “NOKIAN95”)
  - registry key / unique value names
  - typos (eg. SOFTWAE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN\)
  - PDB paths (eg. C:\Users\client7\Desktop\chforce\Release\chforce.pdb)
  - Unique GUID’s
  - internal module names (“MyMalwareKeyloggerLoader.dll”)
  - encoded or encrypted configuration strings

Better: use different conditions

- when possible, use 2-3 groups of conditions:
    - based on unique artefacts found in the malware
    - based on a group of specific non-unique strings
    - based on file properties and structure (sections, entropy, timestamp etc)

Case study #1

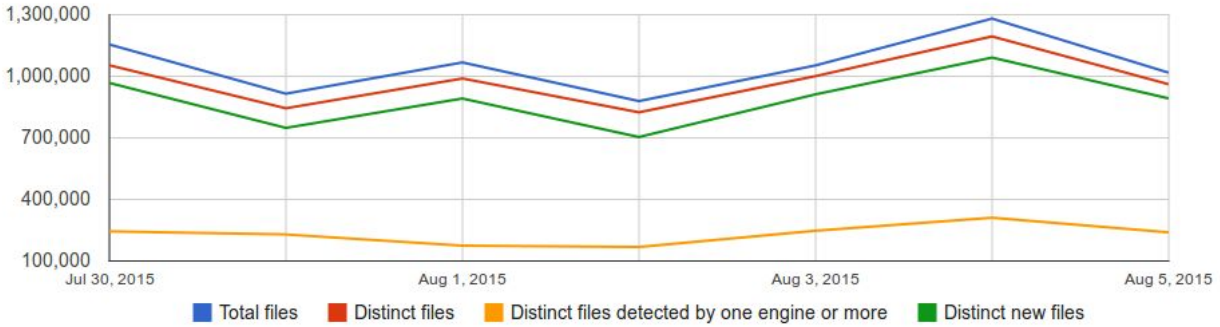
```
rule apt_CN_BlueTraveller {

strings:

    $a1 = "PROXY_PROXY_PROXY_PROXY"  fullword ascii
    $a2 = "0ROXY_TYPE" fullword ascii

    $b1 = "cmd.exe /c hostname" fullword ascii
    $b2 = "/O.htm" fullword ascii
    $b3 = "%s%04d/%s" fullword ascii
    $b4 = "http://%s/%s/%s/" fullword ascii

    $c1 = "cmd.exe /c" fullword ascii
    $c2 = "Upload failed..." fullword ascii
    $c3 = "Download OK!" fullword ascii
    $c4 = "-download" fullword ascii
    $c5 = "-exit" fullword ascii
    $c6 = "james" fullword ascii

condition:
    uint16(0) == 0x5A4D and
    (any of ($a*) or 2 of ($b*) or 4 of ($c*))
    and filesize < 400000
}
```

Virus Total

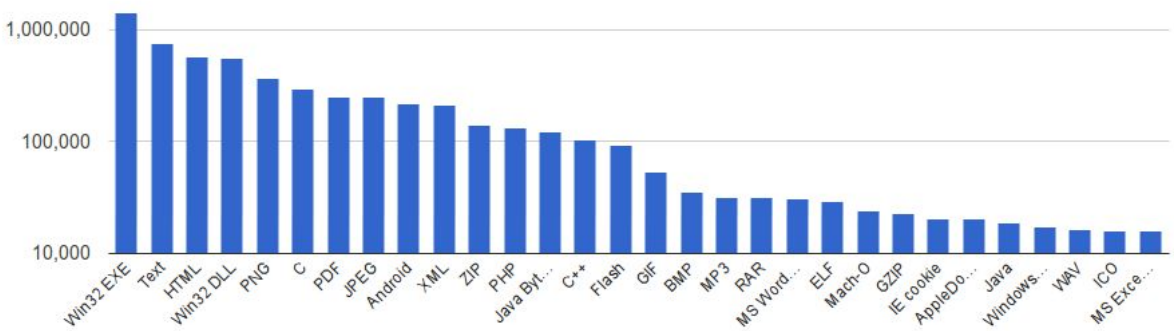
VirusTotal / VTI

- VirusTotal is a custom multiscanner originally created by Bernardo Quintero and Julio Canto, from Hispasec
- Acquired by Google
- Based on commercial subscription model
  - Multiple levels
- Even if you don't have an account, you might get one at some point
- Participation in certain private groups will grant you a VT account
- It's an excellent source of new malware
- It runs Yara rules on all incoming samples with extended Yara syntax
- It automatically unpacks all the samples, you can even search in embedded macroses code for example

Submissions



File types



Regular search

- patterns, specific for known APT groups / malware families
- generic rules for
  - trojans
  - rootkits
  - downloaders
  - keyloggers
  - backdoors
  - etc

Useful VTI Yara tags

- new_file
- tags contains "via-tor"
- positives > 5
- not signatures contains "Adware"
- submissions < 4
- not tags contains "nsrl" and not tags contains "trusted"
- not tags contains "corrupt"
- and not kaspersky

```
rule new_dll_file {

condition:

      new_file     // new sample for VirusTotal

      and positives > 1 // at least one detect

      and positives < 10  //  detected by less
than 3 AV

      and not Kaspersky       //  not by Kaspersky
      AV

      and file_type contains “PEDLL“

      and filesize < 1M

}
```

```
rule new_debug_dll_via_TOR {

strings:

      $pdb = ".pdb"

condition:

      $pdb  and new_file

      and tags contains "via-tor"

      and tags contains "overlay"

      and not tags contains "nsrl"

      and not tags contains "trusted"

      and not tags contains "corrupt"

      and file_type contains "PEDLL"

}
```

```
rule susp_Chinese_DynDNS {

strings:

      $a1 = ".eicp.net"

      $a2 = ".f3322.org"

      $a3 = ".meibu.net"

      $a4 = ".vicp.cc"

condition:

      uint16(0) == 0x5A4D and any of ($a*)

      and not tags contains "trusted"

      and not tags contains "corrupt"

      and filesize > 10000 and filesize < 1000000

}
```

```
rule susp_WordDoc_connect_Net {

strings:

      $dochdr = {D0 CF 11 E0}

      $a1 = "URLDownloadToFile" nocase

      $a2 = "ShellExecute" nocase

      $a3 = "DeleteUrlCacheEntry" nocase

condition:

      $dochdr at 0

      and all of ($a*)

      and not kaspersky

      and filesize < 1000000

}
```

Windows commands used for lateral movement

Appendix A: List of Executed Commands by respective Attack Groups (Attack Group A)

Table 4: Initial Investigation (Attack Group A)

Ranking	Command	Times executed	Option
1	tasklist	119	/s /v
2	ver	92	
3	ipconfig	58	/all
4	net time	30	
5	systeminfo	24	
6	netstat	22	-ano
7	qprocess	15	
8	query	14	user
9	whoami	14	/all
10	net start	10	
11	nslookup	4	
12	fsutil	3	fsinfo drives
13	time	2	/t
14	set	1	

Table 3: Spread of Infection

Ranking	Command	Times executed
1	at	103
2	reg	31
3	wmic	24
4	wusa	7
5	netsh advfirewall	4
6	sc	4
7	rundll32	2

*"wmic" is also used for reconnaissance.

JPCERT CC® Official Blog

Japan Computer Emergency Response Team Coordination Center

```
rule susp_WinRAR_DownloadToFile {

strings:

     $a1 = "WinRAR\\shell\\open\\command"
fullword nocase ascii

     $a2 = "URLDownloadToFile" nocase ascii

condition:

     uint16(0) == 0x5A4D

     and all of ($a*)

     and not kaspersky

     and filesize < 1000000

}
```

```
rule susp_localIP {

strings:

     $a1 = “cmd /c " nocase ascii wide

     $a2 = "SELECT * FROM AntiVirusProduct"
nocase ascii wide

     $a3 = “recycle.bin\\" nocase

     $b1 = "192.168." nocase ascii wide

condition:

     uint16(0) == 0x5A4D

     and any of ($a*) and $b1

}
```

Search of anomalies

- Fake timestamp
- Fake signature
- Connections to known bad Dynamic DNS domains
- Mimics legitimate files (Adobe or MS or Skype etc)
  - Using legit PE info but not signed/trusted
  - Using legit resources (Skype icon for example)
  - Using "legit" metadata and wrong entropy / code page / exports
- By lateral movement patterns
- By crypto signatures
- By atypical actions for this file type

```
rule susp_filesize_svchost_exe {

meta:

      description = "Detects suspicious
svchost.exe file size"

condition:

      uint16(0) == 0x5A4D
      and filename == "svchost.exe"
      and ( filesize > 100KB )
}
```

```
rule susp_filesize_winsta_dll {

meta:

      description = "Detects suspicious
winsta.dll file size"

condition:

      uint16(0) == 0x5A4D
      and filename == "winsta.dll"
      and ( filesize < 50KB or filesize > 1M )
}
```

Demo - VTI

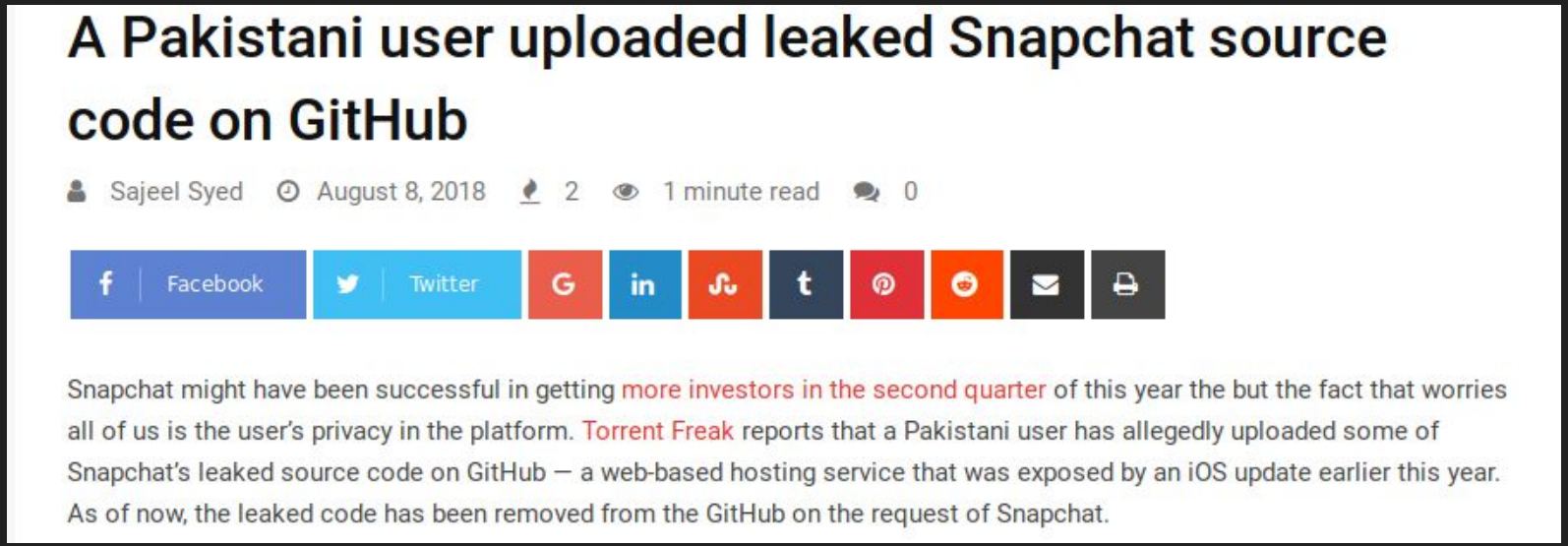
- Looking for a PE from Romania, submitted via VT web interface in the last month, submitted within 5 minutes of compilation:
  - type:peexe subspan:300- submitter:RO submitter:web fs:2021-02-01+ fs:2021-03-29-
- Doc abusing macro to run a file:
  - type:doc tag:macros tag:run-file
- WinRAR 19 Year ACE relative path RCE Vulnerability:
  - cve-2018-20250

Interesting VT dorks

- Malware hosted on Governmental site:
  - itw:".gov" positives:5+
- Recent sample, signed with some detections:
  - ls:2021-04-05+ type:dmg positives:2+ tag:signed
- Enticing user to run macros:
  - content:"click enable editing"
  - content:"click enable content"

VirusTotal Hunting - example

- Taking recent case from the news - Snapchat code leak:



A Pakistani user uploaded leaked Snapchat source code on GitHub

Sajeel Syed August 8, 2018 2 1 minute read 0

Facebook Twitter

Snapchat might have been successful in getting more investors in the second quarter of this year the but the fact that worries all of us is the user's privacy in the platform. Torrent Freak reports that a Pakistani user has allegedly uploaded some of Snapchat's leaked source code on GitHub — a web-based hosting service that was exposed by an iOS update earlier this year. As of now, the leaked code has been removed from the GitHub on the request of Snapchat.

VirusTotal Hunting - example

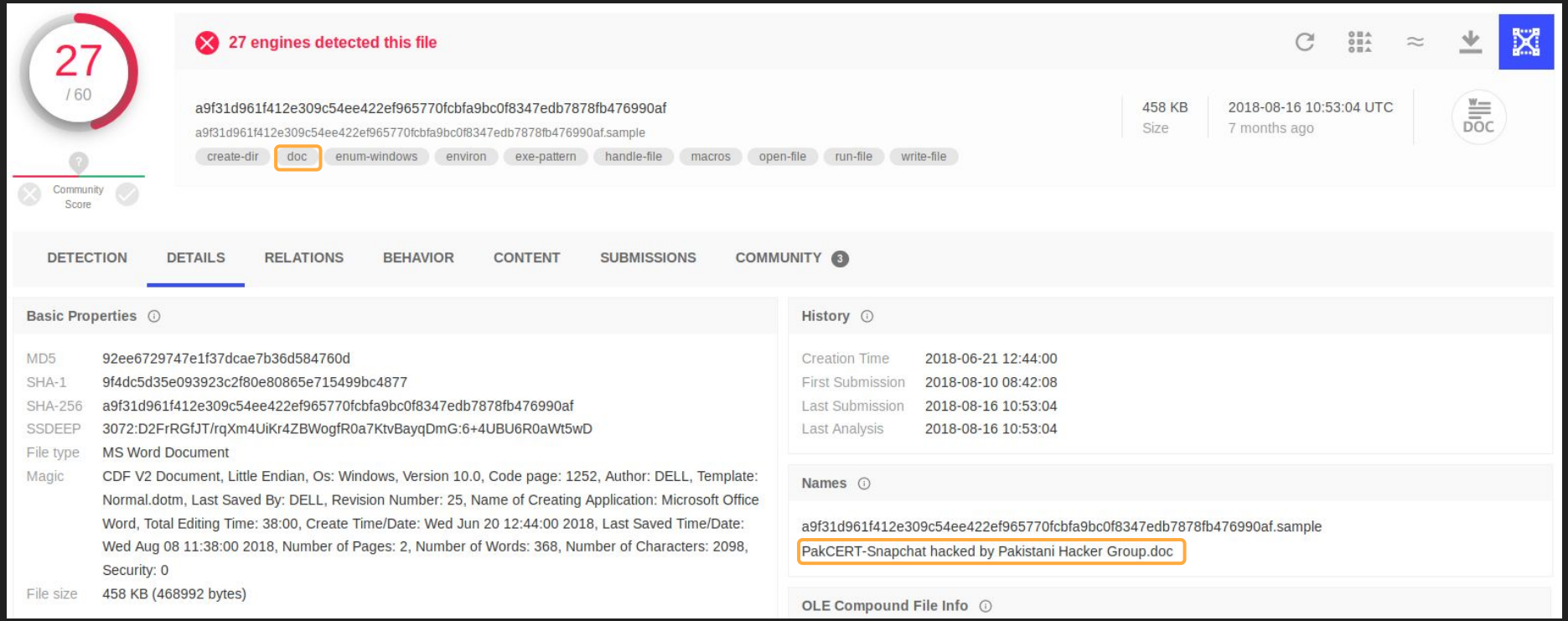
```
condition:
      (
      file_type contains "msoffice" or
      file_type contains "rtf" or
      file_type contains "document" or
      file_type contains "internet email"
      ) and
      positives > 1 and
      submissions < 2 and
      (file_name matches Snapchat or
      file_name matches Pakistan or
      file_name matches Pakistani or
      file_name matches Salman or
       file_name matches Mohammed)
```

Hunting for a malicious document

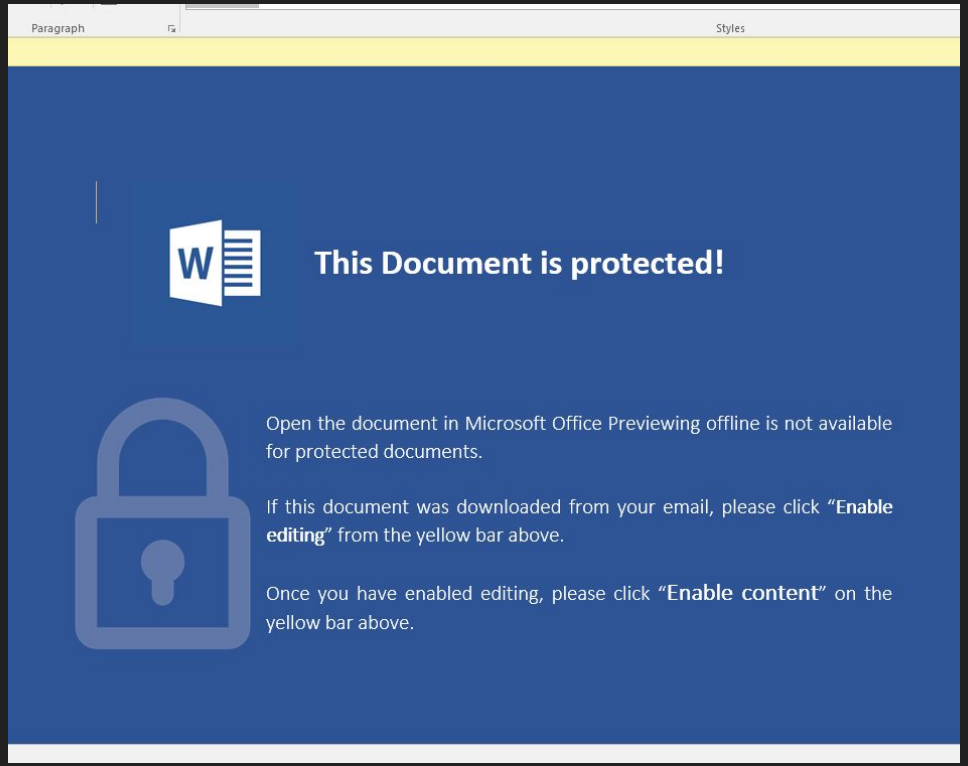
Unique, not yet discovered

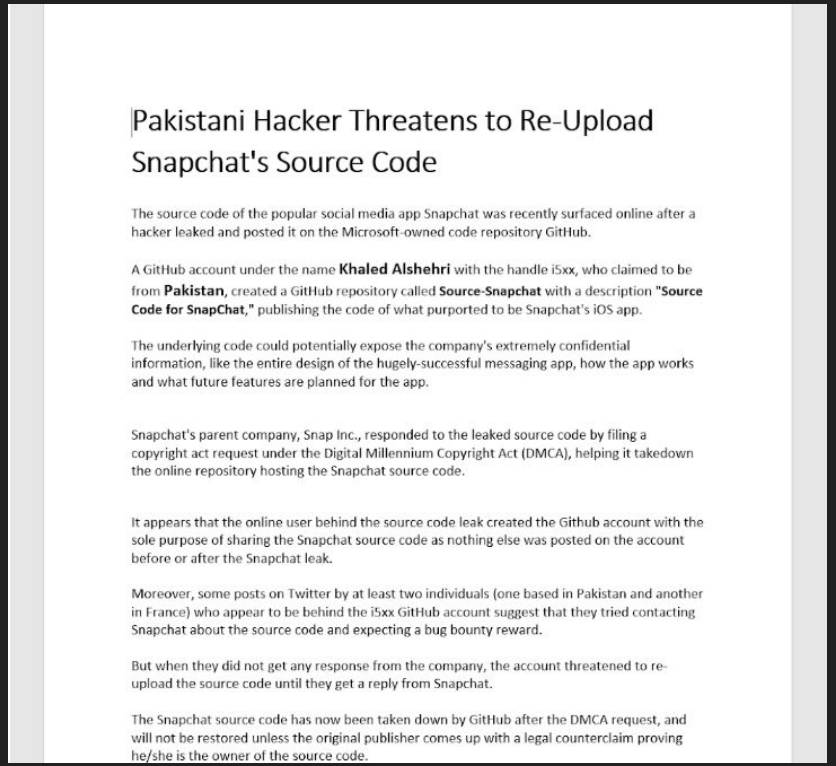
Relating to the subject matter

VirusTotal Hunting - example



VirusTotal Hunting - example





Pakistani Hacker Threatens to Re-Upload Snapchat's Source Code

The source code of the popular social media app Snapchat was recently surfaced online after a hacker leaked and posted it on the Microsoft-owned code repository GitHub.

A GitHub account under the name **Khaled Alshehri** with the handle i5xx, who claimed to be from **Pakistan**, created a GitHub repository called **Source-Snapchat** with a description **"Source Code for SnapChat,"** publishing the code of what purported to be Snapchat's iOS app.

The underlying code could potentially expose the company's extremely confidential information, like the entire design of the hugely-successful messaging app, how the app works and what future features are planned for the app.

Snapchat's parent company, Snap Inc., responded to the leaked source code by filing a copyright act request under the Digital Millennium Copyright Act (DMCA), helping it takedown the online repository hosting the Snapchat source code.

It appears that the online user behind the source code leak created the Github account with the sole purpose of sharing the Snapchat source code as nothing else was posted on the account before or after the Snapchat leak.

Moreover, some posts on Twitter by at least two individuals (one based in Pakistan and another in France) who appear to be behind the i5xx GitHub account suggest that they tried contacting Snapchat about the source code and expecting a bug bounty reward.

But when they did not get any response from the company, the account threatened to re-upload the source code until they get a reply from Snapchat.

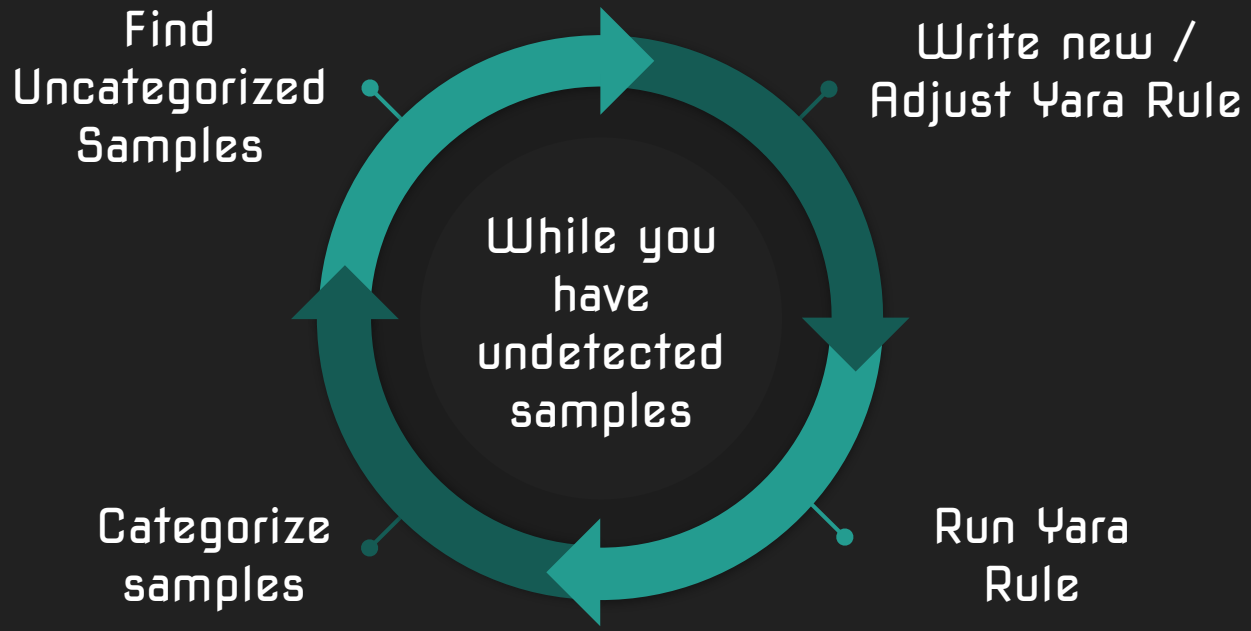
The Snapchat source code has now been taken down by GitHub after the DMCA request, and will not be restored unless the original publisher comes up with a legal counterclaim proving he/she is the owner of the source code.

Wannacry rule

```
rule ransomware_WannaCry_code {
meta:

        description = "Rule for WannaCry code, also matches other Lazarus"
        description = "campaigns: BlueNoroff, ManusCrypt, Decafett"
        hash = "808182340FB1B0B0B301C998E855A7C8"
        hash = "B9B3965D1B218C63CD317AC33EDCB942"
        author = "alice@kaspersky.com"

strings:

        $c1 = {5424FC740CC74424FC00000000015424}
        $c2 = {397424FC740CC74424FC000000000174}
        $c3 = {C74424FC00000000016C24FC83EC0439}
        $c4 = {5C24FC740CC74424FC00000000015C24}
        $c5 = {396C24FC740CC74424FC00000000016C}

condition:

        uint16(0) == 0x5A4D and filesize < 2000000 and all of them
}
```

Using yara to cluster malware

Dealing with large collections - Methodology



IOCs and practical scenarios (1)

[CONTROL SLIDE]

IOCs and their use in real scenarios

A few words about IOCs

- It is a good idea to set policies when dealing with IOCs' lifecycle:
  - Separate Hunting from Detection collections
  - Agree on expiration dates.
  - IoC source quality and trust level.

- Different formats, using a framework (like MISP) might help ingesting them

- Please keep in mind that quantity != quality. Actually too many "useless" IOCs might kill your systems

- Also keep in mind that lack of context might be a problem

IOCs - different formats

Let's take a quick look to different formats and what they provide both in theory and reality:

https://otx.alienvault.com/pulse/5b1850fdca86010c957ca78b?utm_medium=InProduct&utm_source=OTX&utm_content=Email&utm_campaign=new_pulse_from_subscribed

For ingestion of large amounts of information we need both aggregators and frameworks:

https://github.com/PaloAltoNetworks/minemeld/wiki

A few words about SIGMA rules

- Generic Signature Format for SIEM Systems.
  - New standardized format used to leverage SIEM logs for threat detection.

- Different from Yara/IoCs in that it describes a situation, not a file or network capture. e.g. Baby Shark activity:
  - category: process_creation
  - CommandLine: powershell.exe mshta.exe http*
  - CommandLine: cmd.exe /c taskkill /im cmd.exe

- Rules are at: https://github.com/Neo23x0/sigma/tree/master/rules

Dis the point of no return..

Some of the tools we use internally

LiquidOnyx

AptDB

KLara

CyberChef

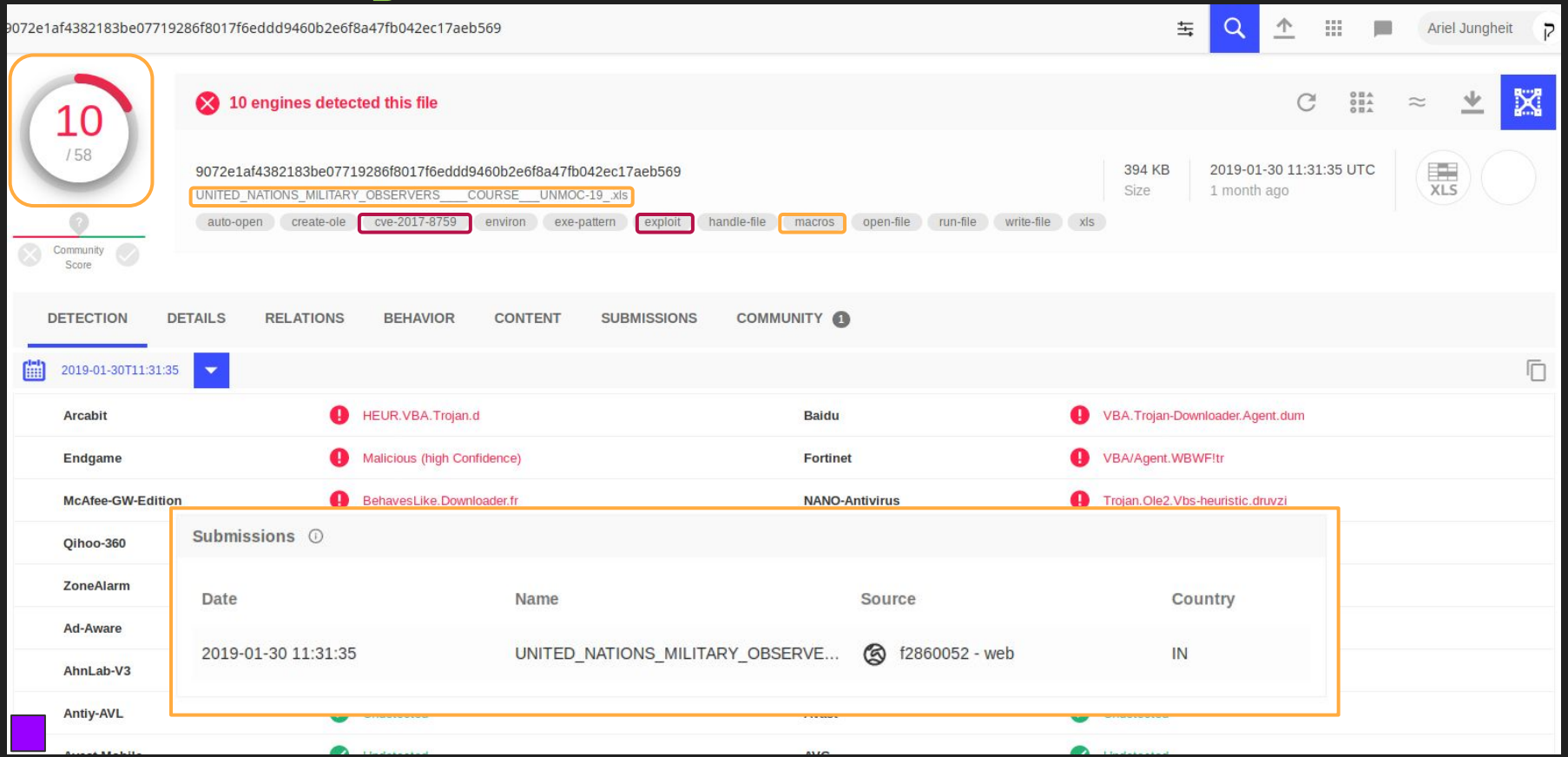
Hunting - practical scenarios

Threat Analysis Exercise

Using a VTI yara hunting rule we came across an interesting doc:

7fa6689ec0a8863e5084d30de4b9b252

Threat Analysis Exercise



Threat Analysis Exercise

Running olevba we see that the file contains heavily-obfuscated macro.

Trying to run this Excel document in a Sandbox we see it drops a file (hadram.exe 801f94bedb9481fb65709457c1f4c47a) which in turn drops another file (ekeoil.exe ab68db5c97f9ee12ca29c1eed881781d).

Extracting the IoCs from the sandbox report we have some artifacts to investigate, but let's try to take a look at the document and binary ourselves and see what else can we learn.

Threat Analysis Exercise - Context

Decoy document name in-the-wild is: UNITED_NATIONS_MILITARY_OBSERVERS____COURSE___UNMOC-19_.xls

Masquerades a “CENTRE FOR UNITED NATIONS PEACEKEEPING” (CUNPK) document.

Has only one submission at 30-01-2019 (from India). Creation time said to be 03-01-2019.

Threat Analysis Exercise - Malware

Hadram.exe:
.NET Stager version 2.2.2.4
PDB Path E:\updates\3\lioeek\lioeek\obj\Release\lioeek.pdb
File comment “lioeek is a application belongs to simbaa corporation(c)”
Compilation Timestamp 2019-01-03 08:20:33
Binary Copyright “Copyright © 2019”
File Size 290.5 KB
Checks for internet by contacting google with the UA “google/dance”
Takes care of extracting and executing ekeoil.exe

ekeoil.exe:
.NET RAT version 3.1.15
PDB Path E:\updates\ekeoil\ekeoil\obj\Release\ekeoil.pdb
Compilation Timestamp 2019-01-03 08:03:32
Binary Copyright "Copyright © 2018”
File Size 38.5 KB
UA “ekeoil/3.1.1.5”

Threat Analysis Exercise - Infrastructure

URLs:
http://firebasebox.com/tootie292/reboshw/c0_nCussi0N.php
http://firebasebox.com/tootie292/reboshw/1Inter-view_Call.php
http://firebasebox.com/tootie292/reboshw/iLIn_Ess_is_Ok.php
http://firebasebox.com/tootie292/reboshw/Sec_urit7Y-d3etaiLs.php
www.quora.com/If-programming-languages-had-honest-slogans-what-would-they-be

C2:
Firebasebox.com first seen 03-01-2019

IP:
157.230.112.219 Digital Ocean

Threat Analysis Exercise - Hunting for more

What do we have so far?

- Possibly targeting Indian individual
- Internet connectivity check via a Quora post
- Macro hidden in Form
- Funny looking .NET RAT

No much info from infrastructure analysis

...We need more samples

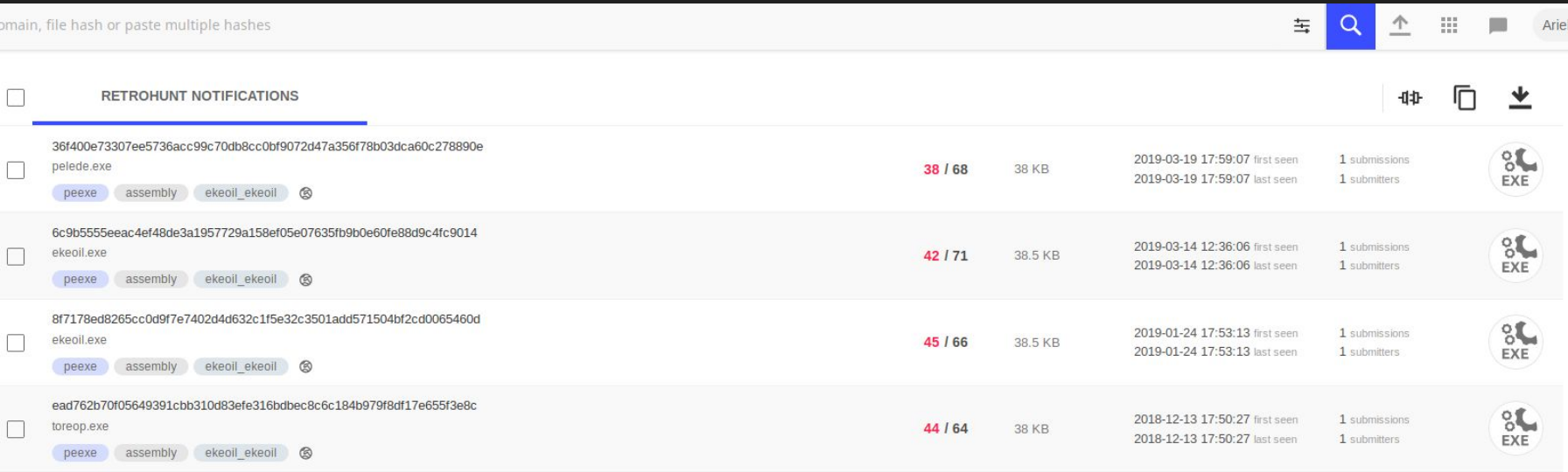
Let's make a yara rule!

Threat Analysis Exercise - Hunting for more

1. python ~/Tools/yarGen/yarGen.py --score --opcodes -m ekeoil/
2. Tune the rule
3. Retrohunt in VTI
4. ????
5. PROFIT!

Threat Analysis Exercise – Hunting for more

We found 4 files:



Commonalities: Imphash, Signature Copyright “Copyright © 2018”

Threat Analysis Exercise - Analyzing result

Looking at the oldest sample - toreop.exe:

.NET Binary

PDB Path E:\icloudi\pelede\pelede\obj\Release\toreop.pdb

Version 3.1.14

Compilation Timestamp 2018-11-26 09:08:46

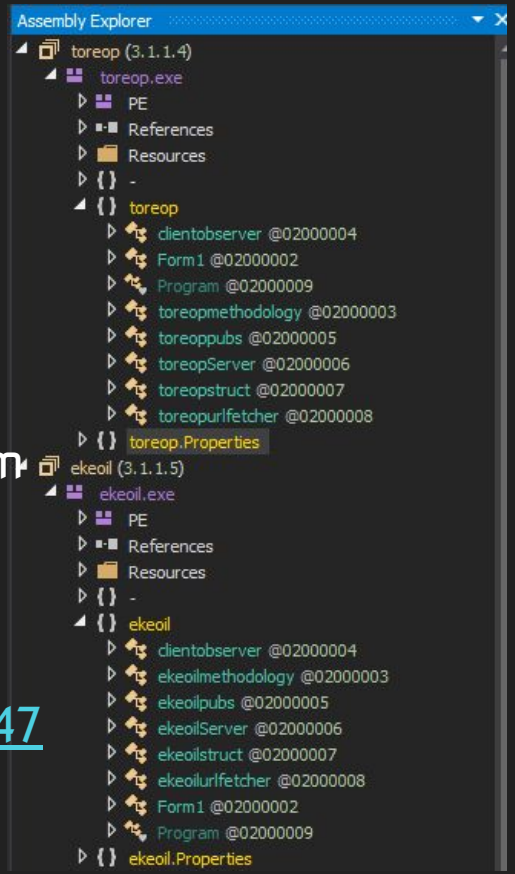
C2: http://ulsdriver[.]com/rotten393/tind_ers/ulsdriver.com

...Sec_urit7Y-d3etaiLs.php

...1Inter-view_Call.php

...iLIn_Ess_is_Ok.php

https://twitter.com/James_inthe_box/status/1078997063470174208



Threat Analysis Exercise – Digging deeper

- Looking at ulsdriver[.]com, we found ulsdriver[.]com/def/mauria.exe.
- mauria.exe is highly similar to samarit.exe (Peppy RAT), C2 is cloudserve[.]online.

- We also found Peppy RAT on firebasebox[.]com/def/Axess.exe, C2 is: ebluemedia[.]net, resolves to 173.212.234.57.

- 173.212.234.57 reverse-resolves to bdrive[.]club, cloudserve[.]club, cloudserve[.]online, dcloudsync[.]com.

- And on it goes..

Threat Analysis Exercise - Visualizing

cloudserve[.]online

samarit.exe

mauria.exe

ulsdriver[.]com

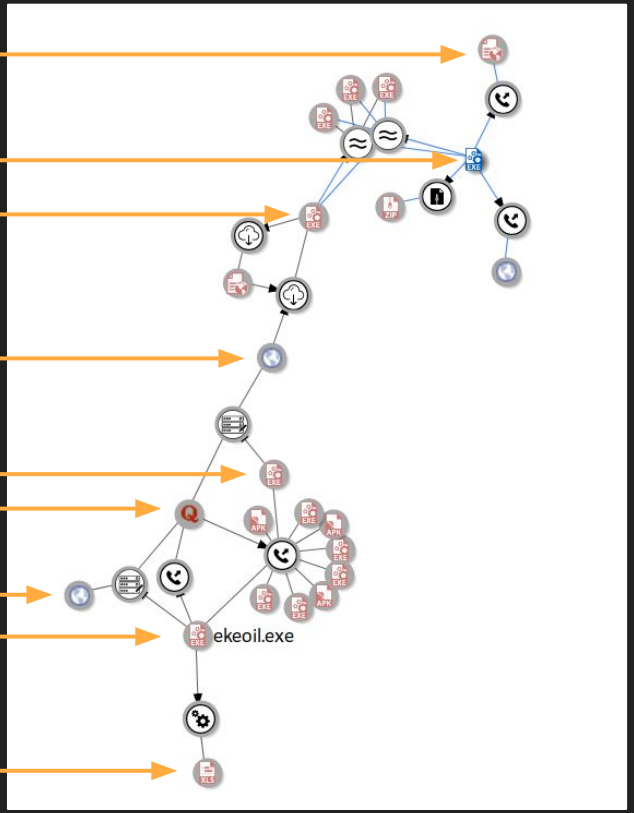
toreop.exe

www.quora.com/If-programming-languages-had-honest-slogans-what-would-they-be

firebasebox[.]com

ekeoil.exe

UNITED_NATIONS_MILITARY_OBSERVERS____COURSE___UNMOC-19_.xls



TIMELINE

Threat Analysis Exercise - Attribution work

- Peppy RAT
https://www.proofpoint.com/sites/default/files/proofpoint-operation-transparent-tribe-threat-insight-en.pdf

- Malicious doc had payload embedded within a userform, decimal encoded
- Macro checks for OS version
https://unit42.paloaltonetworks.com/unit42-tracking-subaat-targeted-phishing-attacks-point-leader-threat-actors-repository/

- Indian military related lure document
https://blog.trendmicro.com/trendlabs-security-intelligence/indian-military-personnel-targeted-by-information-theft-campaign/

Questions and answers

Local data won´t be enough – we need more sources (discussed later)

Perfect correlation with artifacts usually never happens: they can be easily changed

Attackers use generic tools to avoid attribution

Or they simply use false flags (Olympic Destroyer)

Advantages of attribution in an early stage

Knowing the attacker might be the whole objective of the investigation

How to react at a “political” and technical level

Hint about what could be the real objective of the breach

Risk assessment regarding how worrisome the attack might be

Big accelerator regarding how to continue the investigation at a forensic level

Context

Indicators to continue pivoting

Attribution problem #0: The naming scheme chaos

docs.google.com/spreadsheets/u/1/d/1H9_xaxQHpWaa4O_Son4Gx0YOIzlcBWMsdvePFX68

APT Groups and Operations

README | China | Russia | North Korea | Iran | Israel | NATO | Middle East | Others | Unknown | _Download

China

Common Name	CrowdStrike	IRL	Kaspersky	Secureworks	Mandiant	FireEye
Comment Crew	Comment Panda	PLA Unit 61398		TG-8223	APT 1	
APT 2	Putter Panda	PLA Unit 61486		TG-6952	APT 2	
UPS	Gothic Panda			TG-0110	APT 3	
IXESHE	Numbered Panda			TG-2754 (tentative	APT 12	BeeBus
APT 16					APT 16	
Hidden Lynx	Aurora Panda				APT 17	Deputy Dog
Wekby	Dynamite Panda	PLA Navy		TG-0416	APT 18	
Axiom					APT 17	
Winnti Group	Wicked Panda					
Shell Crew	Deep Panda		WebMasters		APT 19	KungFu Kittens
Naikon	Lotus Panda	PLA Unit 78020	Naikon		APT 30	
PLATINUM						
Lotus Blossom			Spring Dragon			

https://docs.google.com/spreadsheets/d/1H9_xaxQHpWaa4O_Son4Gx0YOIzlcBWMsdvePFX68EKU/edit#gid=361554658

https://resecurity.com/blog/supply-chain-the-major-target-of-cyberespionage-groups/

Attribution problem #1: Killing IOCs

All artifacts and infrastructure unique per victim; (i.e. project Sauron)

In-memory artifacts; won´t show up in typical collections;

Modular: modules alone won´t be that suspicous;

Implants tailor their metadata to the environment (timestamps);

Master of pattern escape (also in infrastructure)

No visibility without IR (both KL and Symantec analysis started this way)

What can we use for attribution other than indicators?

SECURELIST THREATS ▾ CATEGORIES ▾

Lazarus Under The Hood

By GReAT on April 3, 2017. 5:57 pm

LATEST OPSEC FAILURE

From the server logs of a C2 in Europe:

2017-01-18 02:54: Apache Tomcat started on port 8080
2017-01-18 04:10: HTTP GET view.jsp (via VPN in France)
2017-01-18 04:10: Testing bot (via VPN in France)
...
2017-01-18 08:12: Testing bot (via VPN in Korea)
...
2017-01-18 11:12: Testing bot (from IP in North Korea)

175.45.***.***

inetnum:	175.45.176.0 - 175.45.179.255
netname:	STAR-KP
descr:	Ryugyong-dong
descr:	Potong-gang District
role:	STAR JOINT VENTURE CO LTD
address:	Ryugyong-dong Potong-gang District
country:	KP


Neel Mehta
@neelmehta
Following
9c7c7149387a1c79679a87dd1ba755bc @ 0x402560, 0x40F598
ac21c8ad899727137c4b94458d7aa8d8 @ 0x10004ba0, 0x10012AA4
#WannaCryptAttribut
8:02 PM - 15 May 2017
237 Retweets 305 Likes
23
237
305
Custom SSL implementation

CCleaner malware – custom base64 encoding

8f56fd14133ccd84b6395913536f5823d74737c410b829f729baaf2fe645a0a9

```
.003E121C: C3                retn ; -^-^-^-^-^-^-^-^-^-^-^-^
.003E121D: 55                push    ebp
.003E121E: 8BEC              mov     ebp,esp
.003E1220: 51                push    ecx
.003E1221: 56                push    esi
.003E1222: 57                push    edi
.003E1223: 8B7D08            mov     edi,[ebp][8]
.003E1226: 85FF              test    edi,edi
.003E1228: 0F843F010000      jz      .0003E136D --↓1
.003E122E: 837D0C00          cmp     d,[ebp][00C],0
.003E1232: 0F8435010000      jz      .0003E136D --↓1
.003E1238: 8B450C            mov     eax,[ebp][00C]
.003E123B: 6A03              push    3
.003E123D: 33D2              xor     edx,edx
.003E123F: 59                pop     ecx
.003E1240: F7F1              div     ecx
.003E1242: 6A03              push    3
.003E1244: 33D2              xor     edx,edx
.003E1246: 5E                pop     esi
.003E1247: 8BC8              mov     ecx,eax
.003E1249: 8B450C            mov     eax,[ebp][00C]
.003E124C: F7F6              div     esi
.003E124E: 8BC1              mov     eax,ecx
.003E1250: C1E002            shl     eax,2
.003E1253: 894508            mov     [ebp][8],eax
.003E1256: 85D2              test    edx,edx
.003E1258: 8955FC            mov     [ebp][-4],edx
.003E125B: 7406              jz      .0003E1263 --↓2
.003E125D: 83C004
.003E1260: 894508
```

Cbkrdr shellcode loader

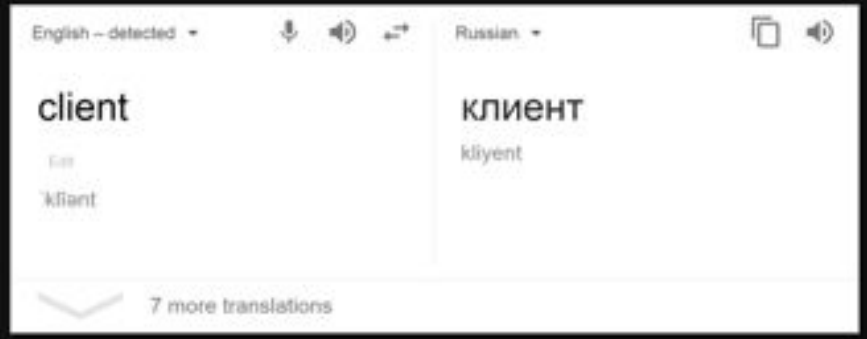
0375b4216334c85a4b29441a3d37e61d7797c2e1cb94b14cf6292449fb25c7b2

```
.00401015: C3                retn ; -^-^-^-^-^-^-^-^-^-^-^-^
.00401016: 55                push    ebp
.00401017: 8BEC              mov     ebp,esp
.00401019: 51                push    ecx
.0040101A: 56                push    esi
.0040101B: 57                push    edi
.0040101C: 8B7D08            mov     edi,[ebp][8]
.0040101F: 85FF              test    edi,edi
.00401021: 0F843F010000      jz      .00401166 --↓1
.00401027: 837D0C00          cmp     d,[ebp][00C],0
.0040102B: 0F8435010000      jz      .00401166 --↓1
.00401031: 8B450C            mov     eax,[ebp][00C]
.00401034: 6A03              push    3
.00401036: 33D2              xor     edx,edx
.00401038: 59                pop     ecx
.00401039: F7F1              div     ecx
.0040103B: 6A03              push    3
.0040103D: 33D2              xor     edx,edx
.0040103F: 5E                pop     esi
.00401040: 8BC8              mov     ecx,eax
.00401042: 8B450C            mov     eax,[ebp][00C]
.00401045: F7F6              div     esi
.00401047: 8BC1              mov     eax,ecx
.00401049: C1E002            shl     eax,2
.0040104C: 894508            mov     [ebp][8],eax
.0040104F: 85D2              test    edx,edx
.00401051: 8955FC            mov     [ebp][-4],edx
.00401054: 7406              jz      .0040105C --↓2
.00401056: 83C004
.00401059: 894508
```

Missl (APT17) backdoor

Effect: FALSE FLAGS

```
"Nachalo" - start communication session
"ustanavlivat" - handshake state
"poluchit" - receive data
"pereslat" - send data
"derzhat" - maintain communication session
"vykhodit" - exit communication session

"kliyent2podklyuchit” - client2connect ??
```




```
private function put_dummy_args(param1:*) : *
      {
         return chainik.call.apply(null,param1);
      }
```

NSA Official Suggests North Korea Was Culprit in Bangladesh Bank Heist

The deputy director of the NSA says he believes states have entered the bank-robbing business.

GREAT

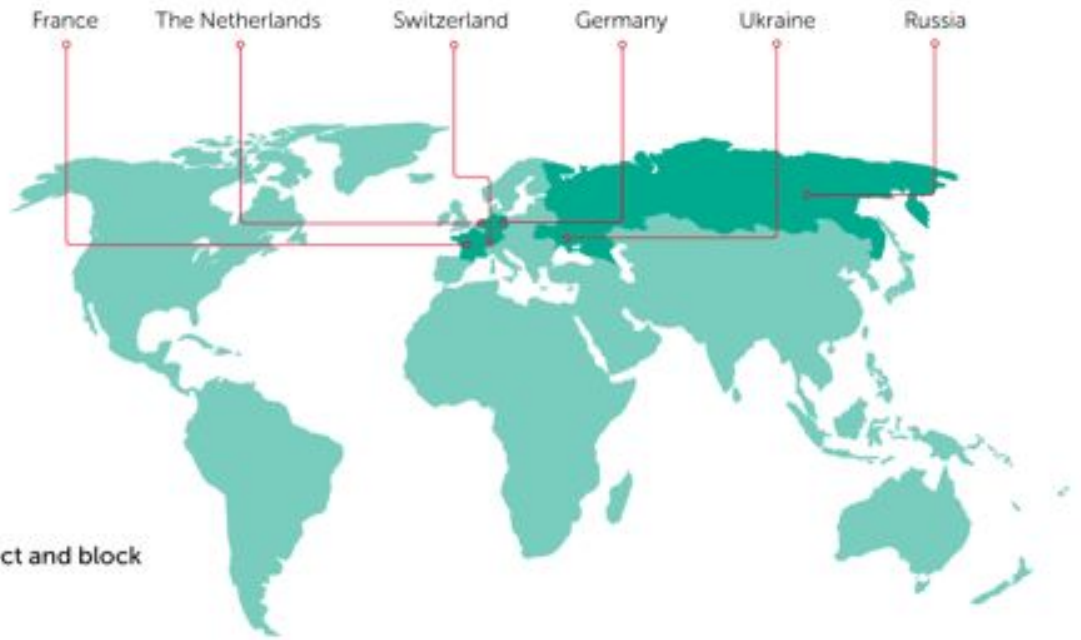
Targets of recent Olympic Destroyer attacks

In May-June 2018 Kaspersky Lab discovered new spear-phishing documents related to Olympic Destroyer. The threat actor had previously attacked Winter Olympics infrastructure.

Targets:

Biological and chemical threat prevention organizations

Financial institutions (in Russia only)



Kaspersky Lab products successfully detect and block Olympic Destroyer-related malware.

KASPERSKY lab GREAT

Hungry for more?

Youtube: “Operation Blue Sky Kaspersky”
https://youtu.be/pX43RhMZ86A

Ping me on Discord :)

Thank you

UNbreakable Romania

kaspersky