Comparing Malicious Files

RVAsec
May 22, 2019

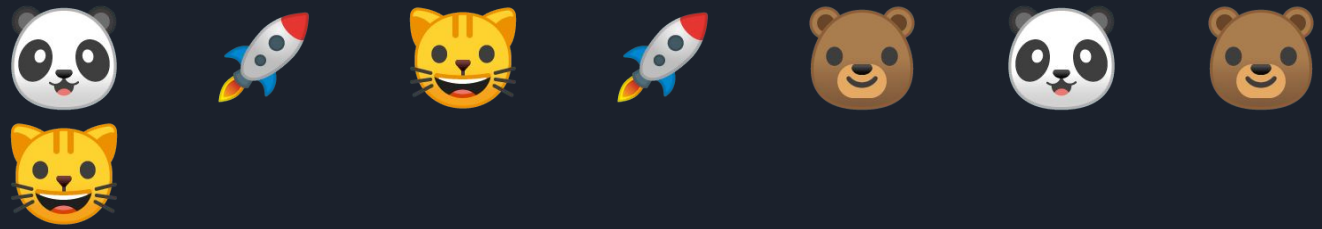
Problem Statements

AV Problem

Many AV companies use their own unique nomenclature for malware and malware families

Marketing Problem

Marketing departments want to brand the malware families that their company has identified



WTF??????

- APT28
- Pawn Storm
- Fancy Bear
- Sednit
- TsarTeam
- TG-4127
- Group-4127
- STRONTIUM
- TAG_0700
- Swallowtail
- IRON TWILIGHT
- Group 74

Missing Criteria

Robert M. Lee @RobertMLee · Feb 6

Intel Analyst: We have unique requirements, collection, & analysis. We name our cluster of intrusions $UniqueName

Twitter: OMG ANOTHER NAME WTH?!

Intel Analyst: Hey we're back and we're tracking this activity as $KnownName

Twitter: OMG THATS NOT HOW WE TRACK IT YOU'RE WRONG!!

2 9 47

Robert M. Lee @RobertMLee · Feb 6

This isn't a subtweet just salty about naming conventions this morning. I don't agree RecordedFuture or anyone should use others' naming conventions. The right approach is defining your own. But it will also lead to critics. So support your consumers and just don't worry about it

11 24

Researcher's Problem

What am I looking at?

Can I relate this to other samples that have already been identified?

Is this a new attack?

Incident Responder's Problem

What is this related to?

Can I locate previous work around this malware, so I can save time?

Solution Methods

Sample Identification

Determine malware family membership of sample

Locating Associated Samples

Within a set of samples, which are related?

Identification Method: Anti-Virus Scanner Results

Shared Engines

Sample: 68119dd7fb9ecb099de50227162bd82f

Scanner Result: Trojan.GenericKD.40437487

AV Companies: Ad-Aware, ALYac, BitDefender, Emsisoft, F-Secure, GData, MicroWorld-eScan

Development Methods

http://who-really-cares-anyway.blogspot.com/2007/03/generic-food.html

Generic

http://www.beerdestroyer.com/wp-content/uploads/2013/05/dc_brau_corruption.jpg

Specific

Vendors with Usable Results

Microsoft

https://www.microsoft.com/en-us/wdsi/threats

ESET

http://www.virusradar.com/en/threat_encyclopaedia

Kaspersky

https://encyclopedia.kaspersky.com

Sophos

https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware.aspx

Boiling Down Results

Sample: c3f9d80d11ab3671cd412e94de4141ad

Boiling Down Results

Remove clearly generic results

Watch for sneaky generic results: Zeus, Zbot, Zusy, etc.

Boiling Down Results

ESET-NOD32	Win32/Adware.VrBrot	Yandex	Trojan.DR.PeBundle.A
McAfee	Artemis!C3F9D80D11AB	VBA32	Trojan.Isbar
Ad-Aware	Gen:Variant.Symmi.89546	DrWeb	Trojan.Isbar.863
ALYac	Gen:Variant.Symmi.89546	Arcabit	Trojan.Symmi.D15DCA
BitDefender	Gen:Variant.Symmi.89546	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim
F-Secure	Gen:Variant.Symmi.89546	ViRobot	Trojan.Win32.Z.Symmi.954946
MicroWorld-eScan	Gen:Variant.Symmi.89546	Avast	Win32:VrBrothers-A [Adw]
Emsisoft	Gen:Variant.Symmi.89546 (B)	AVG	Win32:VrBrothers-A [Adw]
Ikarus	Trojan-Spy.Win32.Sincom	GData	Win32.Backdoor.Hupigon.B
Microsoft	Trojan:Win32/Occamy.C	Kingsoft	Win32.RiskWare.PEBundle.49152

Boiling Down Results

ESET-NOD32	Win32/Adware.VrBrot	Yandex	Trojan.DR.PeBundle.A
McAfee	Artemis!C3F9D80D11AB	VBA32	Trojan.Isbar
Ad-Aware	Gen:Variant.Symmi.89546	DrWeb	Trojan.Isbar.863
ALYac	Gen:Variant.Symmi.89546	Arcabit	Trojan.Symmi.D15DCA
BitDefender	Gen:Variant.Symmi.89546	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim
F-Secure	Gen:Variant.Symmi.89546	ViRobot	Trojan.Win32.Z.Symmi.954946
MicroWorld-eScan	Gen:Variant.Symmi.89546	Avast	Win32:VrBrothers-A [Adw]
Emsisoft	Gen:Variant.Symmi.89546 (B)	AVG	Win32:VrBrothers-A [Adw]
Ikarus	Trojan-Spy.Win32.Sincom	GData	Win32.Backdoor.Hupigon.B
Microsoft	Trojan:Win32/Occamy.C	Kingsoft	Win32.RiskWare.PEBundle.49152

Boiling Down Results

ESET-NOD32	Win32/Adware.VrBrot	Yandex	Trojan.DR.PeBundle.A
McAfee	Artemis!C3F9D80D11AB	VBA32	Trojan.Isbar
Ad-Aware	Gen:Variant.Symmi.89546	DrWeb	Trojan.Isbar.863
ALYac	Gen:Variant.Symmi.89546	Arcabit	Trojan.Symmi.D15DCA
BitDefender	Gen:Variant.Symmi.89546	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim
F-Secure	Gen:Variant.Symmi.89546	ViRobot	Trojan.Win32.Z.Symmi.954946
MicroWorld-eScan	Gen:Variant.Symmi.89546	Avast	Win32:VrBrothers-A [Adw]
Emsisoft	Gen:Variant.Symmi.89546 (B)	AVG	Win32:VrBrothers-A [Adw]
Ikarus	Trojan-Spy.Win32.Sincom	GData	Win32.Backdoor.Hupigon.B
Microsoft	Trojan:Win32/Occamy.C	Kingsoft	Win32.RiskWare.PEBundle.49152

Boiling Down Results

Ad-Aware	Gen:Variant.Symmi.89546	VBA32	Trojan.Isbar
Arcabit	Trojan.Symmi.D15DCA	DrWeb	Trojan.Isbar.863
ViRobot	Trojan.Win32.Z.Symmi.954946	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim

Win32.Trojan.Symmi

Win32.Trojan.Isbar

Boiling Down Results

Ad-Aware	Gen:Variant.Symmi.89546	VBA32	Trojan.Isbar
Arcabit	Trojan.Symmi.D15DCA	DrWeb	Trojan.Isbar.863
ViRobot	Trojan.Win32.Z.Symmi.954946	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim

Win32.Trojan.Symmi

Win32.Trojan.Isbar

Boiling Down Results

Ad-Aware	Gen:Variant.Symmi.89546	VBA32	Trojan.Isbar
Arcabit	Trojan.Symmi.D15DCA	DrWeb	Trojan.Isbar.863
ViRobot	Trojan.Win32.Z.Symmi.954946	NANO-Antivirus	Trojan.Win32.Isbar.fhgjim

Win32.Trojan.Symmi

Win32.Trojan.Isbar

Automation: AVClass

- Family Rankings
- PUP Classification
- Ground Truth Evaluation
- Generic Token Detection
- Alias Detection

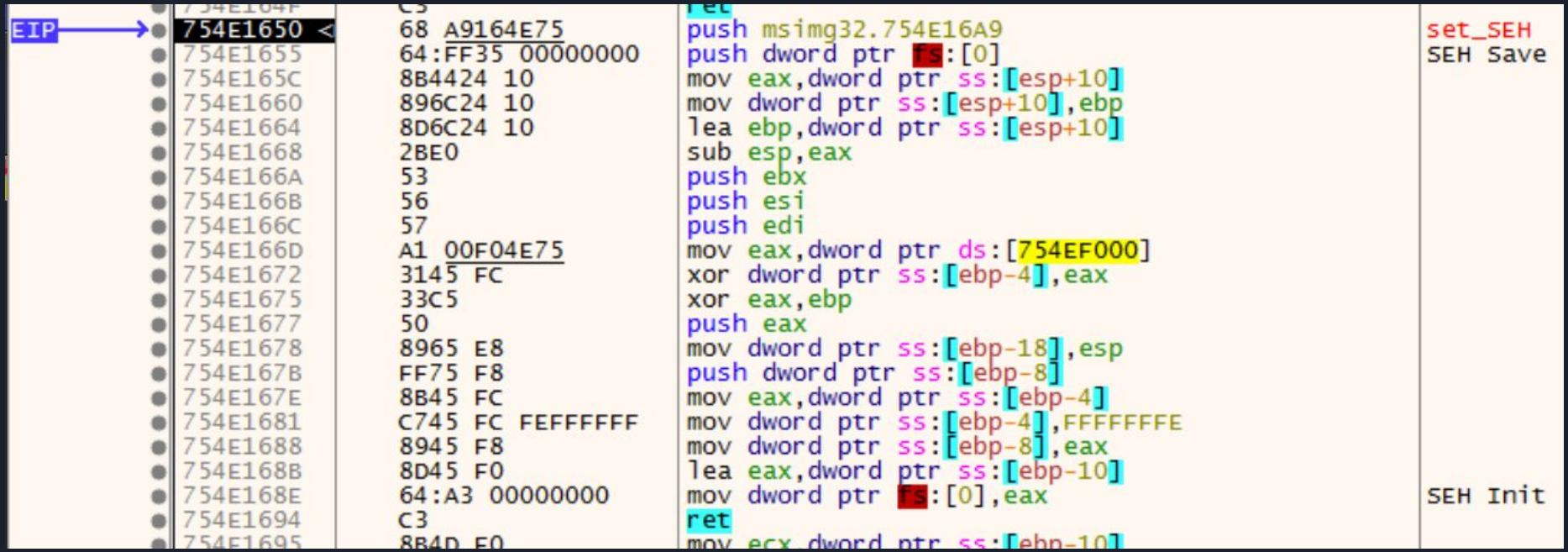
https://github.com/malicialab/avclass

Identification Method: MITRE ATT&CK

ATT&CK

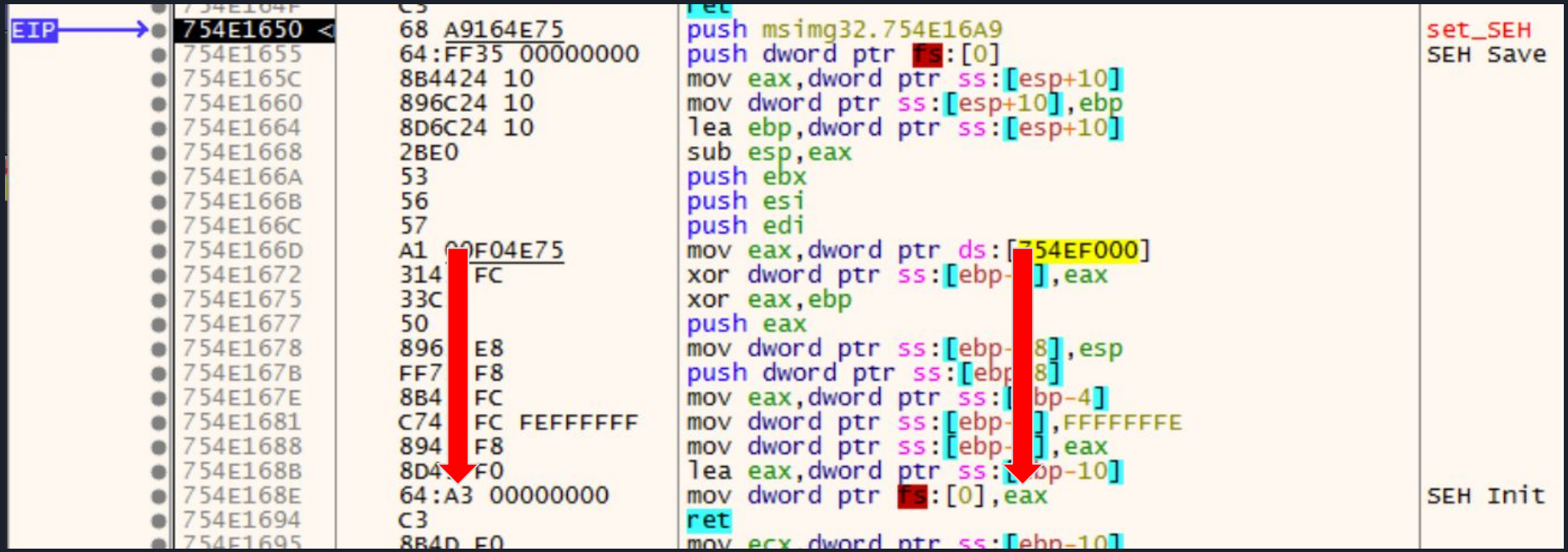
- Framework for categorization of adversary tactics and techniques
- Excellent first step
- Not yet ready for malware classification
- There is a better option!

ATT&CK & Granularity



https://steemit.com/reverseengineering/@utkonos/alphablend-campaign-part-2

ATT&CK & Granularity



SEH Variation

```
rule SEH_Init : Tactic_DefensiveEvasion Technique_AntiDebugging SubTechnique_SEH
{
    meta:
        author = "Malware Utkonos"
        original_author = "naxonez"
        source = "https://github.com/naxonez/yaraRules/blob/master/AntiDebugging.yara"
    strings:
        $a = { 64 A3 00 00 00 00 }
        $b = { 64 89 25 00 00 00 00 }
    condition:
        WindowsPE and ($a or $b)
}
```

Contribute Sub-Techniques

https://attack.mitre.org/resources/contribute/

2FA Interception (T1111)

- SMS interception on the wire (SORM)
- SMS interception by number porting
- Code interception via phishing page (Nile Phish, Charming Kitten)
- Keylogger

Better System

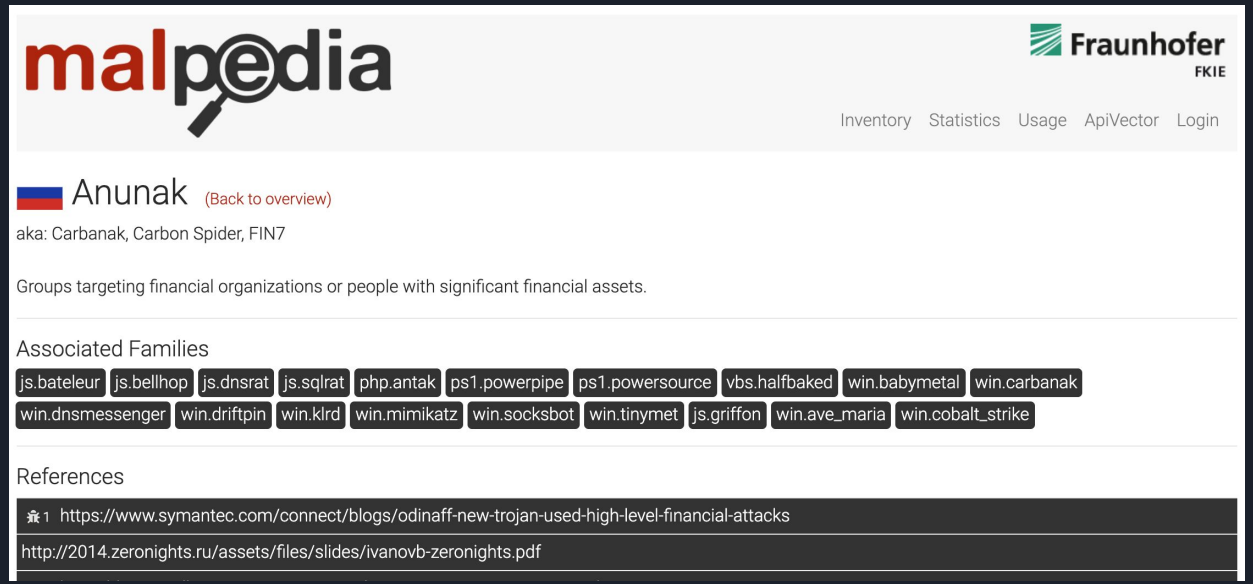
The New MAEC

Anti-Behavioral Analysis	Execution
Anti-Static Analysis	Exfiltration
Collection	Impact
Command and Control	Lateral Movement
Credential Access	Persistence
Defense Evasion	Privilege Escalation
Discovery	

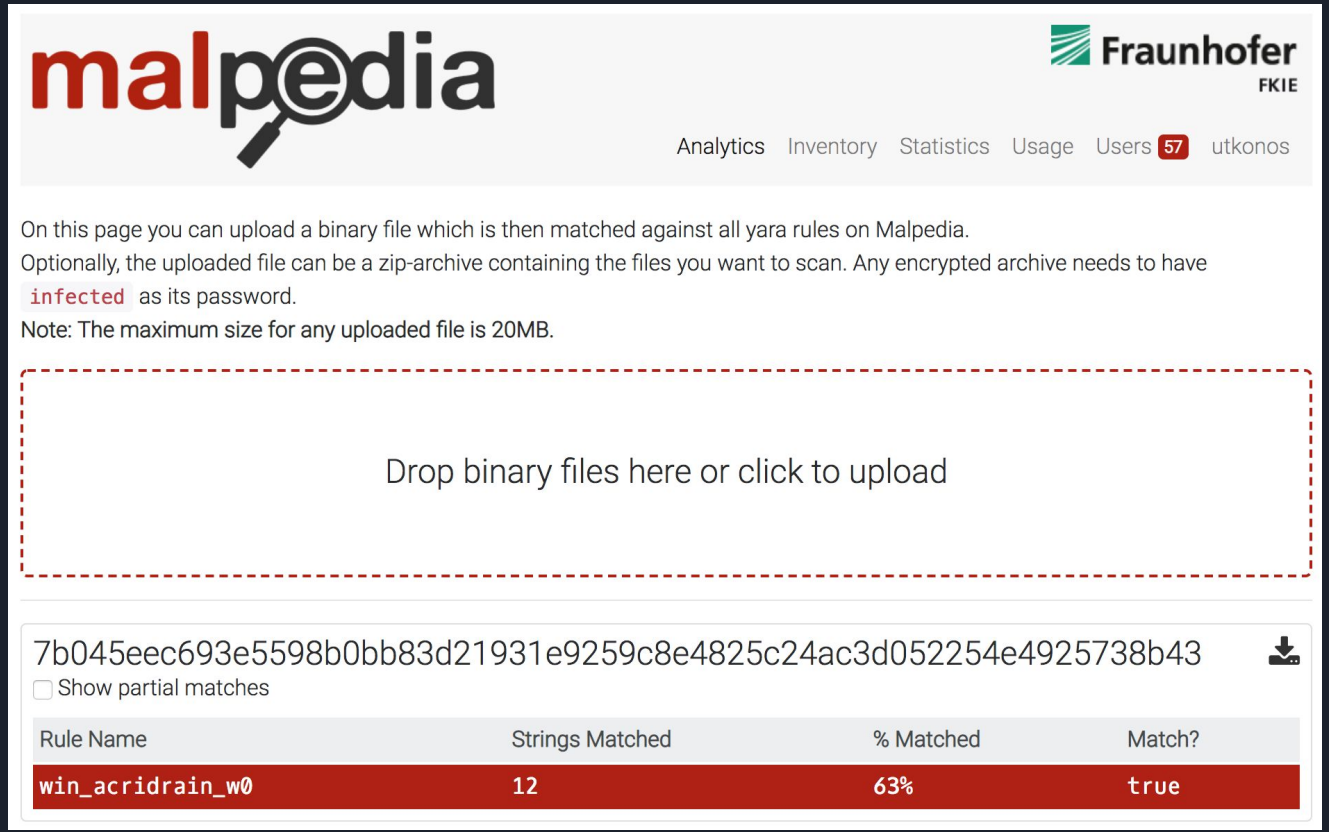
https://github.com/MAECProject/malware-behaviors

Identification Method: Malpedia

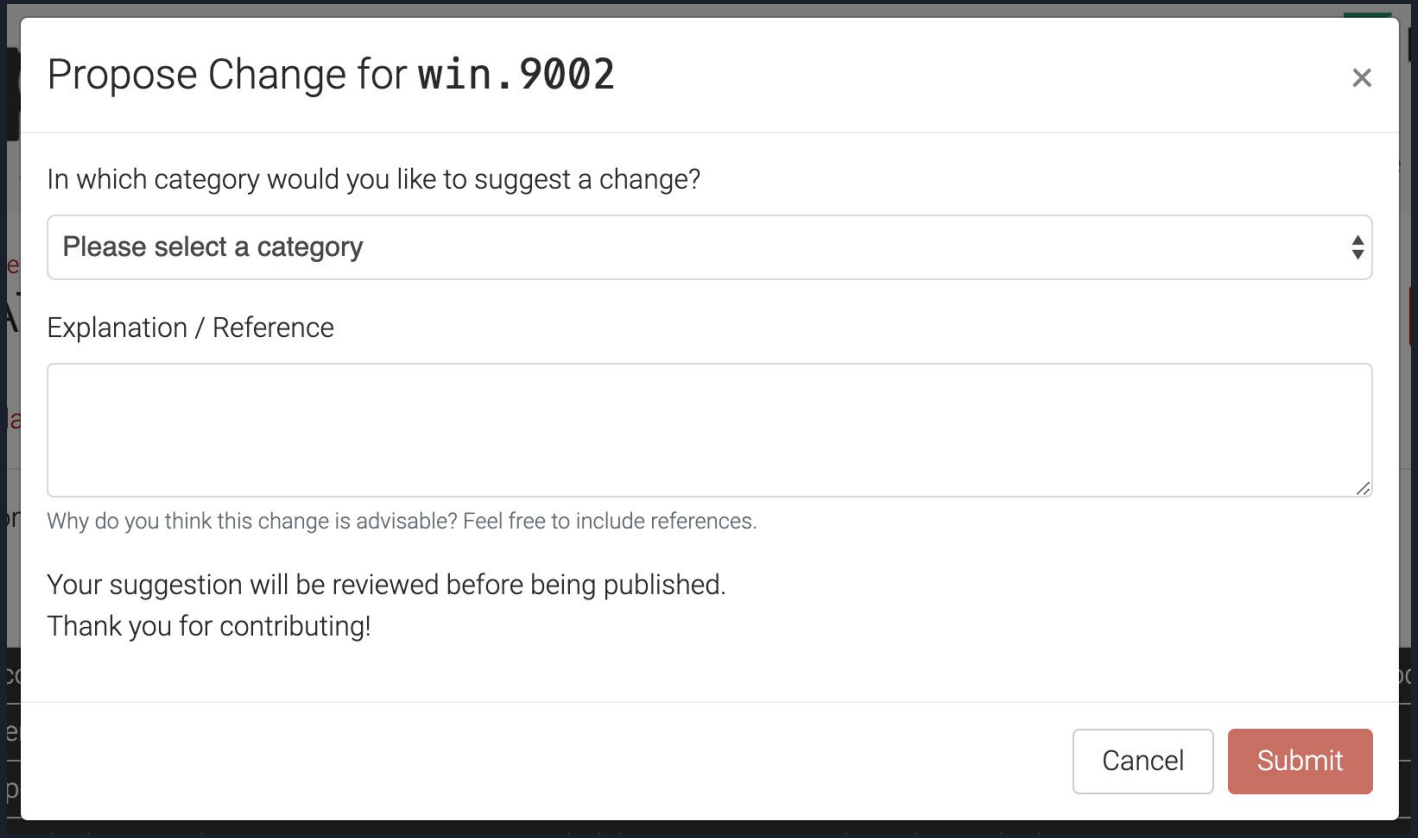
Malpedia: FIN7, Carbanak



https://malpedia.caad.fkie.fraunhofer.de/actor/anunak

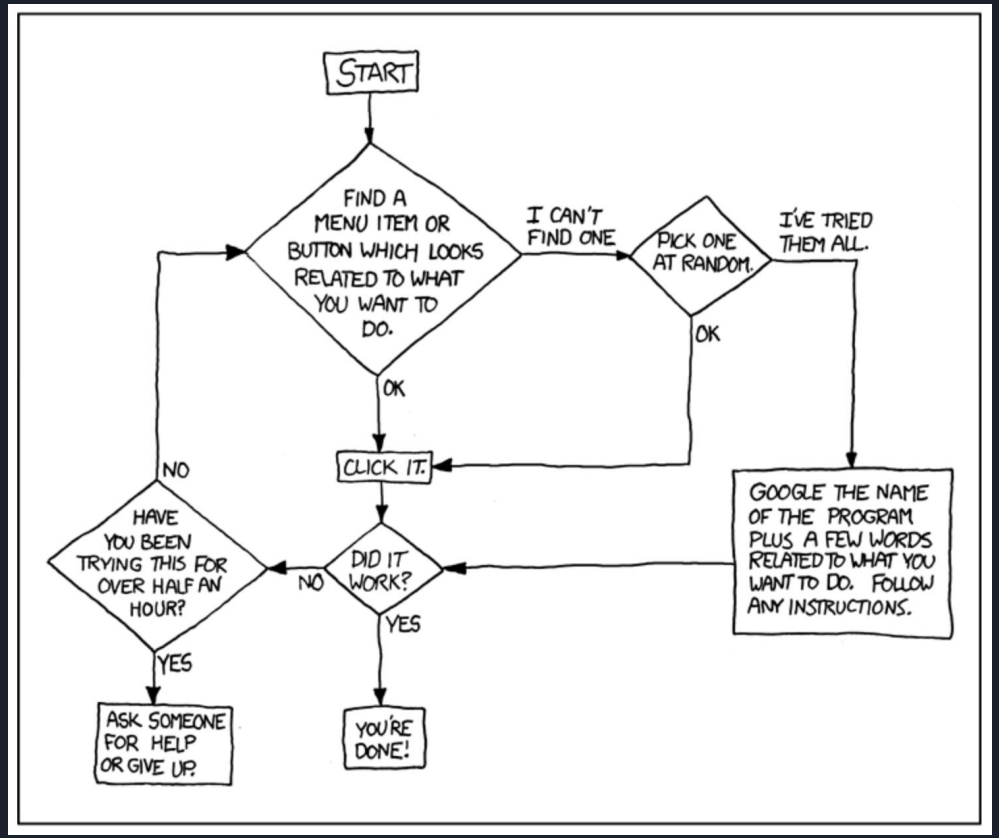


Malpedia Results

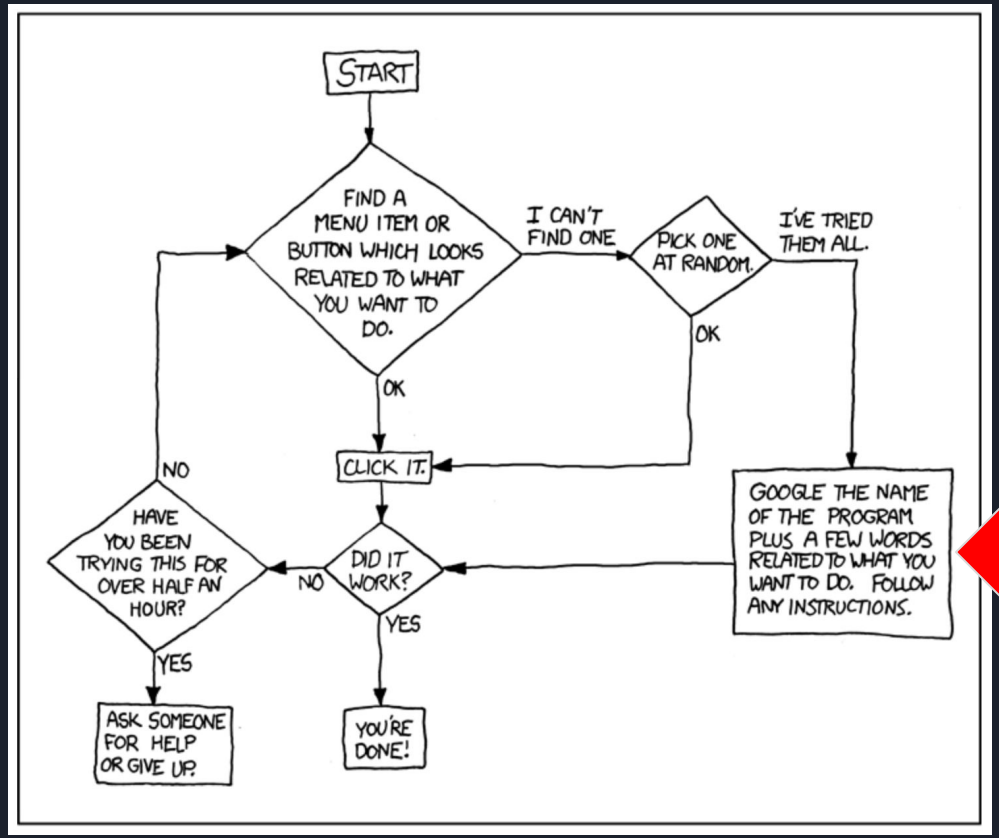

Propose Change for win.9002
In which category would you like to suggest a change?
Please select a category
Explanation / Reference
Why do you think this change is advisable? Feel free to include references.
Your suggestion will be reviewed before being published.
Thank you for contributing!
Cancel
Submit

Contribute!!!!!

Identification Method: Google



https://xkcd.com/627/



https://xkcd.com/627/

Proposal

~~Proposal~~

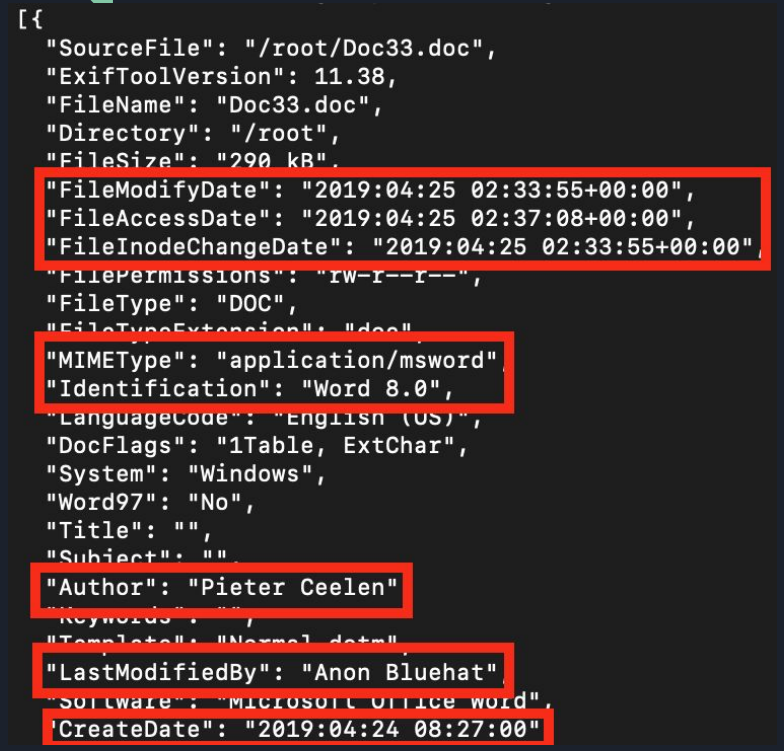
Association Method: Static Analysis

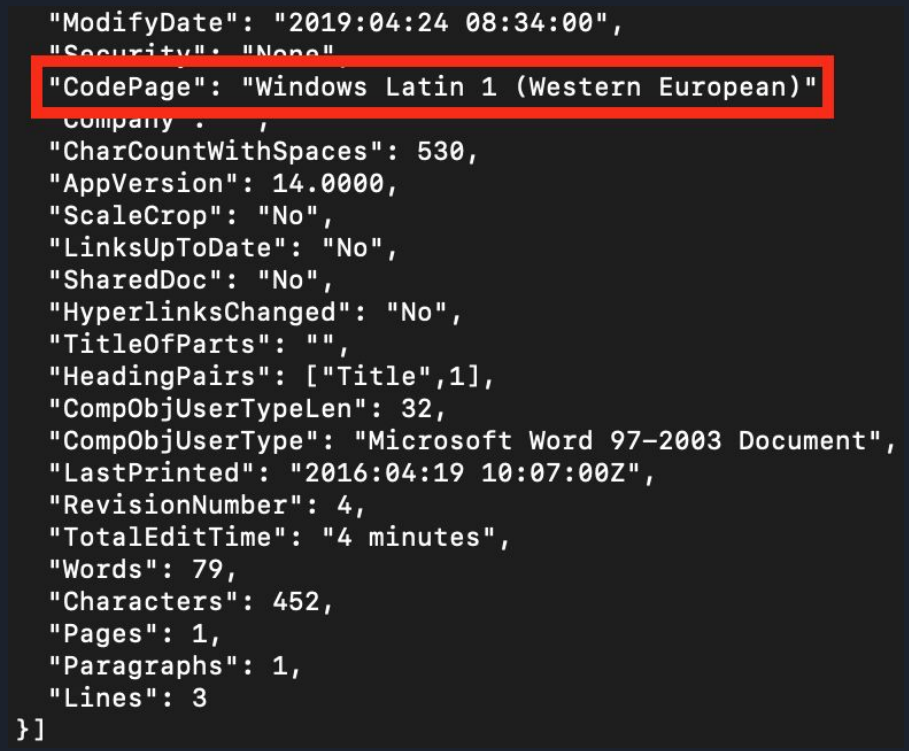
Some Hashes

ssdeep: Context triggered piecewise hash

Import Hash (imphash): Calculated from PE file import table

Exif Metadata





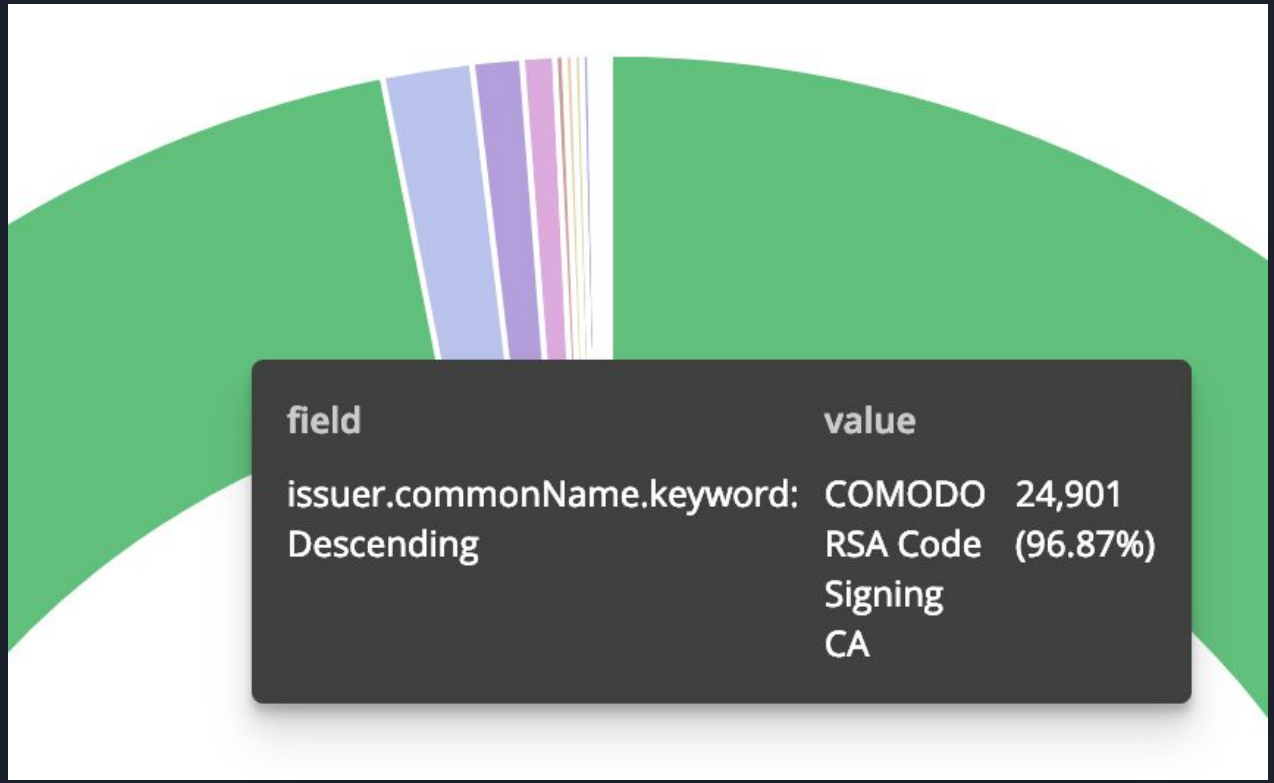
Code Signing Certificate

Signed by fake cert

Signed by real/stolen cert

Signed-ish: broken signature

Abused Certificates



PE Metadata

Sections

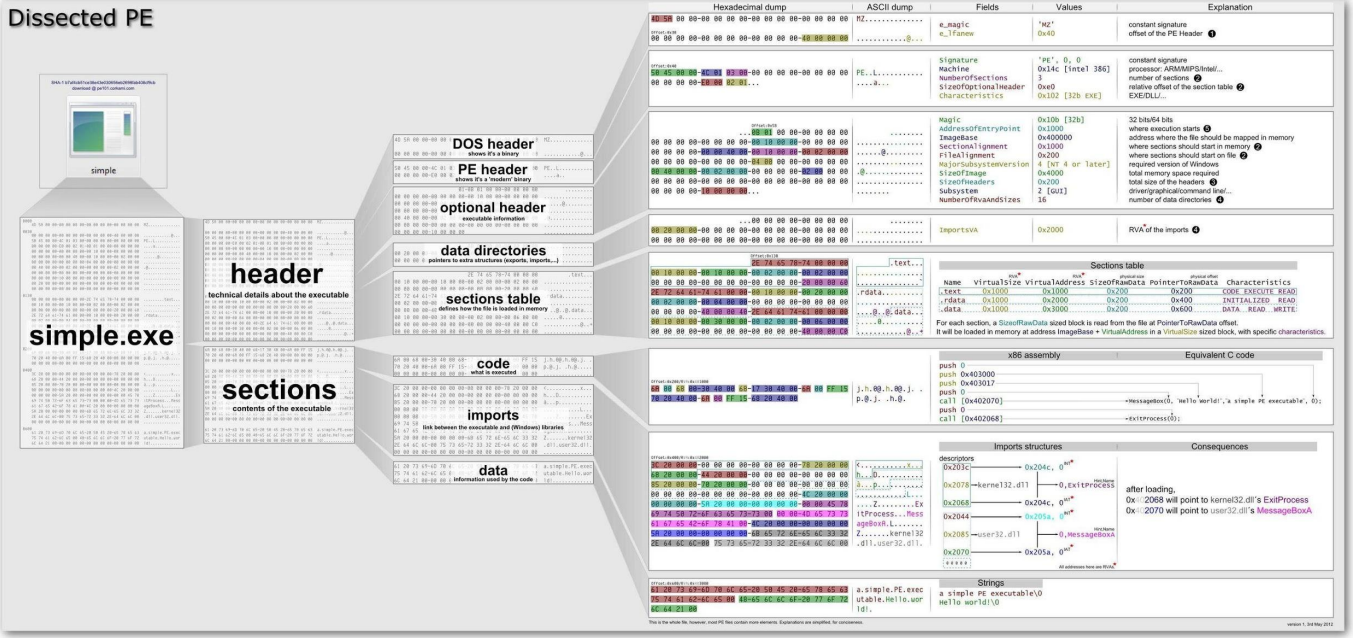
Imports / Exports

Resources

PE101 a windows executable walkthrough

Ange Albertini
corkami.com

Dissected PE

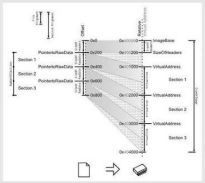


Loading process

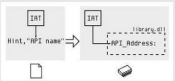
1 Headers
the *DOS Header* is parsed
the *PE Header* is parsed
the offset is *DOS Header*'s e_flanew)
the *Optional Header* is parsed
(it follows the *PE Header*)

2 Sections table
Sections table is parsed
(it is located at: offset (*OptionalHeader*) + *SizeOfOptionalHeader*)
it contains *NumberOfSections* elements
it is checked for validity with alignments:
FileAlignments and *SectionAlignments*

3 Mapping
the file is mapped in memory according to:
the *ImageBase*
the *SizeOfHeaders*
the Sections table



4 Imports
DataDirectories are parsed
they follow the *OptionalHeader*
their number is *NumOfRVAAndSizes*
imports are always #2
Imports are parsed
each *descriptor* specifies a *DLLname*
this DLL is loaded in memory
IAT and *INT* are parsed simultaneously
for each API in *INT*
its address is written in the *IAT* entry



5 Execution
Code is called at the *EntryPoint*
the calls of the code go via the IAT to the APIs

Notes

MZ HEADER *aka* DOS_HEADER
Starts with 'MZ' (initials of *Mark Zbikowski* MS-DOS developer)

PE HEADER *aka* IMAGE_FILE_HEADERS / COFF file header
Starts with 'PE' (Portable Executable)

OPTIONAL HEADER *aka* IMAGE_OPTIONAL_HEADER
Optional only for non-standard PEs but required for executables

RVA Relative Virtual Address
Address relative to ImageBase (at ImageBase, RVA = 0)
Almost all addresses of the headers are RVAs
In code, addresses are *not* relative.

INT Import Name Table
Null-terminated list of pointers to Hint, Name structures

IAT Import Address Table
Null-terminated list of pointers
On file it is a copy of the INT
After loading it points to the imported APIs

HINT
Index in the exports table of a DLL to be imported
Not required but provides a speed-up by reducing look-up

Sections

Sample: 0a9545f9fc7a6d8596cf07a59f400fd3

Name: .reloc

MD5: 3a64e2292f5eb1bbe70428c1c6ee22d5

Sections

Sample: 0a9545f9fc7a6d8596cf07a59f400fd3

Name: .reloc

MD5: 3a64e2292f5eb1bbe70428c1c6ee22d5

Resources

Sample: c7577748e6e7c71cdf5a950655b2456e

Name: RT_VERSION

SHA256:
4df4bf2f6de1beb10586f49b4155fffb946279e8b0
a69d6fbbe695158bbb63ae

ReversingLabs Hash Algorithm

https://www.reversinglabs.com/technology/reversinglabs-hash-algorithm.html

VirusTotal similar-to:

Proprietary black magic, but very effective

Document Metadata

Author

Timestamps

Language

PDF Producer

Association Method: Dynamic Analysis

Filenames

Boring: finding exactly the same filename

More exciting: develop regex for a pattern of generated filenames.

URL Structure: Download

Related to the vulnerability in the CMS that was exploited to create the URL

URL Structure: Download

Example:

http://terumoindonesia.com/wp-content/themes/twentysixteen/

Regex:

wp-[a-z]+/themes/twenty(?:ten|eleven|twelve|thirteen|fourteen|fifteen|sixteen|seventeen|eighteen)

URL Structure: C2

Directly related to the malware family

URL Structure: C2

Example:

http://dinttobogo.com/zapoy/gate.php

Mutual Exclusion (Mutex)

Prevents race conditions with

multiple processes and multiple threads.

https://en.wikipedia.org/wiki/Mutual_exclusion

Registry Key

Hierarchical database for

low-level OS and application settings.

https://en.wikipedia.org/wiki/Windows_Registry

Association Method: Clustering Algorithms

Standing on Shoulders of Giants

“Python and Machine Learning: How to clusterize a malware dataset?”

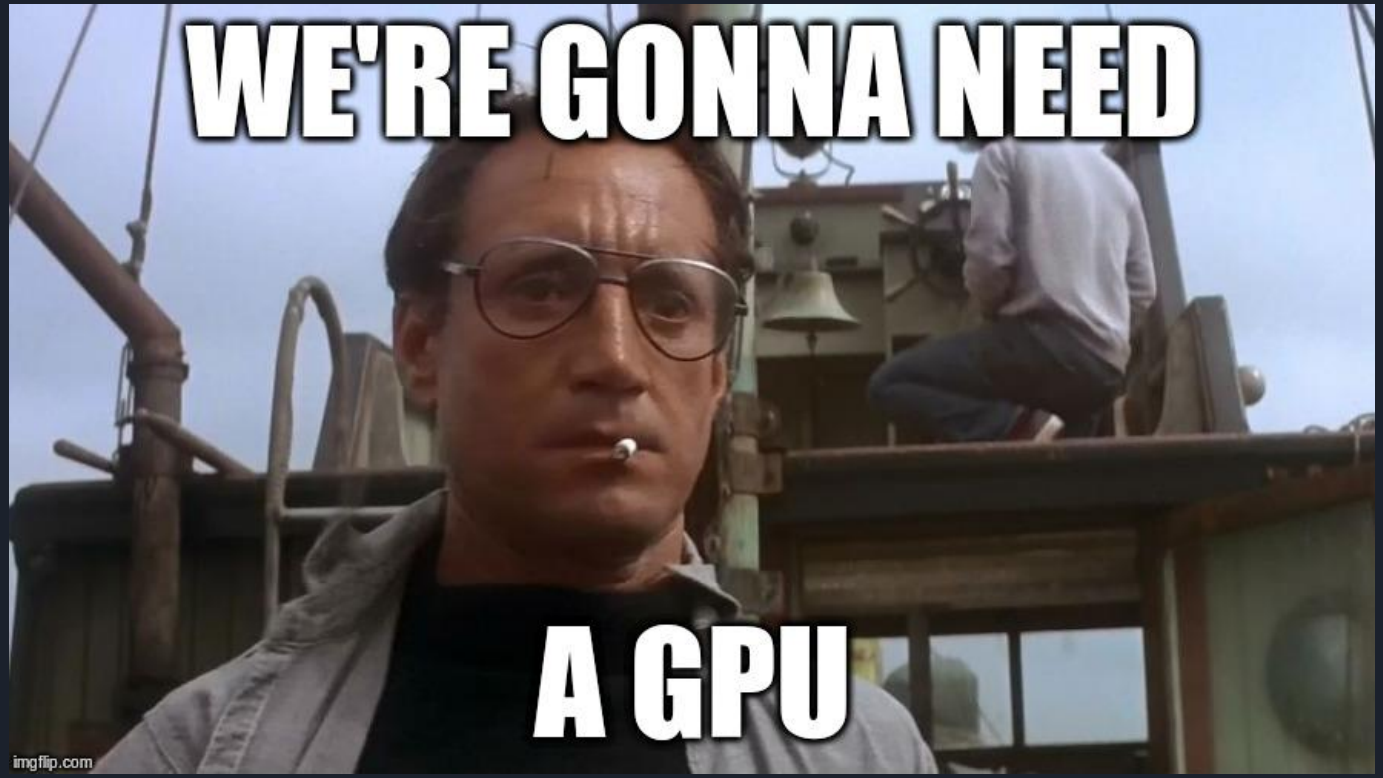
https://github.com/sebdraven/hack_lu_2017

And botconf!

Algorithms

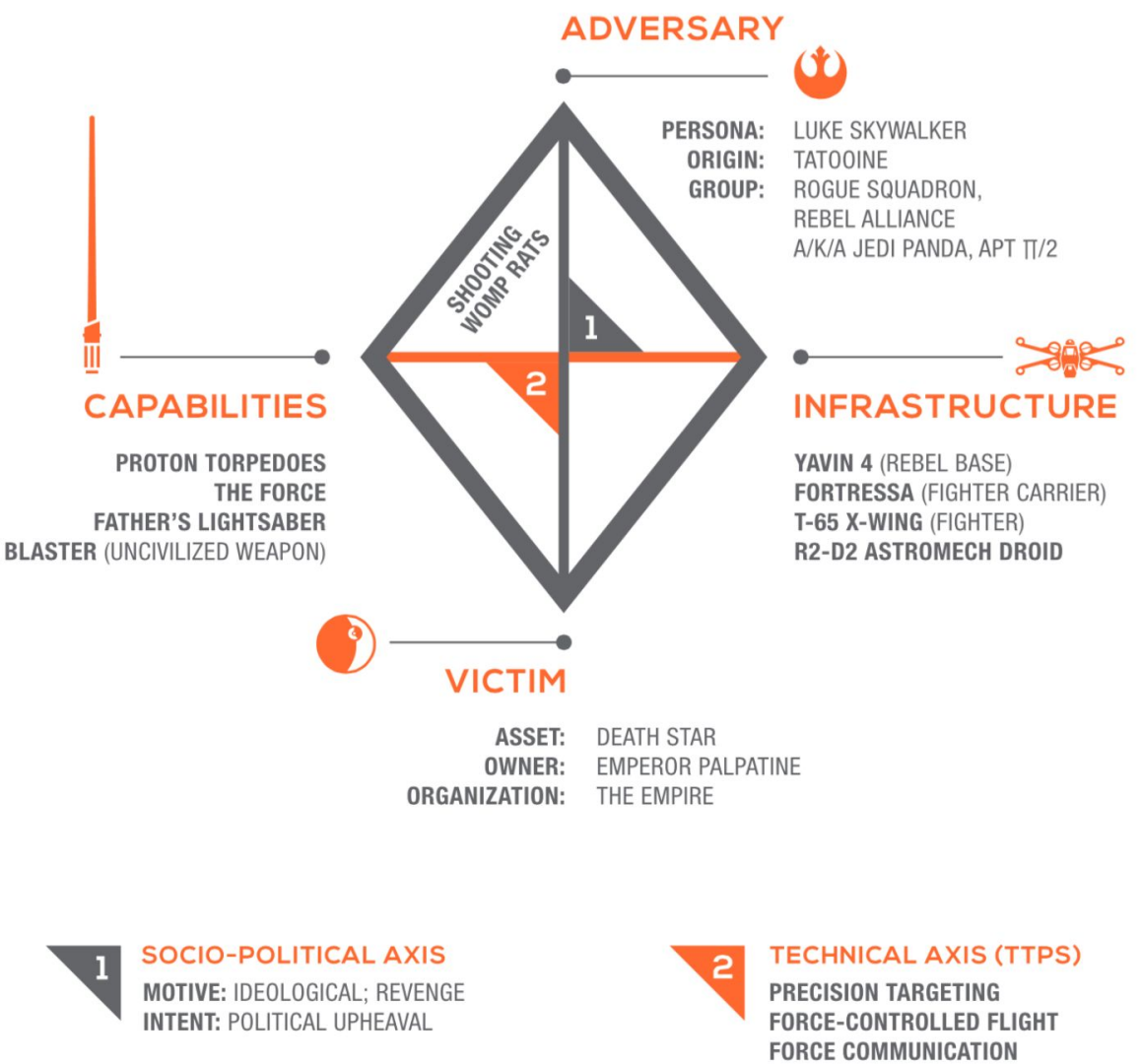
K-Means

DBScan



https://thescinder.files.wordpress.com/2017/06/goingtoneedagpuimgflip1.jpg

Association Method: Diamond Model of Intrusion Analysis



Diamond Model

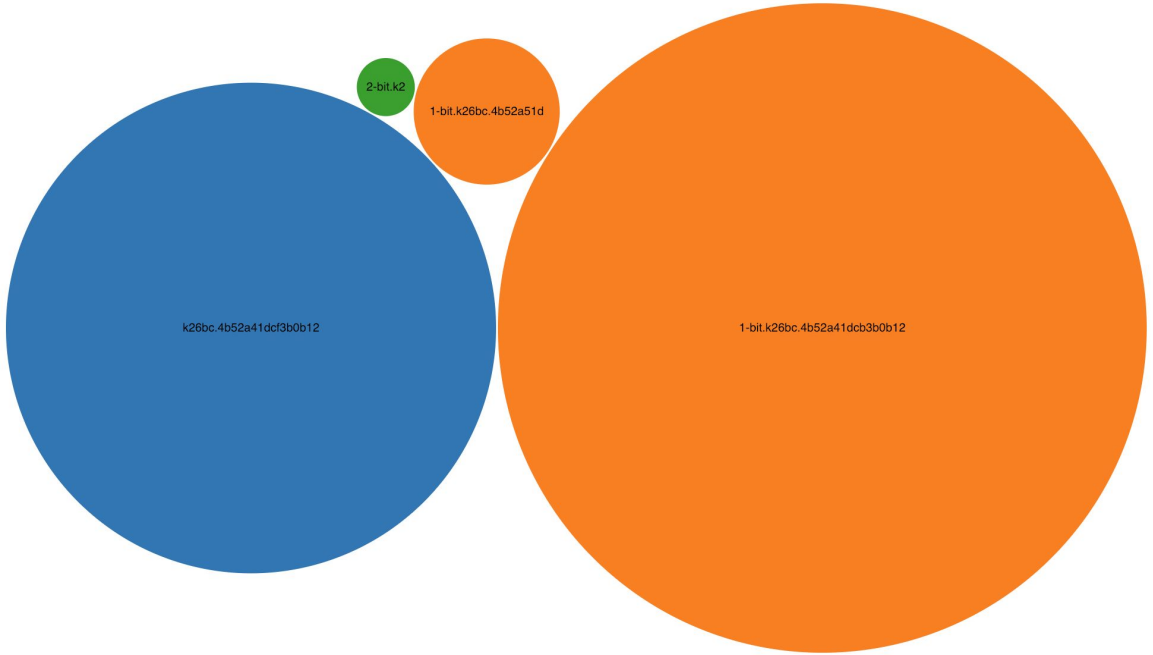
http://www.dtic.mil/docs/citations/ADA586960

Association Method: Icewater

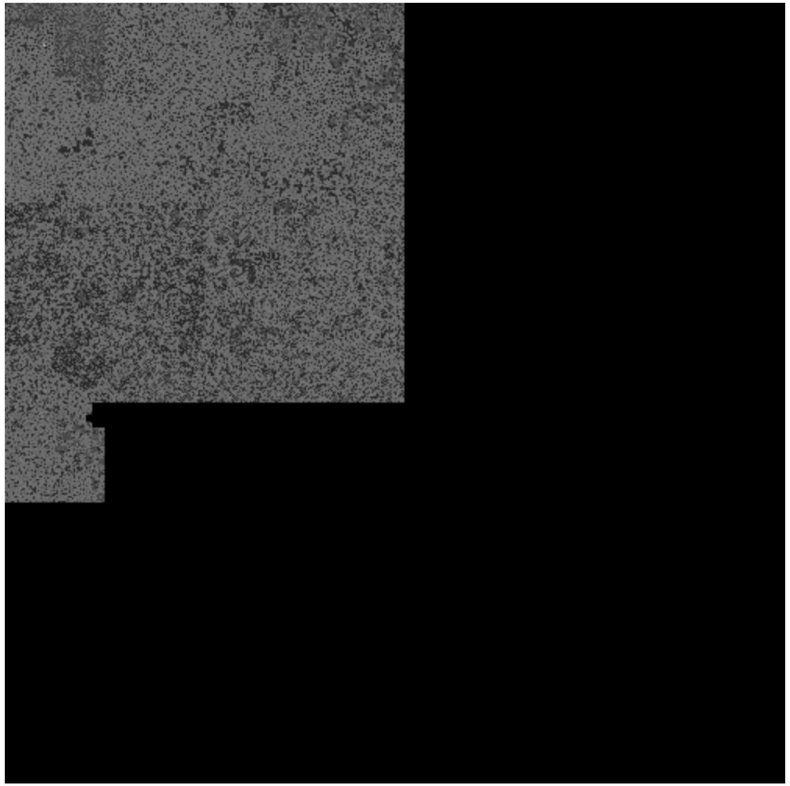
Icewater

http://icewater.io/search

Cluster Neighbors for k26bc.4b52a41dcf3b0b12



124a591871e93e0272873568be5c5e75



cluster: m26bc.359b6a48c0000b32 with 6844 members
md5: 124a591871e93e0272873568be5c5e75
sha1: 9d5cefcaea659c8974aeafd38c8187440d69a78f
sha256: 67978d6e3bd57f85b81efa04d83d579e5f85d45c0908cadfa59348aeaf81572f

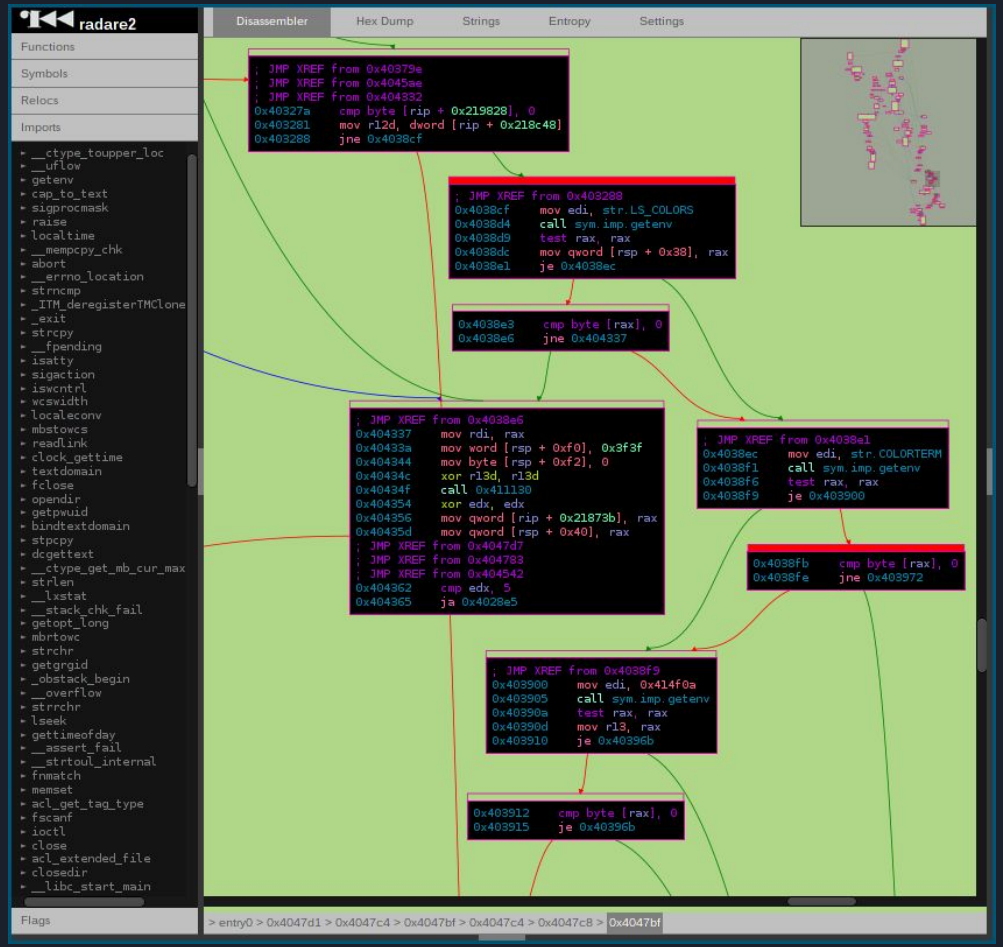
Association Method:
Control Flow Graph Analysis

Control Flow Graph Analysis

Control Flow Graph Based Virus Scanning
(DerbyCon 2014)

Douglas Goddard

https://www.youtube.com/watch?v=I0KXjN67hkA



https://rada.re/r/img/webui-graph.png

Analysis Technique: Graphing Threat Data

Schema: STIX

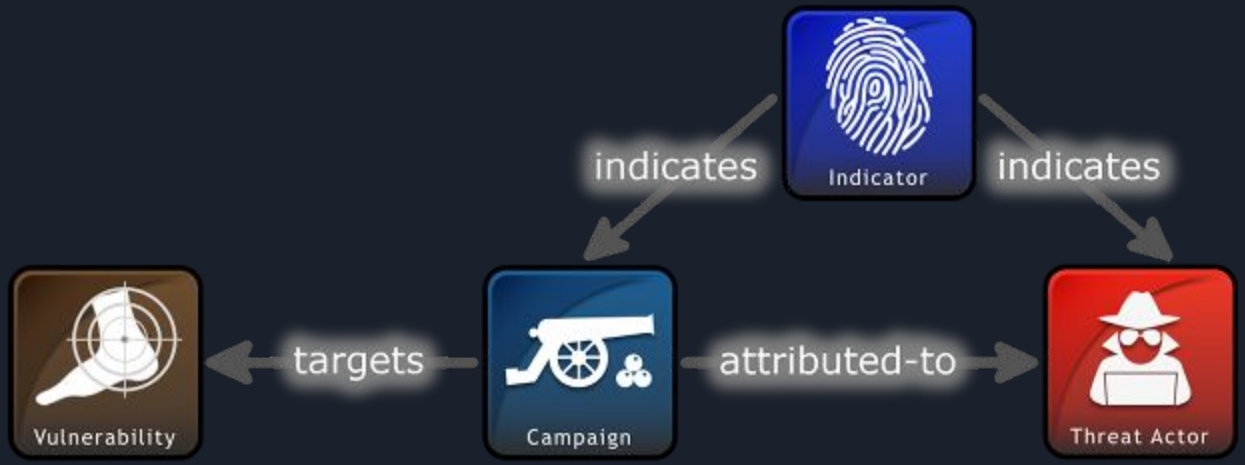
STIX Domain Objects (SDO)

Attack Pattern	Indicator	Malware
Campaign	Intrusion Set	Observed Data
Course of Action	Tool	Report
Identity	Vulnerability	Threat Actor

STIX Relationship Objects (SRO)

Relationship	Sighting

Schema: STIX



https://oasis-open.github.io/cti-documentation/stix/intro

Graph Tools: Data Formats

- Resource Description Framework (RDF)
  - https://www.w3.org/RDF/
- JSON for Linking Data
  - https://json-ld.org/

JSON for Linking Data: JSON-LD

```
{
  "@context": "https://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

RDF N-Quad

```
{
  set {
    _:filea <md5> "49634699ad232551c92081abd2d87d90" .
    _:filea <filesize> "213152" .
    _:filea <has_filename> _:setup .
    _:filea <malware_family_member_of> _:mfam .
    _:filea <has_imphash> _:imphasha .

    _:fileb <md5> "7a81e7811f19022f05aaaa433b6f8e8f" .
    _:fileb <filesize> "235536" .
    _:fileb <has_filename> _:install .
    _:fileb <malware_family_member_of> _:mfam .
    _:fileb <has_imphash> _:imphasha .

    _:setup <filename> "setup.exe" .

    _:install <filename> "install.exe" .

    _:mfam <malware_family> "Neshta" .

    _:imphasha <imphash> "9f4693fc0c511135129493f2161d1e86" .
  }
}
```

Graph Tools: Graph Databases

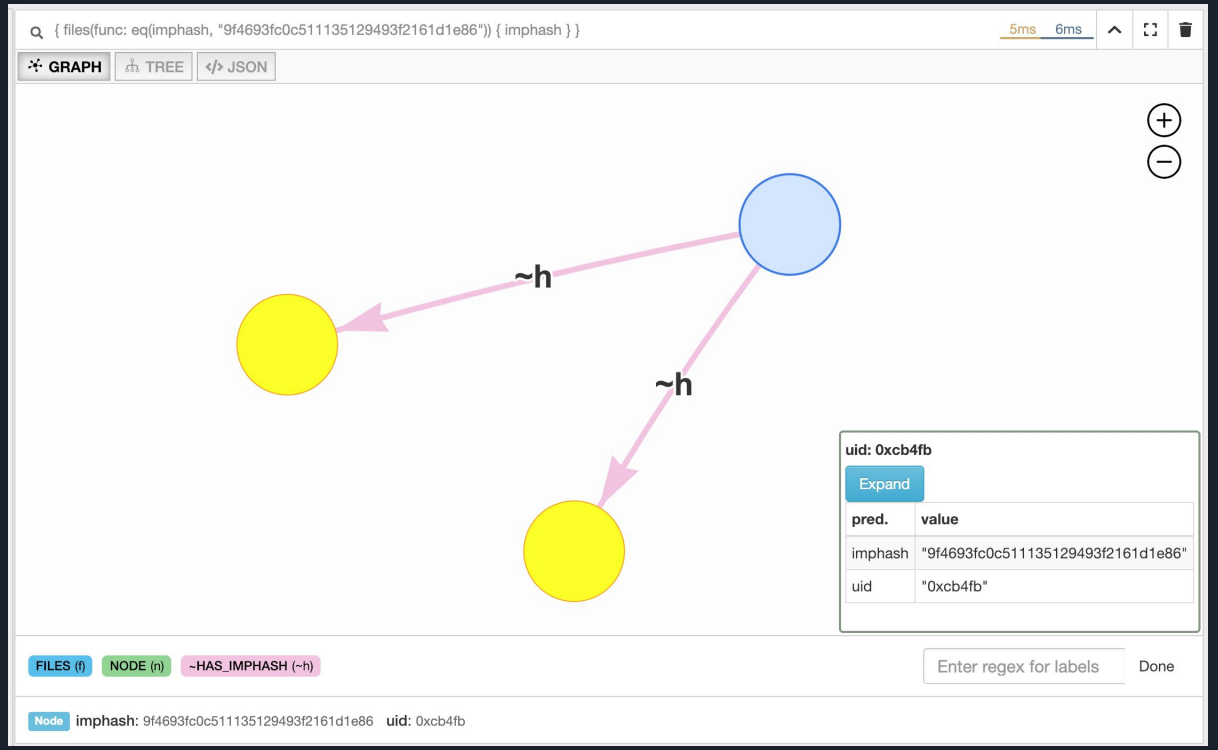
- Neo4j
  - https://neo4j.com/
- DGraph
  - https://dgraph.io/

Book

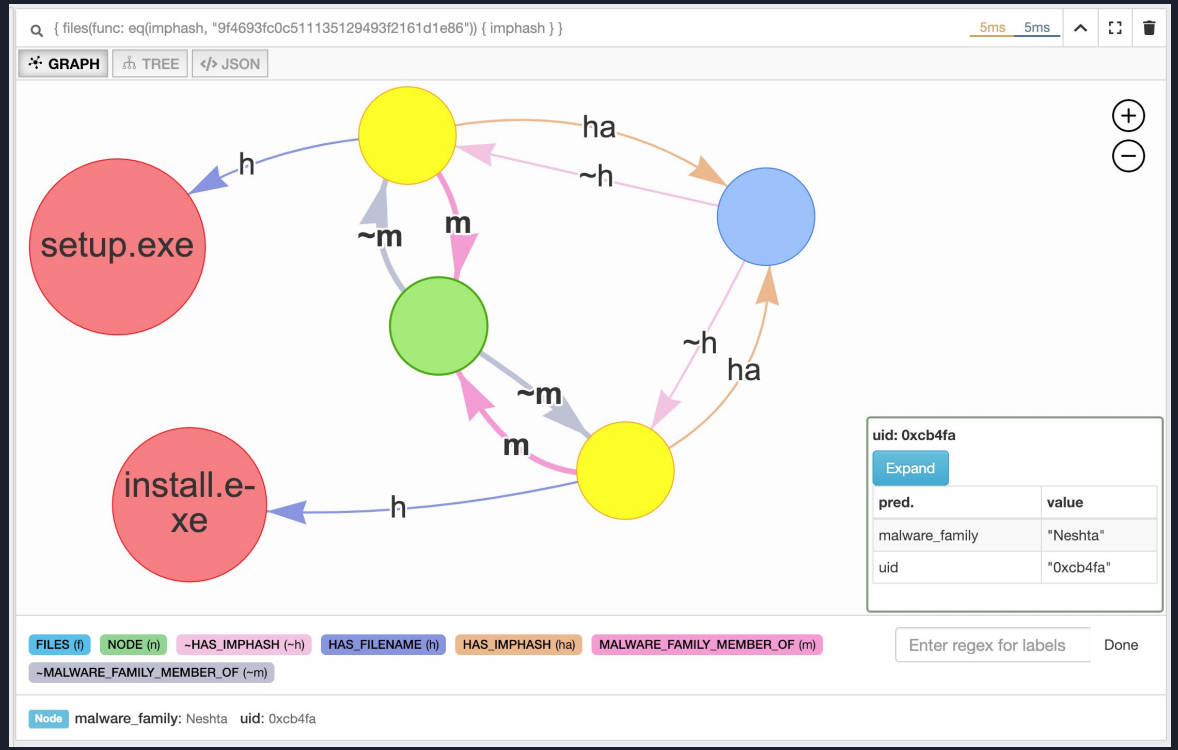
Introduction to Graph Theory

Richard J. Trudeau

Network Graph



Network Graph



Questions?

@MalwareUtkonos