Who littered the sandbox?
Scooping up new malware behavior.

Olaf Hartong | Defensive Specialist | FalconForce

EU ATT&CK Community– 02-06-2021

Who am I?

Olaf Hartong

Defensive Specialist

FalconForce

Detection Engineer / Security Researcher

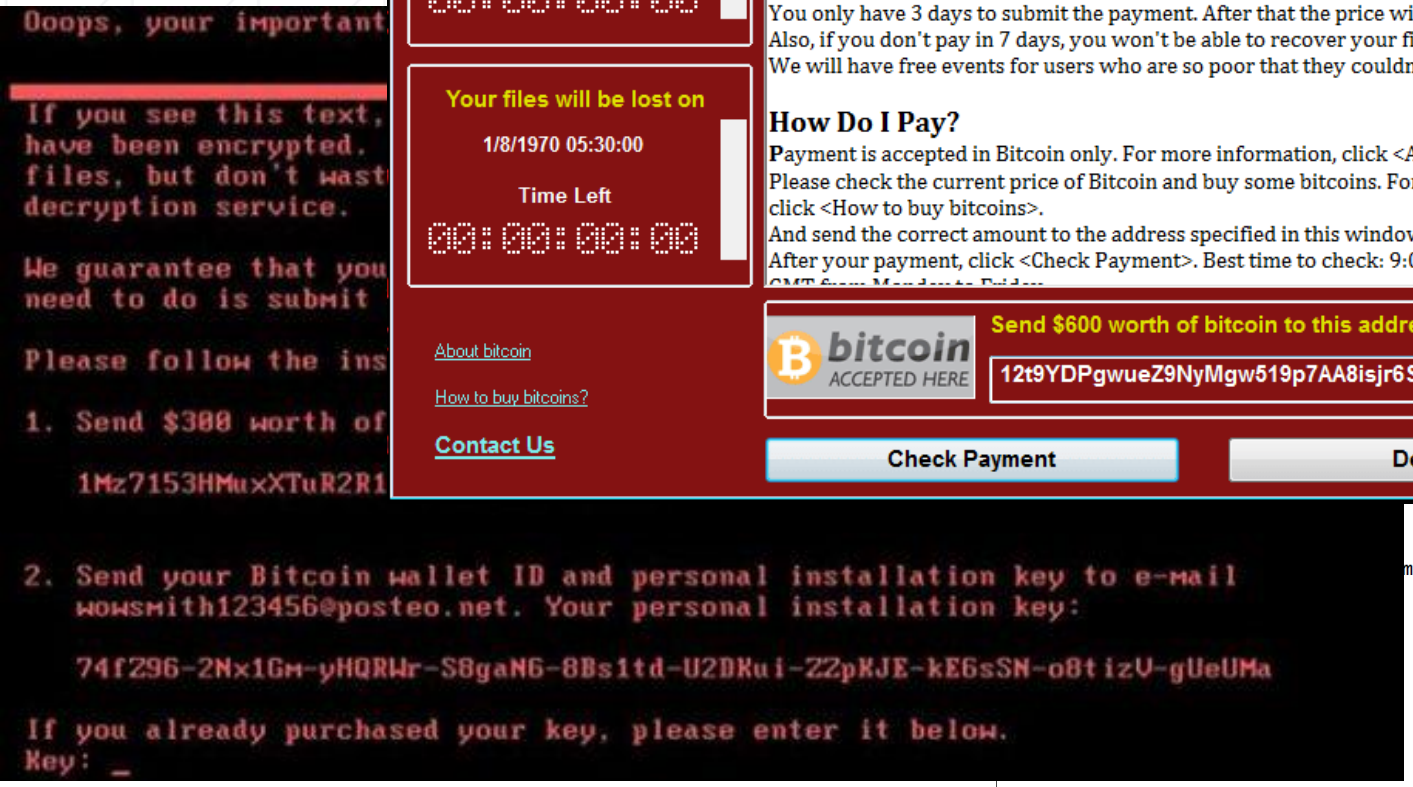
- Built and/or led Security Operations Centers
- Threat hunting / Incident Response

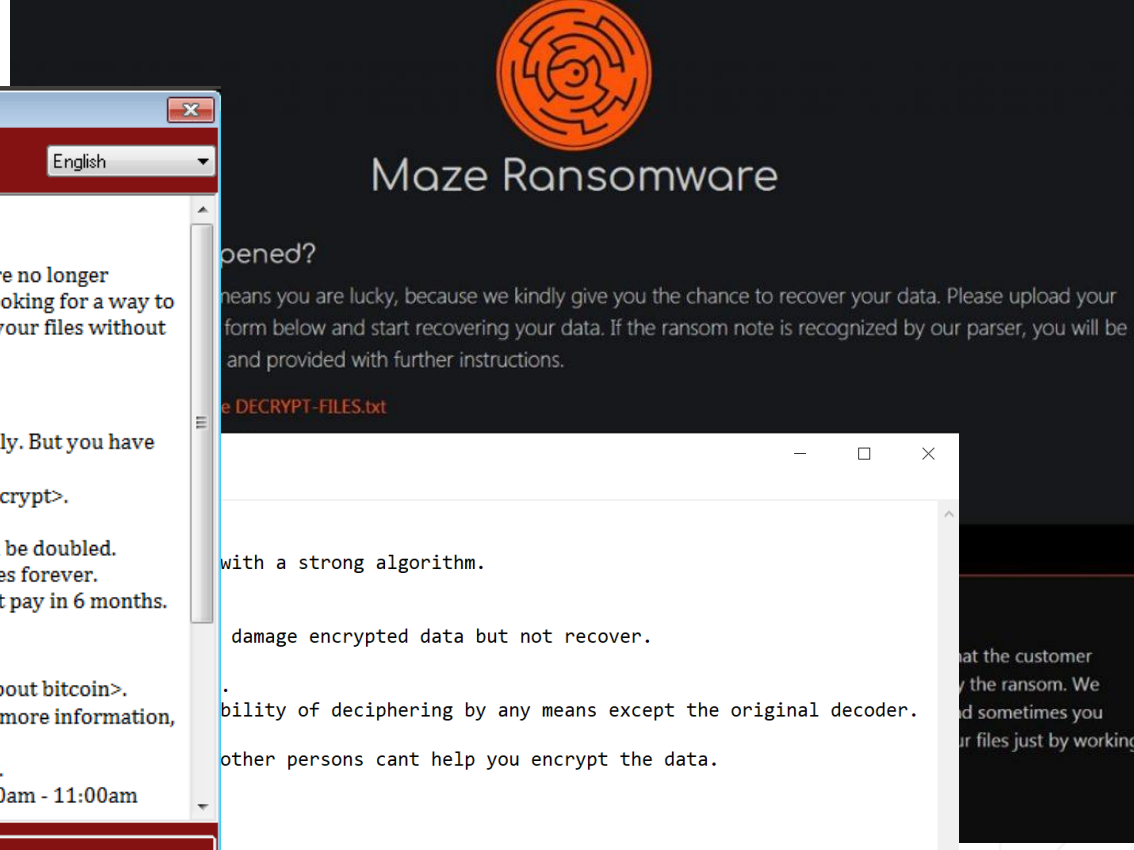
Former documentary photographer
Father of 2 boys
“I like warm hugs”

@olafhartong
github.com/olafhartong
olaf@falconforce.nl
olafhartong.nl / falconforce.nl

SO ?


Wana Decrypt0r 2.0
Ooops, your files have been encrypted!
English
What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.
Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.
How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am
Payment will be raised on
1/4/1970 05:30:00
Time Left
00:00:00:00
Your files will be lost on
1/8/1970 05:30:00
Time Left
00:00:00:00
About bitcoin
How to buy bitcoins?
Contact Us
Send $600 worth of bitcoin to this address:
bitcoin ACCEPTED HERE
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
Copy
Check Payment
Decrypt


Ooops, your important
If you see this text,
have been encrypted.
files, but don't wast
decryption service.
We guarantee that you
need to do is submit
Please follow the ins
1. Send $300 worth of
1Mz7153HMuxXTuR2R1
2. Send your Bitcoin wallet ID and personal installation key to e-mail
wowsmith123456@posteo.net. Your personal installation key:
74f296-2Nx1Gm-yHQRWr-S8gaN6-8Bs1td-U2DKui-Z2pKJE-kE6sSN-o8tizV-gUeUMa
If you already purchased your key, please enter it below.
Key: _


Maze Ransomware
pened?
neans you are lucky, because we kindly give you the chance to recover your data. Please upload your
form below and start recovering your data. If the ransom note is recognized by our parser, you will be
and provided with further instructions.
e DECRYPT-FILES.txt
with a strong algorithm.
damage encrypted data but not recover.
bility of deciphering by any means except the original decoder.
other persons cant help you encrypt the data.
iles and you will get it decrypted.
ure that one key decrypts everything.
ment in the reply letter
at the customer
the ransom. We
d sometimes you
r files just by workin

Where it all began

ZDNet

MUST READ: The IT skills gap is a giant p

The malware that usu and you need to remo

If you see any of these malware strains on your enter systems.

By Catalin Cimpanu for Zero Da GMT (05:45 GMT) | Topic: Secu

Image: Lina White

Emotet

Emotet is considered today's biggest malware botnet.

There are few cases where Emotet has dealt with ransomware gangs directly, but many ransomware infections have been traced back to initial Emotet infections.

Usually, Emotet sold access to its infected systems to other malware gangs, which later sold their own access to ransomware gangs.

Today, the most common ransomware infection chain linked back to Emotet is: **Emotet—Trickbot—Ryuk**

Trickbot

Trickbot is a malware botnet and cybercrime similar to Emotet. Trickbot infects its own victims but is also known to buy access to Emotet-infected systems in order to boost its numbers.

Over the past two years, security researchers have seen Trickbot sell access to its systems to cybercrime gangs that later deployed Ryuk, and later the Conti ransomware.

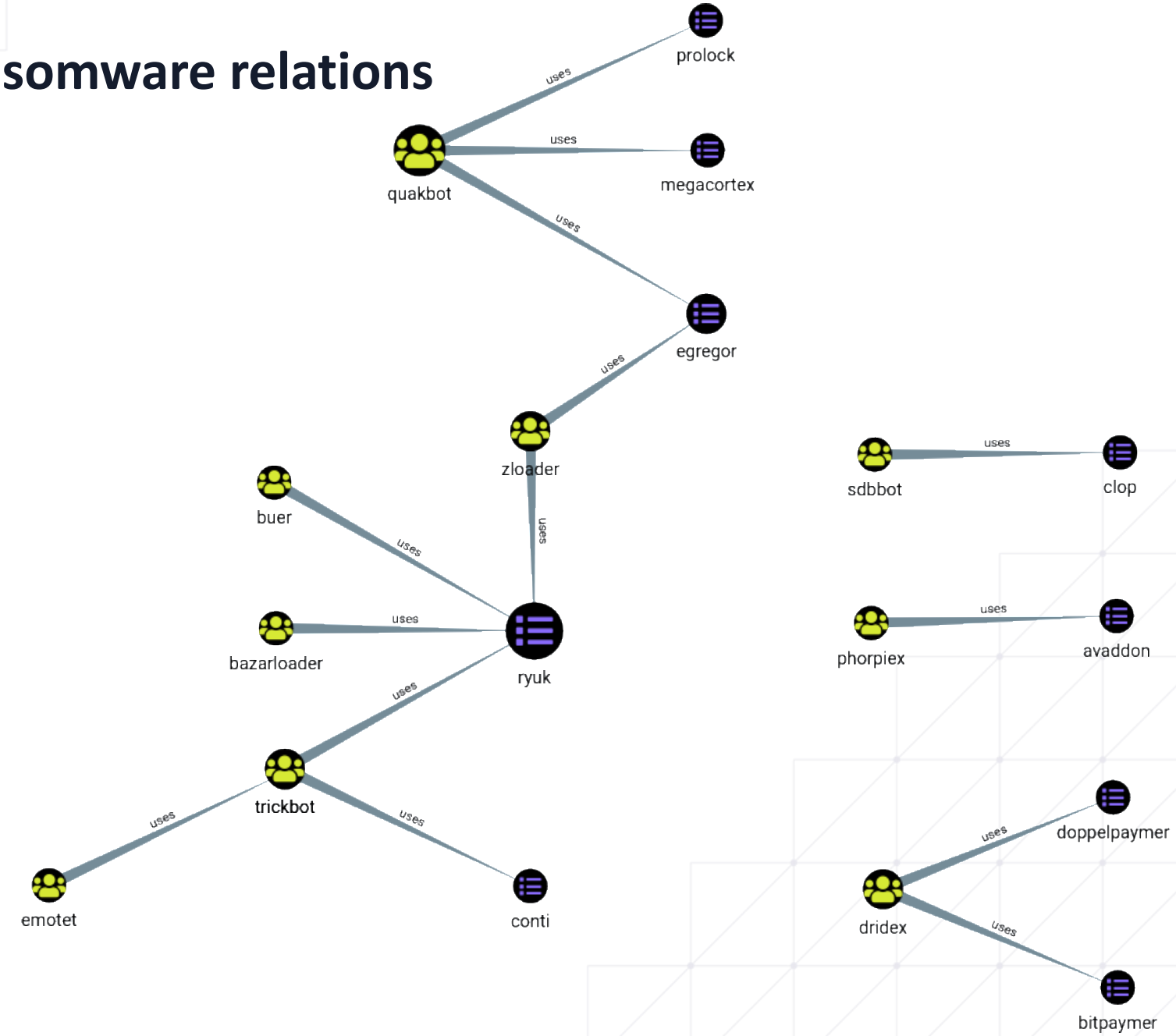
Trickbot—Conti

Trickbot—Ryuk

ElectroRAT malware

Malware launching Ransomware relations

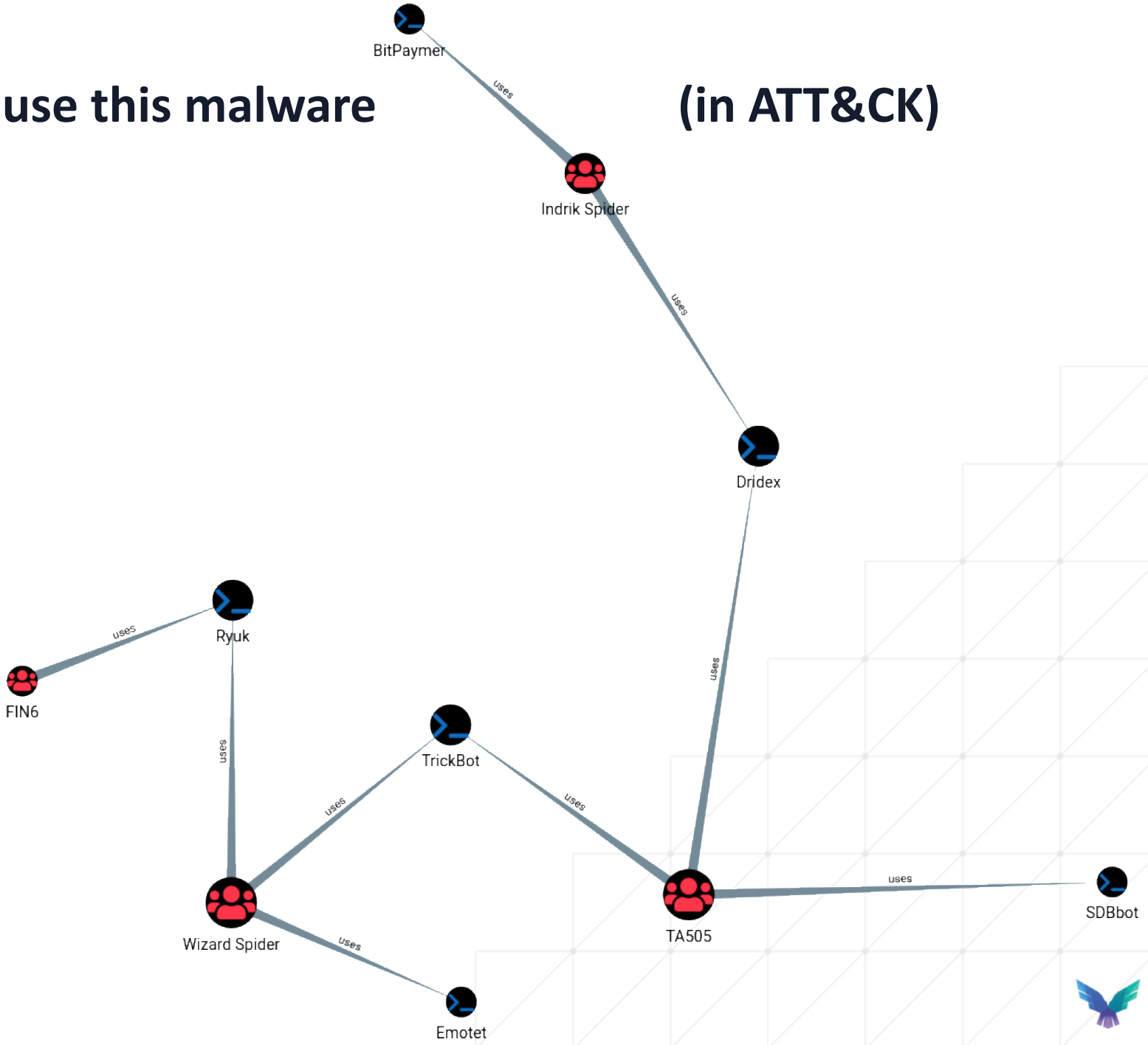
From the report :
Emotet —> Trickbot
Trickbot —> Conti
Trickbot —> Ryuk
BazarLoader —> Ryuk
QakBot —> MegaCortex
QakBot —> ProLock
QakBot —> Egregor
SDBBot —> Clop
Dridex —> BitPaymer
Dridex —> DoppelPaymer
Zloader —> Egregor
Zloader —> Ryuk
Buer —> Ryuk
Phorpiex —> Avaddon



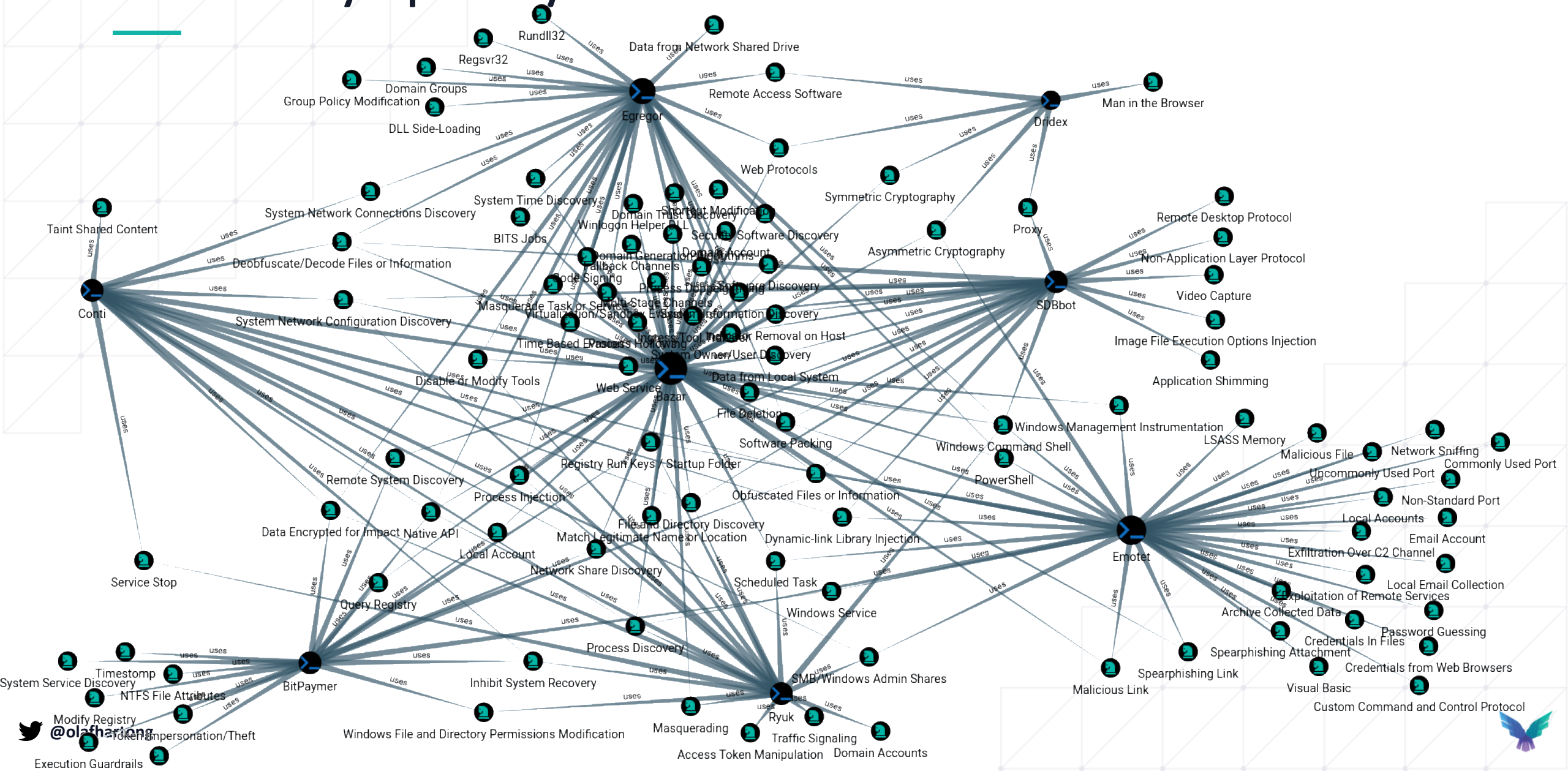
https://zdnet.com/article/the-malware-that-usually-installs-ransomware-and-you-need-to-remove-right-away

What groups are known to use this malware (in ATT&CK)

- 7 out of 18 malware families are known in ATT&CK
- 6 out of 18 malware families are attributed to an actor group
- Not all possible uses are mapped to a group or software, only the ones publicly reported and processed by MITRE



What do they reportedly do ?

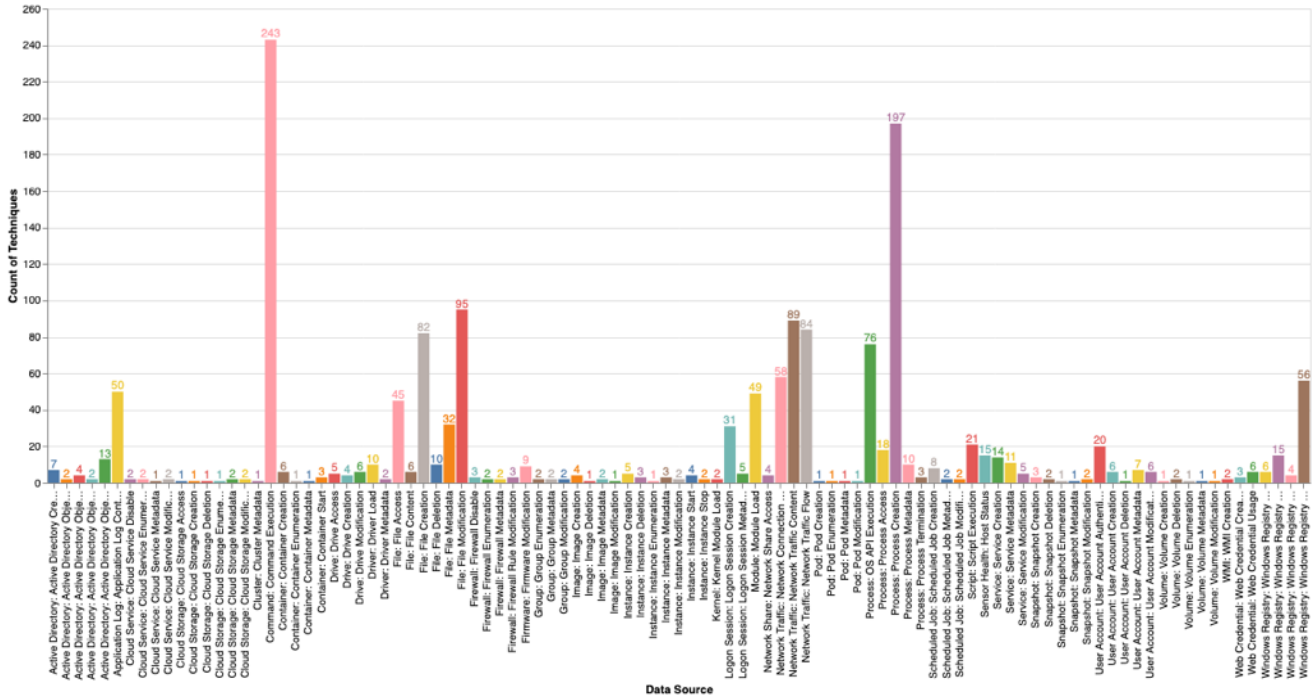


@olafhartong

It always starts with data

Be aware that is a *huge* pile of data waiting to be analyzed, in this talk I focus primarily on **Windows**. However there is a lot more you might require to have solid coverage.

<u>Don't just hoard it though!</u>



Windows:
Event logs
Sysmon
EDR logs
PowerShell scriptblock, modules and transcripts
Application logs
ETW
Debug logs

Linux:
Syslog
EDR logs
Auditd
Osquery

OS X:
Syslog
EDR logs
Osquery
ESF logs

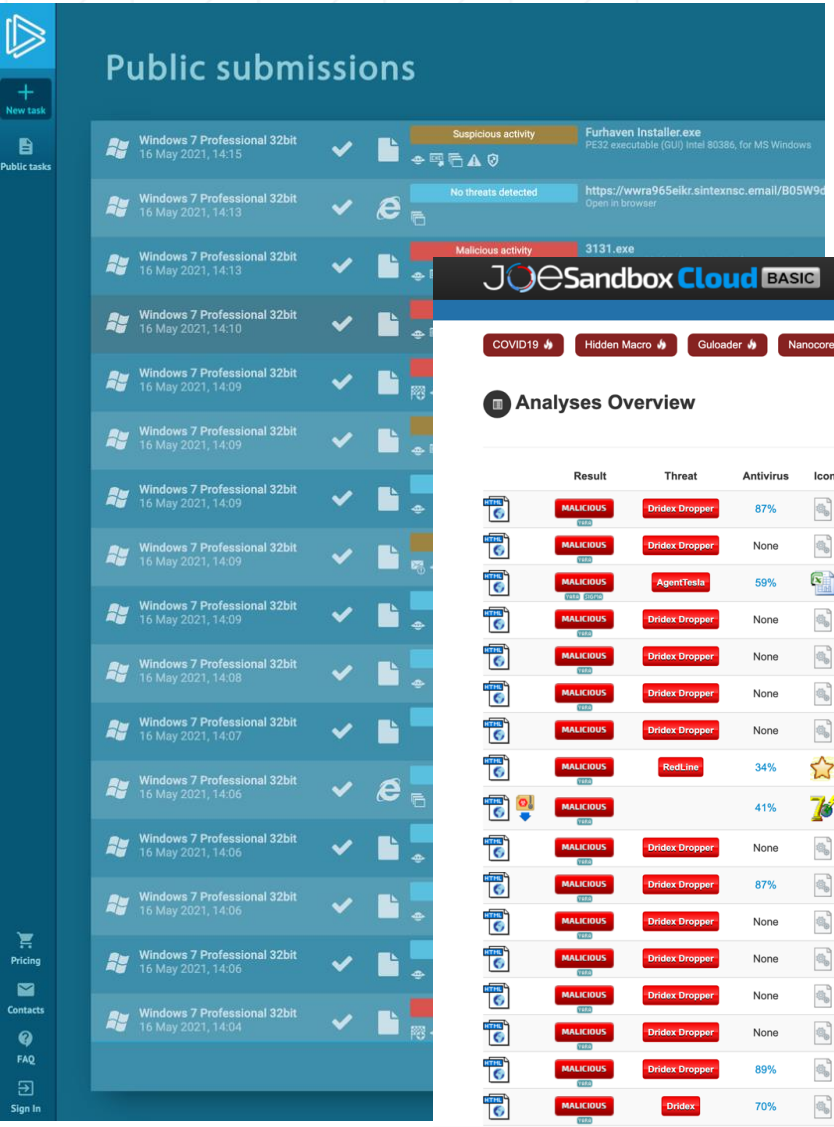
Network:
Proxy
DNS
Firewall
IDS/IPS
Bro/Zeek
Netflow
PCAP data
Appliance logs

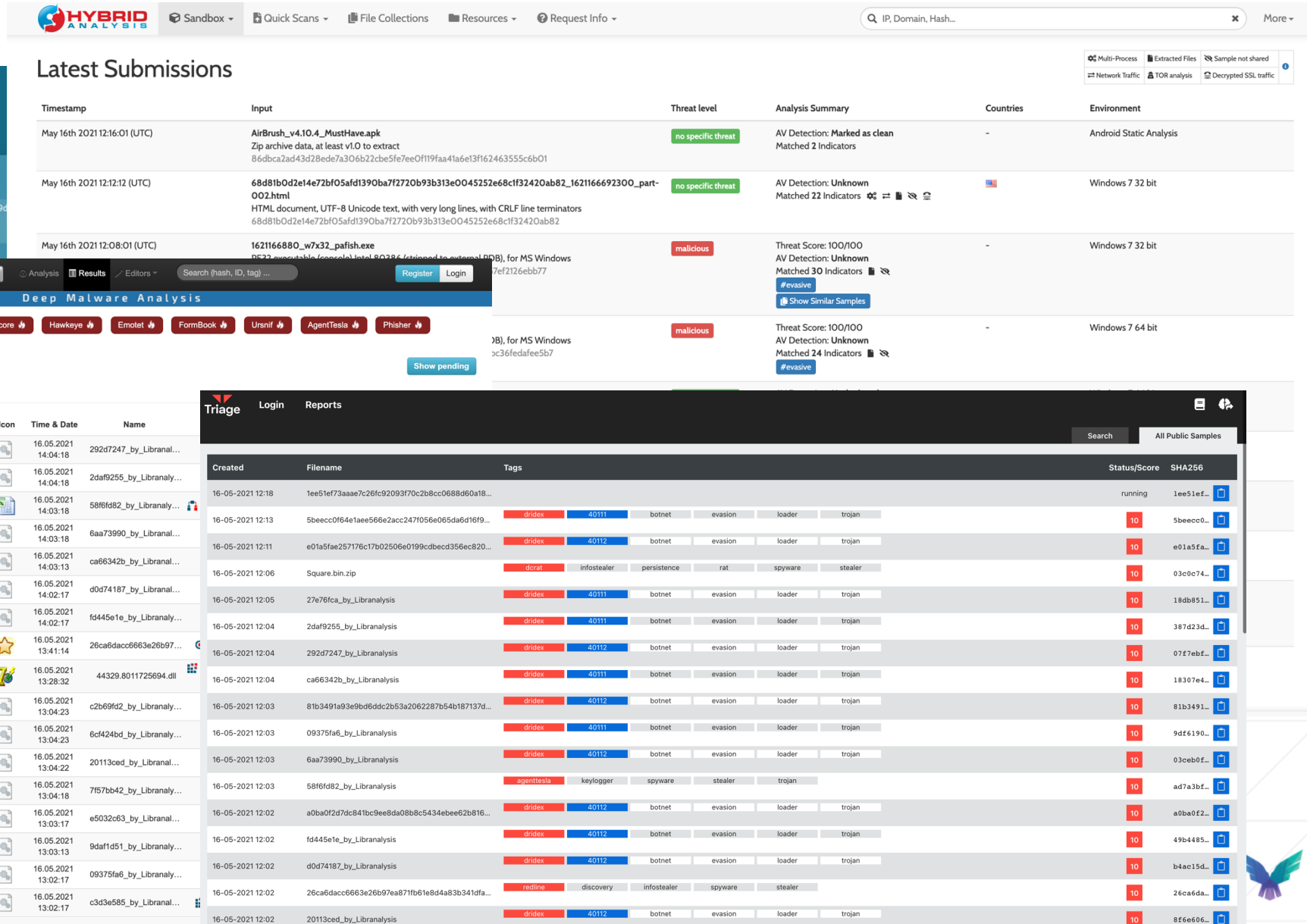
Cloud platforms
Email log
SaaS apps
Webserver Logs
Internal applications
Sandboxes

Much much more…..

Shout out to the ATT&CK team for improving the data source model!

Public sandboxes

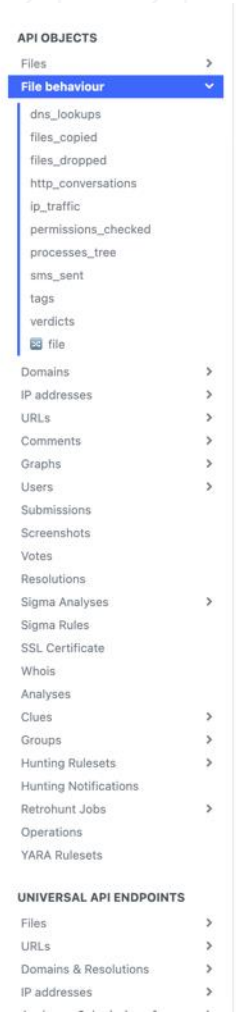




Why I didn’t decide to use them

- Great resources for analyzing a handful of files
- Very manual for the free versions, not ideal at scale
- Output not easy to parse and compare

Virustotal



File behaviour

File behaviour reports

File behaviour reports are obtained either by using the GET /files/{id}/behaviours endpoint or the sandbox behavior feed . They summarize the observed behaviour during the execution or opening of a file. Note that some of these actions could be triggered by children of the file under consideration.

Object Attributes

In a file_behaviour object you will find these attributes:

- `analysis_date` : *<integer>* Unix epoch UTC time (seconds).
- `behash` : *<string>* used to find similar behaviour analyses.
- `calls_highlighted` : *<list of strings>* API calls/Syscalls worth highlighting.
- `command_executions` : *<list of strings>* shell command executions observed during the analysis of the given file.
- `files_opened` : *<list of strings>* files opened during execution.
- `files_written` : *<list of strings>* files written to during execution.
- `files_deleted` *<list of strings>* names of the files deleted.
- `files_attribute_changed` : *<list of strings>* full path of files subject to some sort of active attribute modification.
- `has_html_report` : whether there is an HTML report for this behaviour analysis.
- `has_pcap` : whether there is a PCAP network capture for this behaviour analysis.
- `hosts_file` : *<string>* the hosts file field stores the content of the local hostname-ip mapping hosts file IF AND ONLY IF the file was modified, else this field is not populated.
- `ids_alerts` : *<list of dictionaries>* list of IDS alerts, sorted by timestamp. Every item contains the following keys:
  - `alert_context` : *<dictionary>* matched alert context:
    - `dest_ip` : *<string>* destination IP.
    - `dest_port` : *<integer>* destination port.
    - `hostname` : *<string>* if the alert is related to an HTTP communication, destination hostname.
    - `protocol` : *<string>* communication protocol name.
    - `src_ip` : *<string>* source IP.
    - `src_port` : *<integer>* source port.
    - `url` : *<string>* if the alert is related to an HTTP communication, destination URL.
  - `alert_severity` : *<string>* one of `high` , `medium` , `low` or `info` .
  - `rule_category` : *<string>* alert category.
  - `rule_id` : *<string>* rule SID.
  - `rule_msg` : *<string>* alert description.
  - `rule_source` : *<string>* rule source, determined by SID range.
- `processes_terminated` : *<list of strings>* name of the processes that were terminated during the execution of a given file.
- `processes_killed` : *<list of strings>* name of the processes that were killed during the execution of a given file.
- `processes_injected` : *<list of strings>* name of the processes that were subjected to some kind of code injection during the execution of the given file.
- `services_opened` : *<list of strings>* names of the services for which a handle was acquired during the analysis of the given file.
- `services_created` : *<list of strings>* new services created.
- `services_started` : *<list of strings>* new services started.
- `services_stopped` : *<list of strings>* services stopped during the execution of the given file.
- `services_deleted` : *<list of strings>* services deleted during the execution of the given file.

Why I didn’t decide to use them

- Great API, also available for free up to 500 files a day (although not properly enforced)
- Loads of files and behavioral data available
- Several different vendor sandbox solutions on the backend,
  with VERY different quality of output, making it untrustworthy for comparison analysis

Custom sandbox source

- Customized Cuckoo farm
- JSON output
- 2500-3500 processed samples a day
- Broad dataset,
  Behavior and static information
  - Process
  - File
  - DLL-loads
  - Network
  - API
  - Registry
  - Macros
  - Imports, Exports, Strings
  - Much more

Ingesting the data

As an analysis platform I chose Azure Sentinel

- Cloud hosted
- Fast
- Kusto Query Language (KQL)
- Nested JSON support

Ingesting custom data can be done through:

- Syslog
- The API directly
- Logstash with Microsoft supported output plugin

Easy right ?

https://azure.microsoft.com/en-us/services/azure-sentinel/
https://www.elastic.co/logstash

Well…..not completely

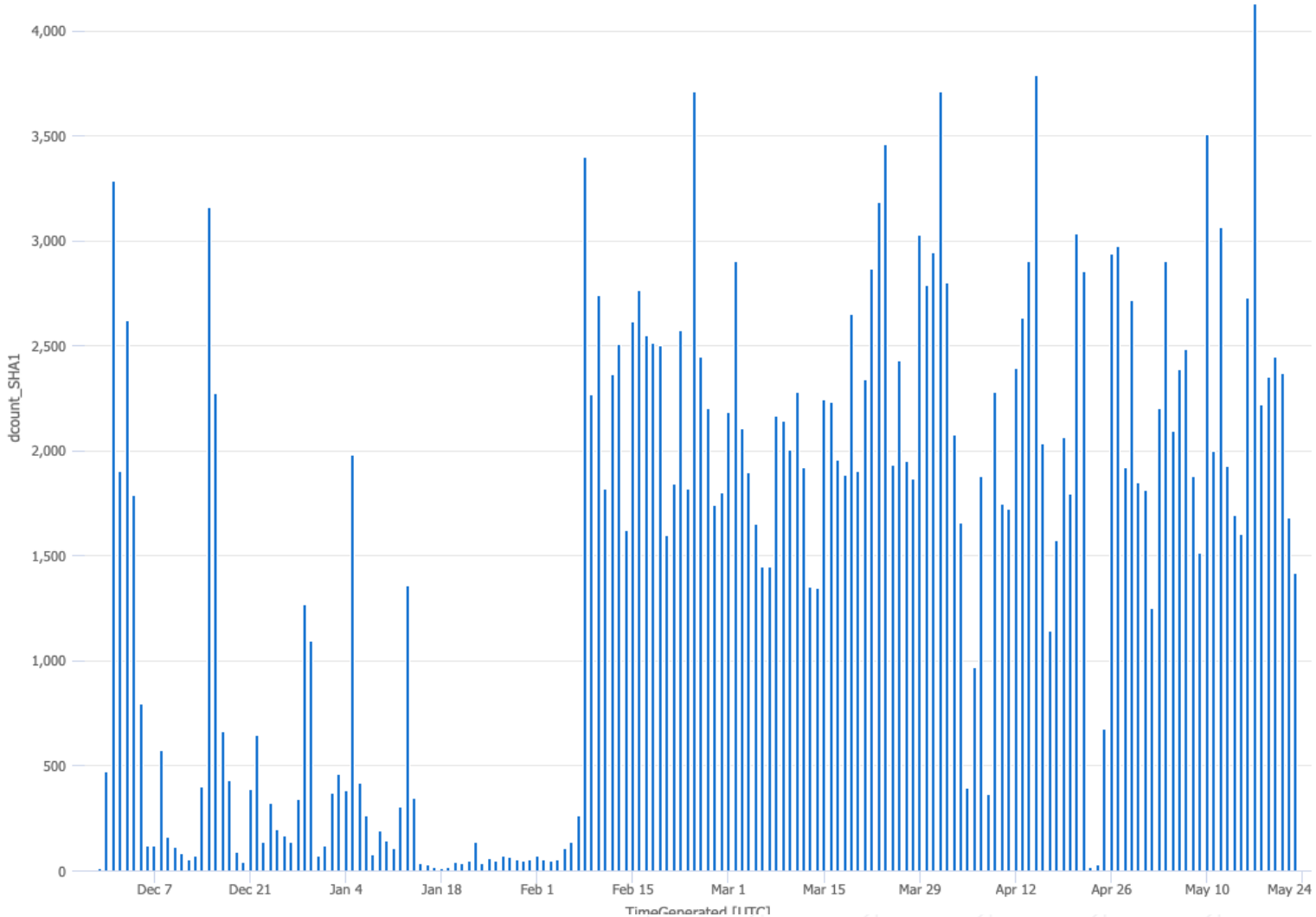
Some limitations I found out the hard way :P

- Logstash does not like large Pretty-JSON files
- The API only accepts 4 layers deep of nested data
- There is a limit of 64kb data per field
- If you overfeed the API it will rate limit and confuse Logstash

```
[INFO ] 2021-01-17 20:44:35.424 [[main]-pipeline-manager] javapipeline - Pipeline started {"pipeline.id"=>"main"}
[INFO ] 2021-01-17 20:44:35.511 [Agent thread] agent - Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[INFO ] 2021-01-17 20:44:35.538 [[main]<file] observingread - START, creating Discoverer, Watch with file and sincedb collections
[INFO ] 2021-01-17 20:44:35.869 [Api Webserver] agent - Successfully started Logstash API endpoint {:port=>9600}
[INFO ] 2021-01-17 20:45:08.436 [[main]>worker0] azureloganalytics - Changing buffer size.[configuration='2000' , new_size='1900']
[INFO ] 2021-01-17 20:45:11.115 [[main]>worker0] azureloganalytics - Successfully posted 2 logs into custom log analytics table[OlafsMalwareStories].
[INFO ] 2021-01-17 20:45:15.912 [Ruby-0-Thread-4: :1] azureloganalytics - Changing buffer size.[configuration='1900' , new_size='1800']
[ERROR] 2021-01-17 20:45:25.007 [Ruby-0-Thread-4: :1] azureloganalytics - Exception in posting data to Azure Loganalytics.
[Exception: '413 Payload Too Large]'
[INFO ] 2021-01-17 20:45:25.467 [Ruby-0-Thread-4: :1] azureloganalytics - Resending 498 documents as log type OlafsMalwareStories to DataCollector API in 2 seconds.
[INFO ] 2021-01-17 20:45:37.842 [Ruby-0-Thread-4: :1] azureloganalytics - Resending 498 documents as log type OlafsMalwareStories to DataCollector API in 2 seconds.
```


Success! We have data

- Built parser functions per windows data source
- MDE like schema for ease of use
- 7 months of data
- 2370 malware families
- 400k unique samples
- Largest sample 638 MB!
- Smallest sample 782 bytes



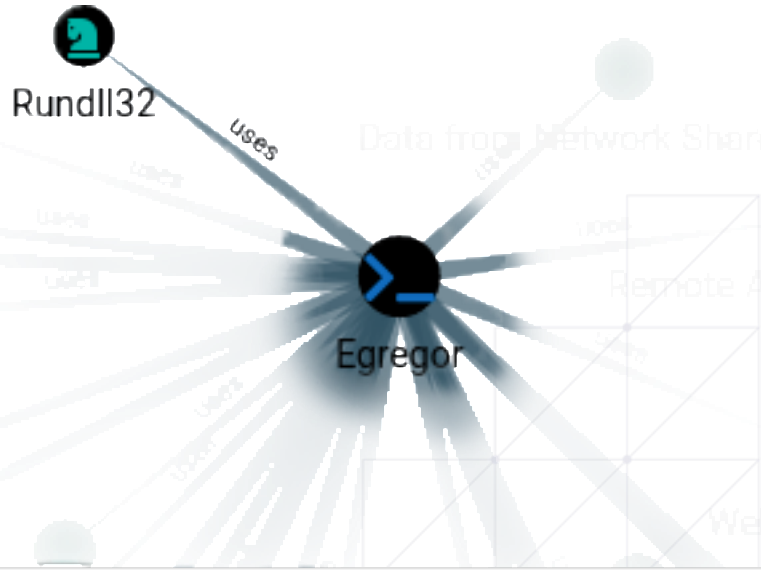
Why do I through all this trouble ?

Realistic and current use by malware of:

- Lolbins
- DLL Functions
- Registry techniques
- Macro techniques
- Network indicators
- API calls
- DLL load patterns
- Tool prevalence
  (like cobaltstrike, metasploit etc.)
- Safe and volumetric
- File drop patterns
- Enumeration or changes for SACL ideas
- Anything we can actually detect that stands out…

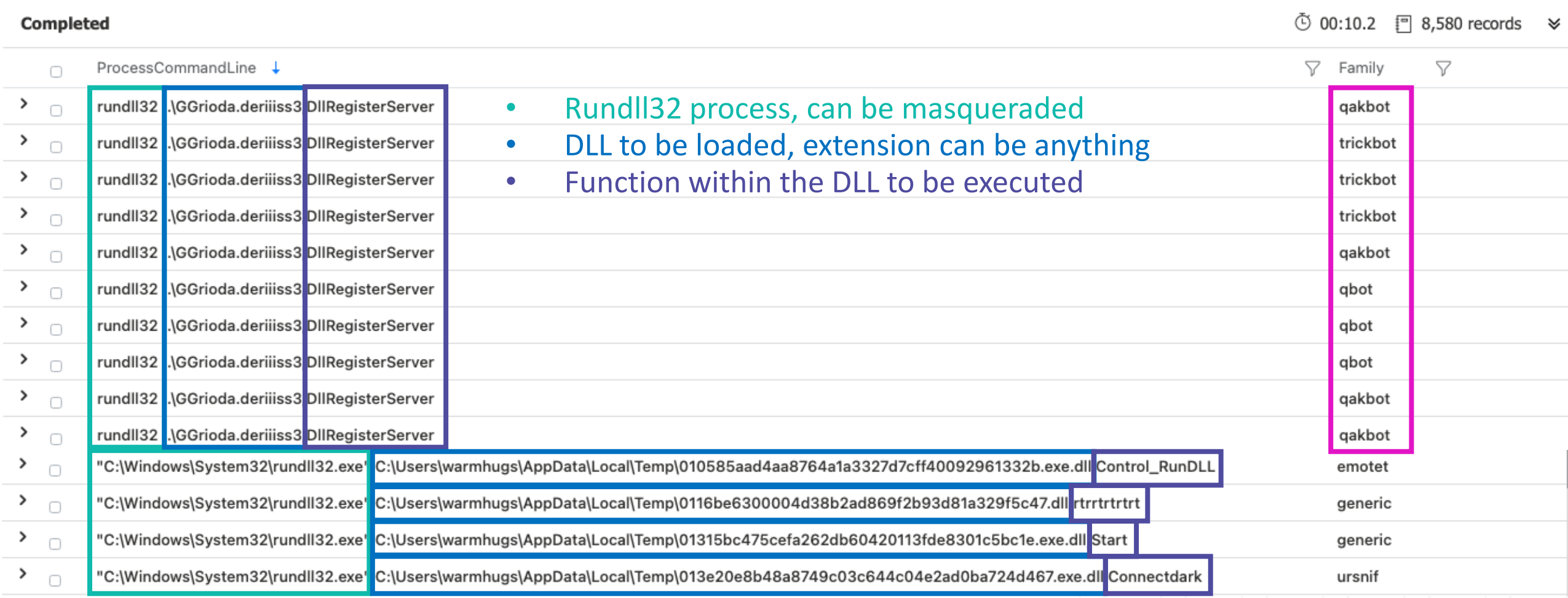
Rundll32

- Really? Out of the selected malware, only Egregor uses this?
- Remember, Not all possible uses are mapped to a technique, only the ones publicly reported / contributed and processed by MITRE



ProcessCommandLine	Family
"C:\Windows\System32\rundll32.exe" C:\Users\warmhugs\AppData\Local\Temp\do_not_delete_a20585939e9a811f5db9ac7fe22ca72273209a23.exe.dll,DllInstall	egregor
"C:\Windows\System32\rundll32.exe" C:\Users\warmhugs\AppData\Local\Temp\bc8f21f2e76828359e7b40bad85ea59819a2ab27.exe.dll,Boris	egregor
"C:\Windows\System32\rundll32.exe" C:\Users\warmhugs\AppData\Local\Temp\8af5d9218f4cc15ad4a75236398c13e1c2581cea.exe.dll,Boris	egregor
"C:\Windows\System32\rundll32.exe" C:\Users\warmhugs\AppData\Local\Temp\do_not_delete_312cf54c3fe873cf0159a0071f9492dce2628f59.exe.dll,DllInstall	egregor

Rundll32 command-lines



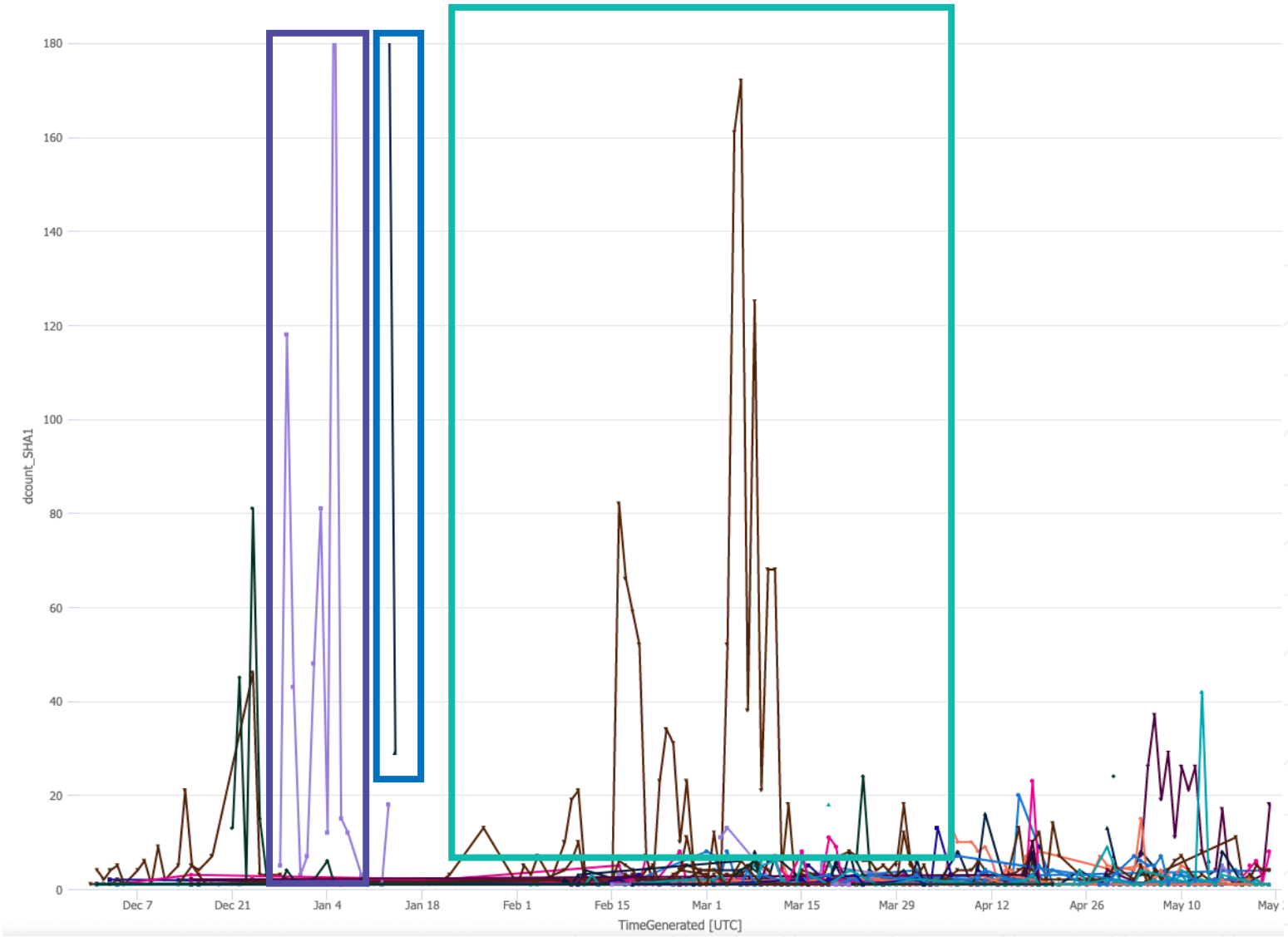
Completed 00:10.2 8,580 records

ProcessCommandLine			Family
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qakbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	trickbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	trickbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	trickbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qakbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qakbot
rundll32	.\GGrioda.deriiiss3	DllRegisterServer	qakbot
"C:\Windows\System32\rundll32.exe"	C:\Users\warmhugs\AppData\Local\Temp\010585aad4aa8764a1a3327d7cff40092961332b.exe.dll	Control_RunDLL	emotet
"C:\Windows\System32\rundll32.exe"	C:\Users\warmhugs\AppData\Local\Temp\0116be6300004d38b2ad869f2b93d81a329f5c47.dll	rtrrtrtrtrt	generic
"C:\Windows\System32\rundll32.exe"	C:\Users\warmhugs\AppData\Local\Temp\01315bc475cefa262db60420113fde8301c5bc1e.exe.dll	Start	generic
"C:\Windows\System32\rundll32.exe"	C:\Users\warmhugs\AppData\Local\Temp\013e20e8b48a8749c03c644c04e2ad0ba724d467.exe.dll	Connectdark	ursnif

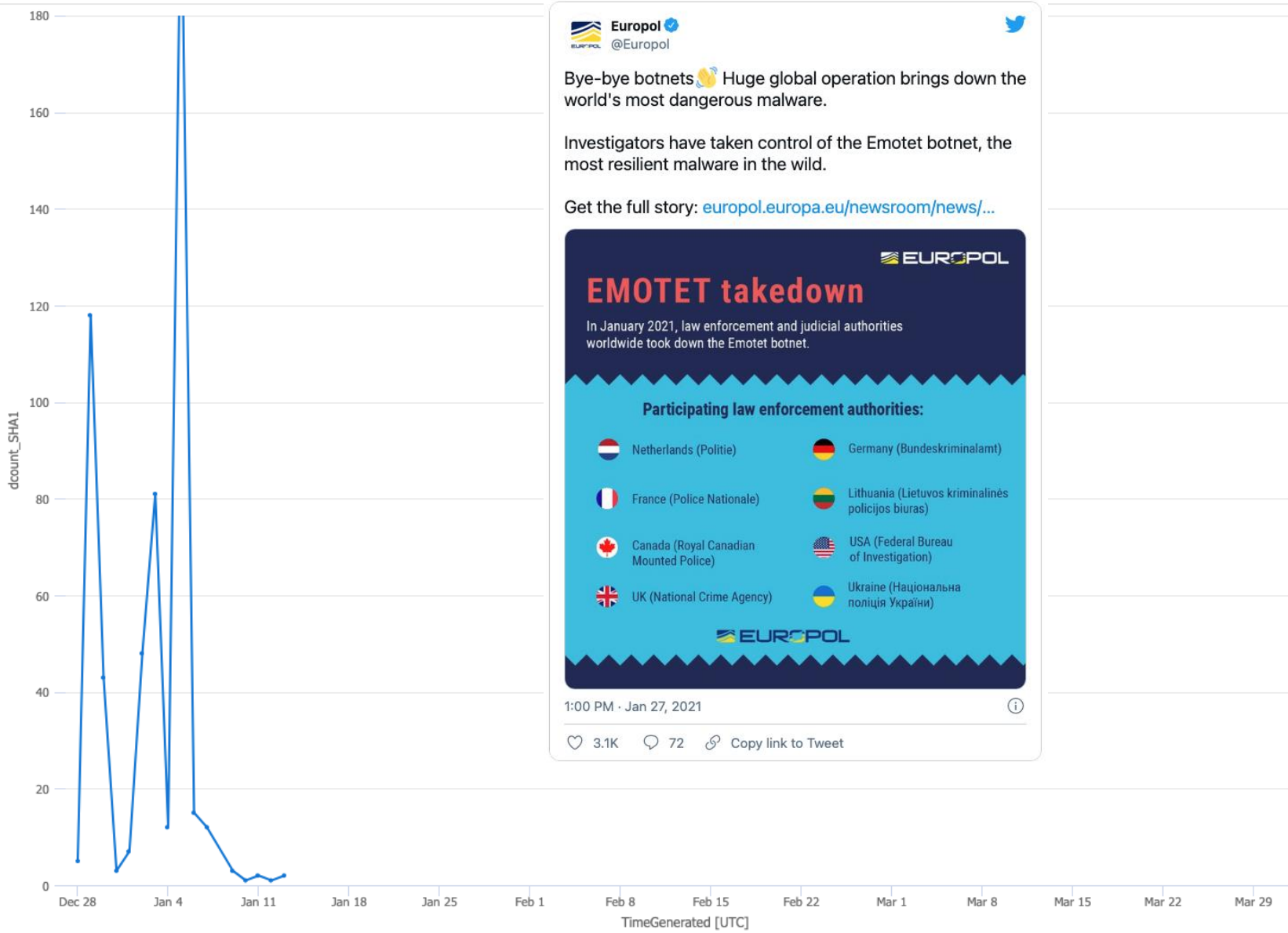
- Rundll32 process, can be masqueraded
- DLL to be loaded, extension can be anything
- Function within the DLL to be executed

What are the top used functions?

Function	count_
DllRegisterServer	4,713
TestRasm12	691
Control_RunDLL	573
Connectdark	260
RunDLL	181
None	162
DllServer	132
dbkFCallWrapperAddr	101
_ReflectiveLoader	78
StartW	78
DllCanUnloadNow	65
rtrrtrtrtrt	48
uvlcopdlxoed	44
DllRegisterServer1	42
Function	41
Init	39
Bgcedtxsf	36
Brightnight	32
GetFileVersionInfoA	32



Who uses that Control_RunDLL function?



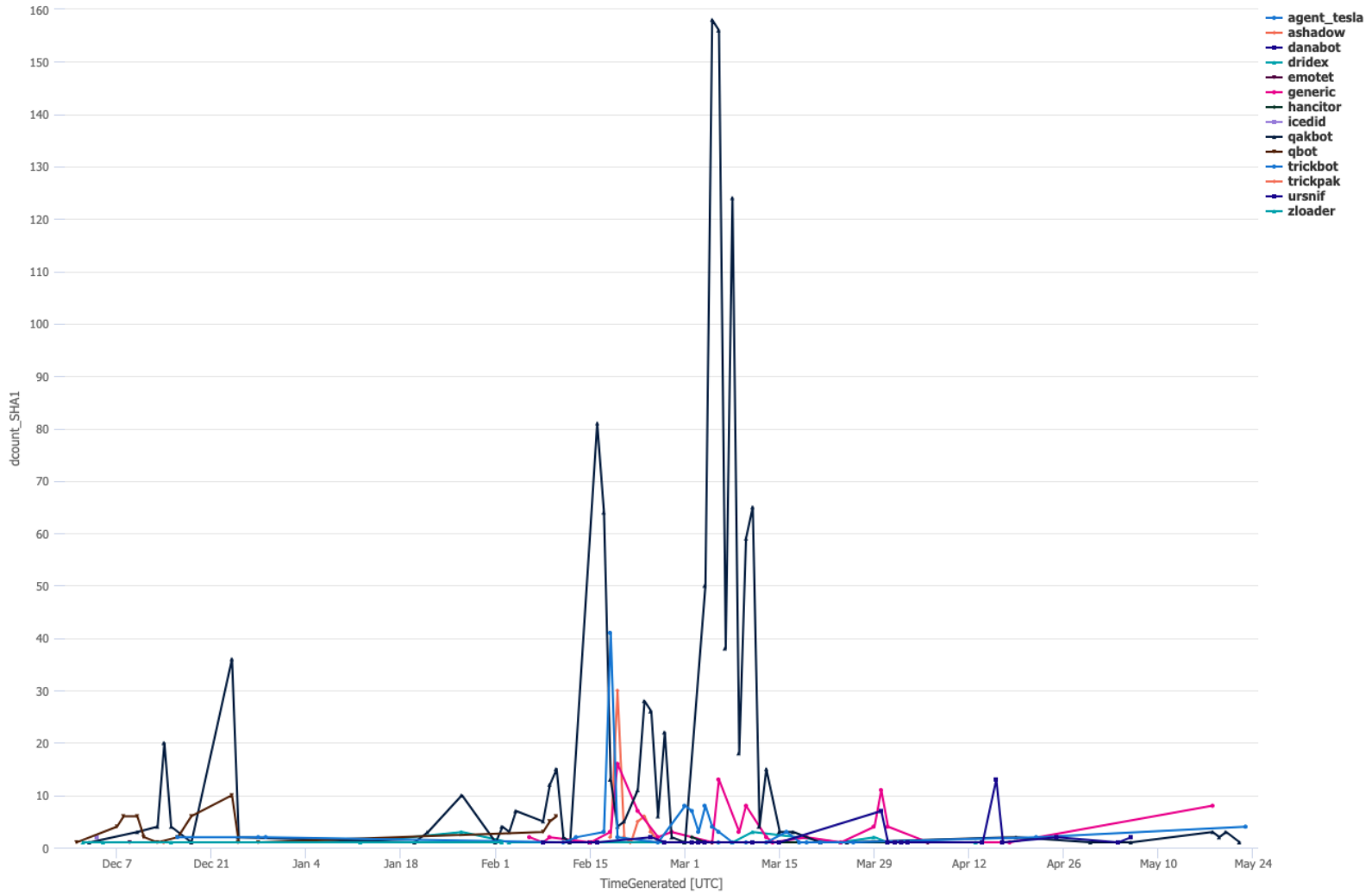
Who uses that DllRegisterServer function?

Top 3:

- qakbot
- trickbot
- ashadow

But also:

- dridex
- zloader



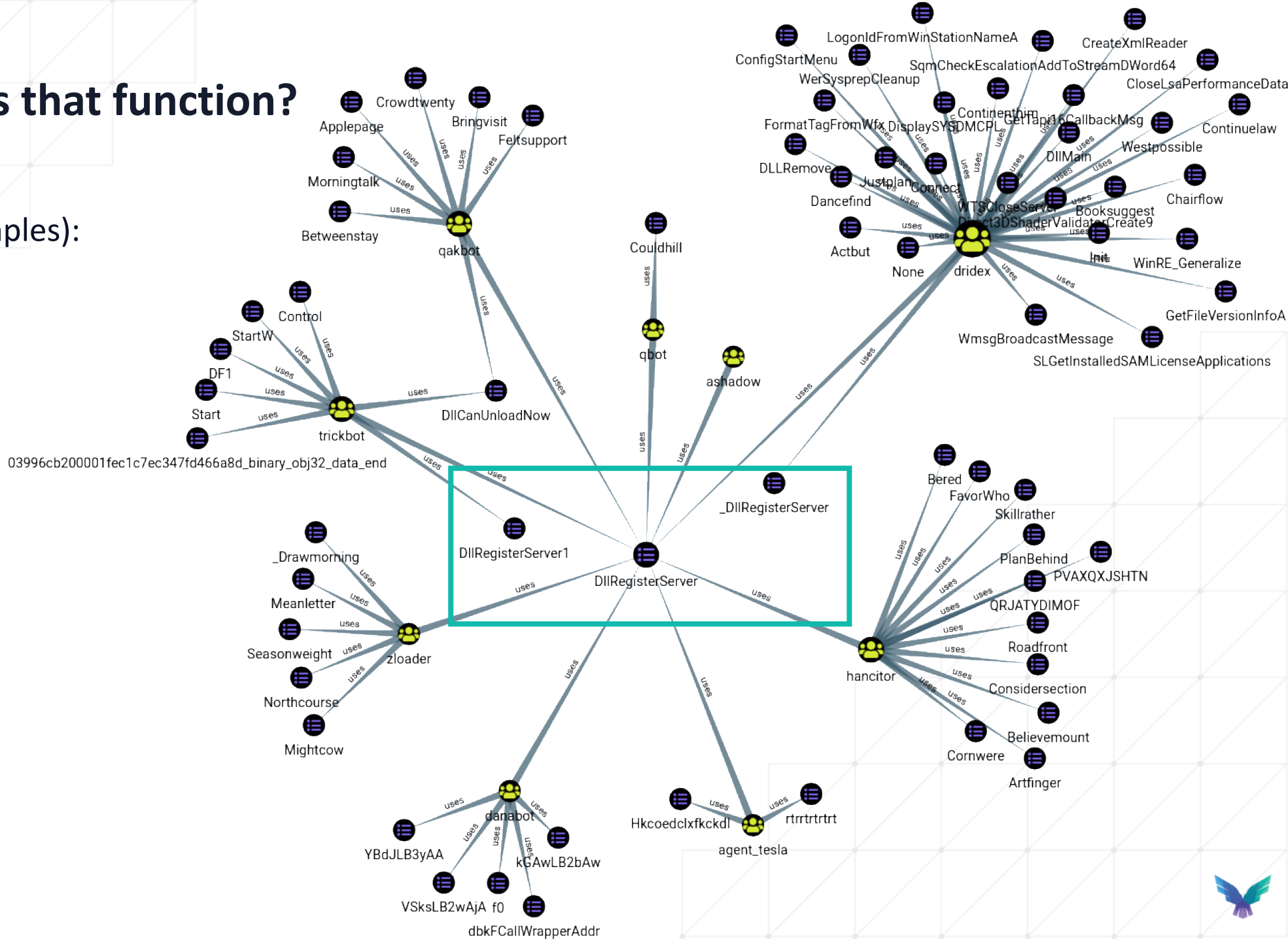
Who else uses that function?

Top 3 (unique samples):

- qakbot
- trickbot
- ashadow

But also:

- dridex
- zloader

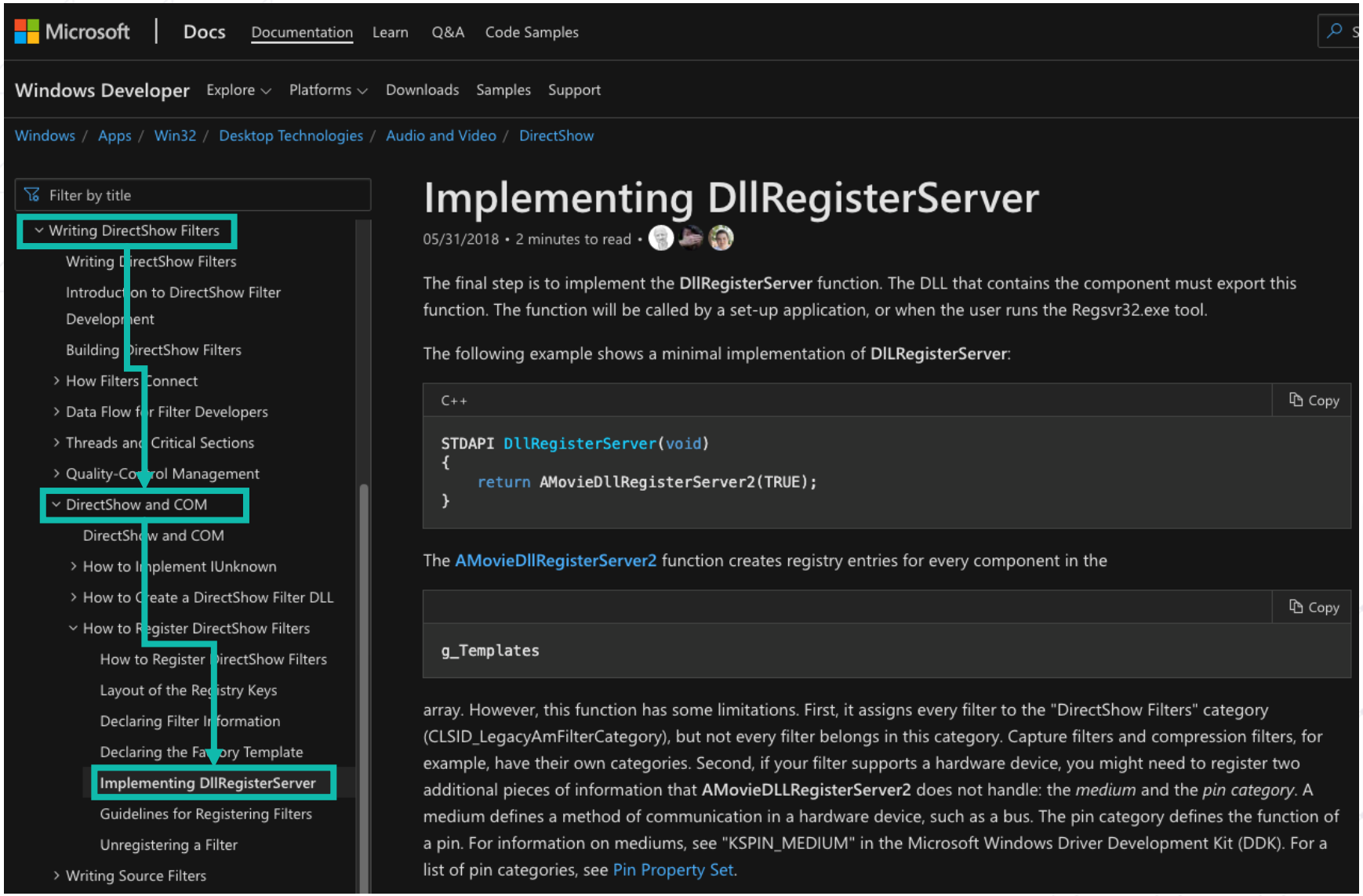


And what about in a production environment?

In several very large global environments these are the only results of the same function name being used

Function	FileName	InitiatingProcessFileName	ProcessCommandLine
DllRegisterServer	rundll32.exe	install.exe	"rundll32.exe" "C:\Program Files (x86)\Microsoft Silverlight\xapauthenticodesip.dll",DllRegisterServer
DllRegisterServer	rundll32.exe	rundll32.exe	"rundll32.exe" "C:\Program Files (x86)\Microsoft Silverlight\xapauthenticodesip.dll",DllRegisterServer

So, what does this function do?



So, what does this function do?

Microsoft | Docs Documentation Learn Q&A Code Samples

Windows Developer Explore Platforms Downloads Samples Support

Windows / Apps / Win32 / Desktop Technologies / Audio and Video / DirectShow

Filter by title

- DirectShow
- Introduction to DirectShow
  - Introduction to DirectShow
  - What's New in DirectShow
  - Supported Formats in DirectShow
  - DirectShow FAQ
- Getting Started
- About DirectShow

Introduction to DirectShow

05/31/2018 • 2 minutes to read •

Microsoft® DirectShow® is an architecture for streaming media on the Microsoft Windows® platform. DirectShow provides for high-quality capture and playback of multimedia streams. It supports a wide variety of formats, including Advanced Systems Format (ASF), Motion Picture Experts Group (MPEG), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3), and WAV sound files. It supports capture from digital and analog devices based on the Windows Driver Model (WDM) or Video for Windows. It automatically detects and uses video and audio acceleration hardware when available, but also supports systems without acceleration hardware.

DirectShow is based on the Component Object Model (COM). To write a DirectShow application or component, you must understand COM client programming. For most applications, you do not need to implement your own COM objects. DirectShow provides the components you need. If you want to extend DirectShow by writing your own components, however, you must implement them as COM objects.

DirectShow is designed for C++. Microsoft does not provide a managed API for DirectShow.

DirectShow simplifies media playback, format conversion, and capture tasks. At the same time, it provides access to the underlying stream control architecture for applications that require custom solutions. You can also create your own DirectShow components to support new formats or custom effects.

Examples of the types of application you can write with DirectShow include file players, TV and DVD players, video editing applications, file format converters, audio-video capture applications, encoders and decoders, digital signal processors, and more.

So, what is this used for?

InprocServer32

05/31/2018 • 2 minutes to read •

Registers a 32-bit in-process server and specifies the threading model of the apartment the server can run in.

Event Triggered Execution: Component Object Model Hijacking

Other sub-techniques of Event Triggered Execution (15)

Adversaries may establish persistence by executing malicious content triggered by hijacked references to Component Object Model (COM) objects. COM is a system within Windows to enable interaction between software components through the operating system.[1] References to various COM objects are stored in the Registry.

Adversaries can use the COM system to insert malicious code that can be executed in place of legitimate software through hijacking the COM references and relationships as a means for persistence. Hijacking a COM object requires a change in the Registry to replace a reference to a legitimate system component which may cause that component to not work when executed. When that system component is executed through normal system operation the adversary's code will be executed instead.[2] An adversary is likely to hijack objects that are used frequently enough to maintain a consistent level of persistence, but are unlikely to break noticeable functionality within the system as to avoid system instability that could lead to detection.

ID: T1546.015

Sub-technique of: T1546

ⓘ Tactics: Privilege Escalation, Persistence

ⓘ Platforms: Windows

ⓘ Permissions Required: User

ⓘ Data Sources: Command: Command Execution, Module: Module Load, Process: Process Creation, Windows Registry: Windows Registry Key Modification

Contributors: Elastic

Version: 1.0

Created: 16 March 2020

Last Modified: 10 November 2020

Family	ClassSID	Class	SubClass	
agent_tesla	{9d6aa569-9f30-41ad-885a-346685c74928}	inprocserver32	threadingmodel	Google Toolbar Helper
agent_tesla	{9d6aa569-9f30-41ad-885a-346685c74928}	inprocserver32	(default)	
agent_tesla	{4ec3c18e-7203-41e7-990d-a72b57e286a9}	inprocserver32	(default)	
agent_tesla	{4ec3c18e-7203-41e7-990d-a72b57e286a9}	inprocserver32	threadingmodel	Google Update
agent_tesla	{4ec3c18e-7203-41e7-990d-a72b57e286a9}	inprocserver32	(default)	
agent_tesla	{c6271107-a214-4f11-98c0-3f16bc670d28}	inprocserver32	threadingmodel	
agent_tesla	{c6271107-a214-4f11-98c0-3f16bc670d28}	inprocserver32	threadingmodel	Google Update
agent_tesla	{c6271107-a214-4f11-98c0-3f16bc670d28}	inprocserver32	(default)	
agent_tesla	{c6271107-a214-4f11-98c0-3f16bc670d28}	inprocserver32	(default)	

So, what is this used for?

Check all scheduled tasks on the system

ClassSID	ClassName	Family
{0f87369f-a4e5-4cfc-bd3e-73e6154572dd}	CLSID TaskScheduler class	agent_tesla
{0f87369f-a4e5-4cfc-bd3e-73e6154572dd}	CLSID TaskScheduler class	trickbot

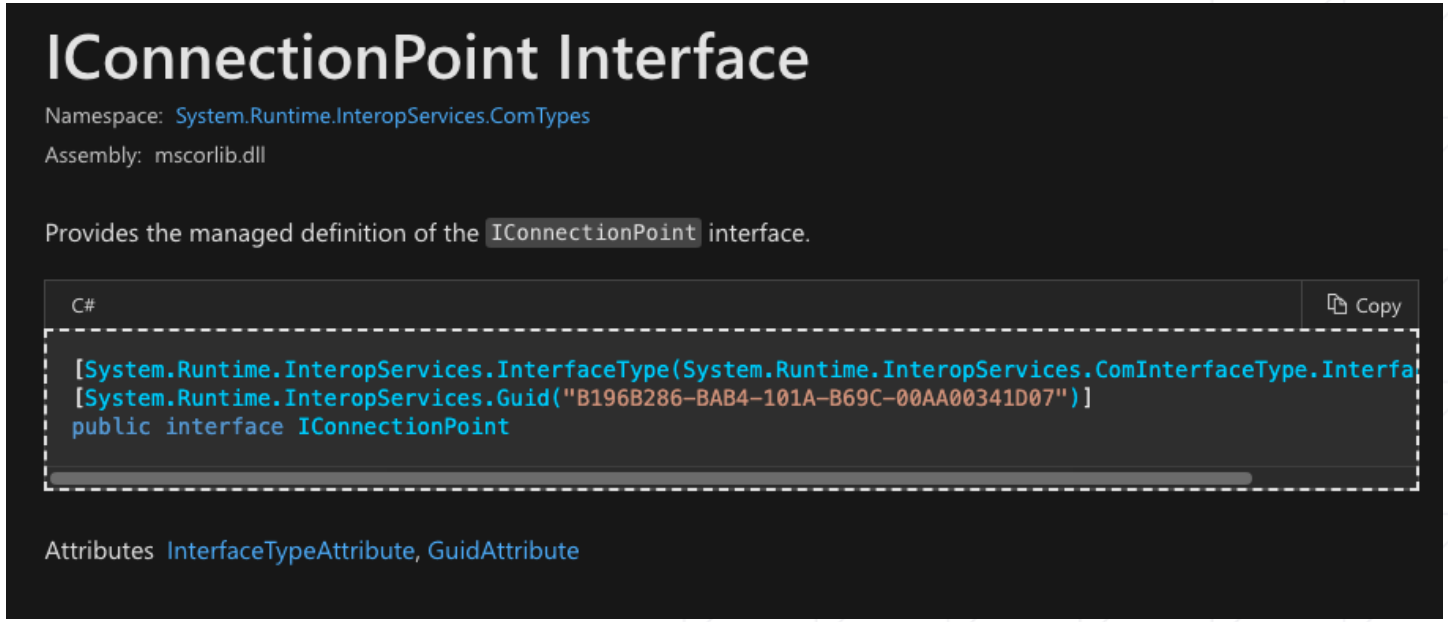
ClassSID ↑	ClassName	Family
{20d04fe0-3aea-1069-a2d8-08002b30309d}		dridex
{20d04fe0-3aea-1069-a2d8-08002b30309d}		trickbot
{20d04fe0-3aea-1069-a2d8-08002b30309d}		qakbot
{20d04fe0-3aea-1069-a2d8-08002b30309d}		agent_tesla
{217fc9c0-3aea-1069-a2db-08002b30309d}		agent_tesla
{21b22460-3aea-1069-a2dc-08002b30309d}		agent_tesla

So, what is this used for?

>	{b196b286-bab4-101a-b69c-00aa00341d07}	trickbot
>	{b196b286-bab4-101a-b69c-00aa00341d07}	qakbot
>	{b196b286-bab4-101a-b69c-00aa00341d07}	agent_tesla
>	{b196b286-bab4-101a-b69c-00aa00341d07}	dridex

They can create or re-use a listener that will execute commands sent to that interface.

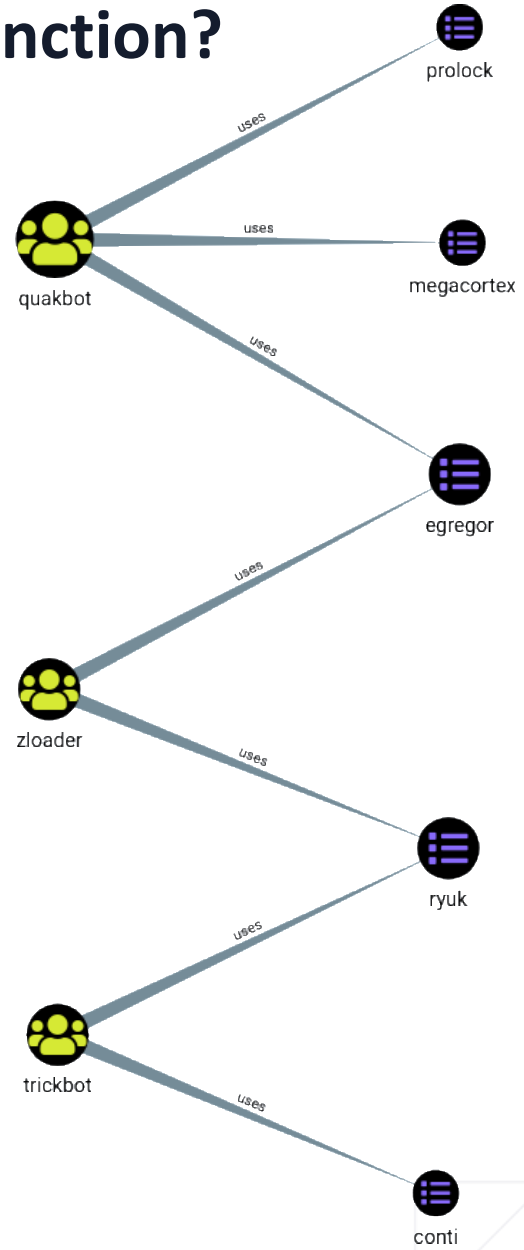
Some use the older soon to be deprecated interface **UCOMIConnectionPoint**

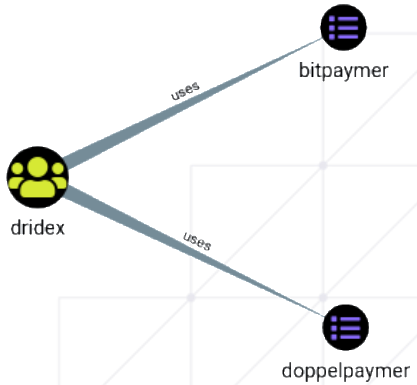


Who uses that DllRegisterServer function?

This is a relatively old and obscure function so, there are some remaining questions:

- Is someone developing parts for all these families?
- Maybe code is shared/sold on underground markets that is reused?
- Are these crews reversing each other and stealing tricks from each other ?





Fun find

Malware writers with an unintended sense of humor?

FileName	Family	count_
bears.jpg.exe	trickbot	37
bears.htm.exe	trickbot	44

Summarizing

Sandbox telemetry:

- … is great for researching and developing new detections based on actual behavior
- … can be used as a ‘does this actually happen’ reference
- … is not complete, lots of malware actively tries to detect it
- … lacks the data from where an attacker might go interactive

But also:

- Focus on building custom detection on patterns and behavior
- ATT&CK is amazing for guidance and a common language, it’s a lot of work to maintain and never complete.
- Try to build mitigations and detection rules for the things that slip by or the mitigation

Thank you!

Olaf Hartong

@olafhartong
github.com/olafhartong
olaf@falconforce.nl
olafhartong.nl / falconforce.nl