Automation and Structured Knowledge in Tactical Threat Intelligence

Ivan Kwiatkowski and Ronan Mouchoux
Global Research and Analysis Team, Kaspersky Labs
Botconf 2018, Toulouse, France

KASPERSKY lab

Agenda

- Introduction
- Automation in static malware analysis
- Current Methods and Tools in Tactical Threat Intelligence
- Knowledge Integration in Tactical Threat Intelligence
- Conclusion

About the Speakers

Ivan Kwiatkowski
Senior Reseacher at GReAT, Kaspersky Lab

Activities

Reverse Engineering | Tooling | APT Investigation

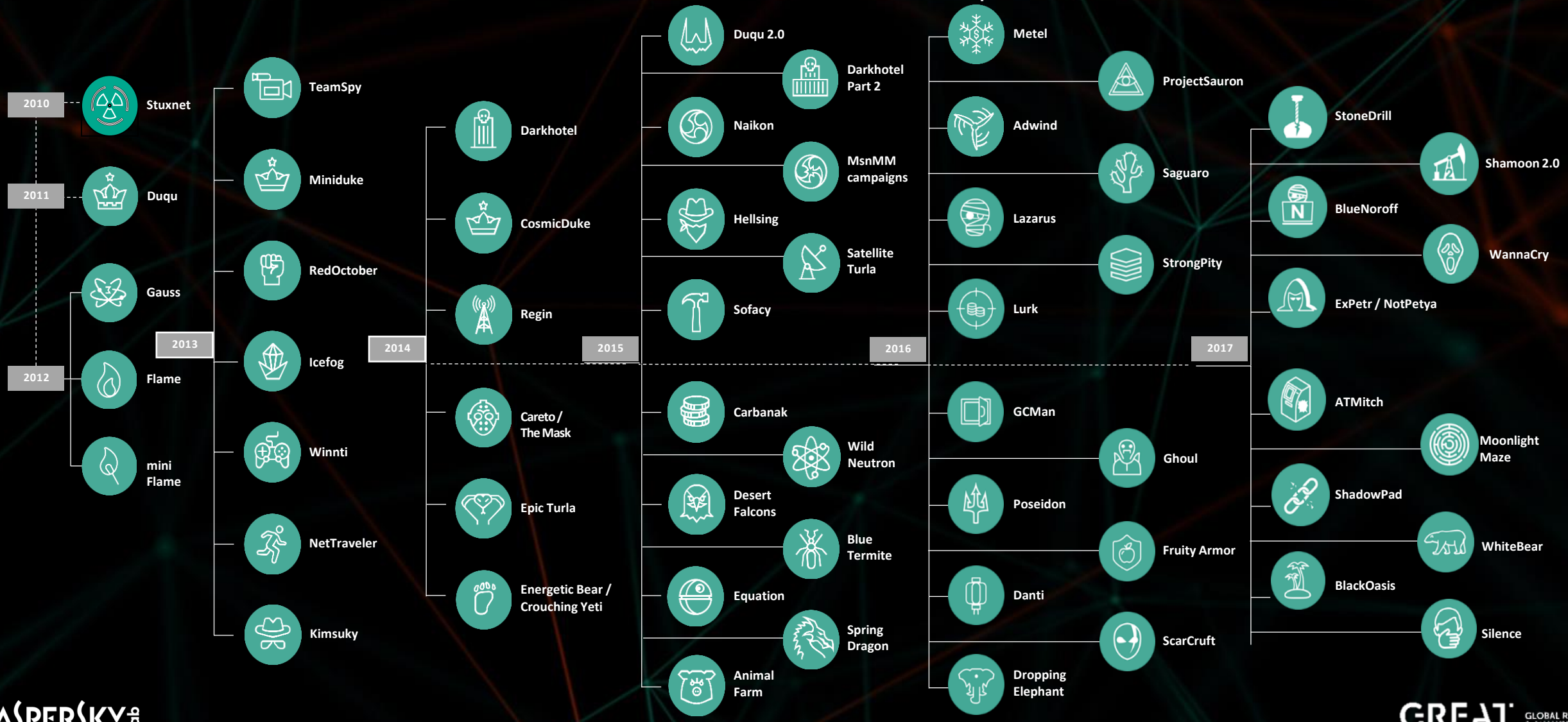
Ronan Mouchoux
Security Reseacher at GReAT, Kaspersky Lab

Activities

Intelligence Tradecraft | Scripting | APT Investigation

About the Team

Global Research & Analysis Team (GReAT) – Kaspersky Labs
40+ researchers dedicated to follow 100+ active advanced threat actors and operations



KASPERSKY lab

GREAT GLOBAL RESEARCH & ANALYSIS TEAM

Introduction

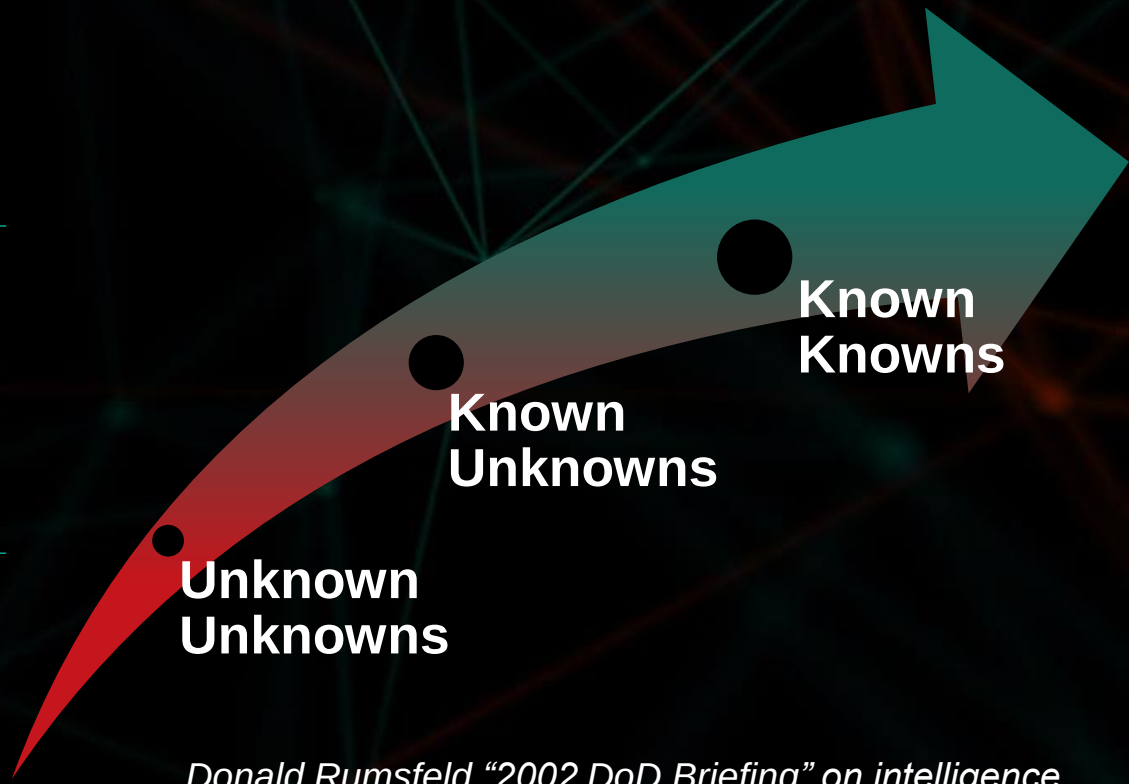
Knowledge production and ever-evolving threats landscape investigation

(Cyber) Threat Intelligence

Individual security events may be part of **broader intrusion strategies** requiring to mitigate a **threat** and not an isolated incident.

Intelligence is the **process** of creating **operational knowledge** about an evolving situation.

Intelligence is the **product** that helps its consumers **take decision and action** to optimize counter measures.



Donald Rumsfeld "2002 DoD Briefing" on intelligence

Intelligence is a consumer-driven activity

Strategic: high-level knowledge covering adversary strategies consumed by strategic instances of the organization to protect.

Operational: short-term knowledge covering ongoing adversary operations and communications consumed by cyber defence strategic instances.

Tactical: mid-to-long term knowledge covering adversary behaviour consumed by Incident Response, Supervision and Anticipation Team.

Technical: exploitation and management of data that are consumable by Incident Response, Threat Hunting and Supervision tools.

Threat Intelligence is still a young field, down sides

Strategic: has to fill the gap between Competitive Intelligence and Cybersecurity.

Operational: easy to collect on hacktivist and low-level cybercrime, out-of-reach for private companies on advanced adversaries.

Tactical: easy to get, hard to master.

Technical: is not about data feeds, it's about integrating and managing them.

Tactical Threat Intelligence Diagnosis

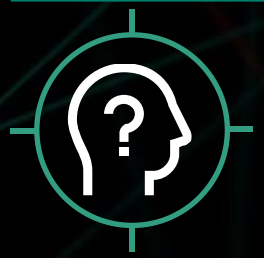
Most of the work remains manual and based exclusively on the tactical analyst's prior knowledge. Manual behavioural heuristics creation is nowadays the gold standard in Tactical Threat Intelligence.

The outcomes of tactical analysis are mainly used to feed strategic insights and data feeds (IOC). There is a huge lack in human analyst consumption and operationalisation of tactical analysis.

Consumers expect fancy attribution and flashy criminal-like behaviour profiling. There is little effort to understand adversary paths of intrusion in victim networks to assist network and security architects to adapt the defense posture of their infrastructure.

Tactical Threat Intelligence Problematics

In a general manner, **how do we build up human expertise and turn it into tools**?



In Tactical Threat Intelligence, **what are the current tools and methods and what are their limitations**?

In Tactical Threat Intelligence, **what is the first biggest barrier to automation**?

Automation in static malware analysis

How do we build up human expertise and turn it into tools ?

Automation in static malware analysis

Automating tasks is **expensive**. With experience the feeling of frustration when doing manually something that a machine could do goes away because automating the task can take way more effort to **automate it properly** than do it manually, even over long periods of time.

Human intelligence **doesn't scale**. The sheer volume of data produced by our society gives us no choice but walk towards globalized automated processing. In malware analysis, too many new samples are discovered everyday for the whole industry to look at : while expensive, automation cannot be avoided.

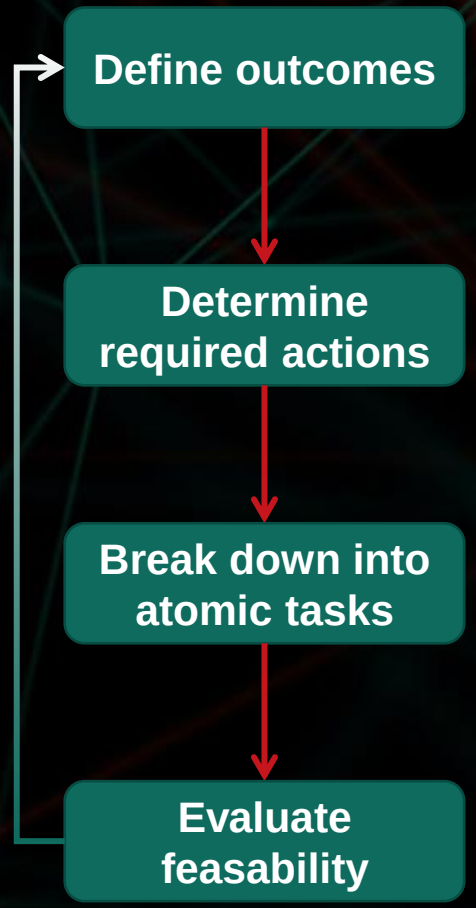
When your only resources are free time and programming skills, what balance will yield the biggest **return on investment**?

Automation in static malware analysis

Automation tasks most of time start with an ideal result. One start with apriori expectations that need to be identified and defined.

A task may be seen as inputs, an applied process and outputs. The applied process is a sequence of steps that are process manually. To translate them into code one need to determined the kind of transformation you should applied to inputs to get the desired outputs.

By breaking down these steps into atomic elements, one finally need to evaluate whether tasks can be programmed. If a single one of them needs human supervision / interaction, the expectations from the tool should probably be reduced.

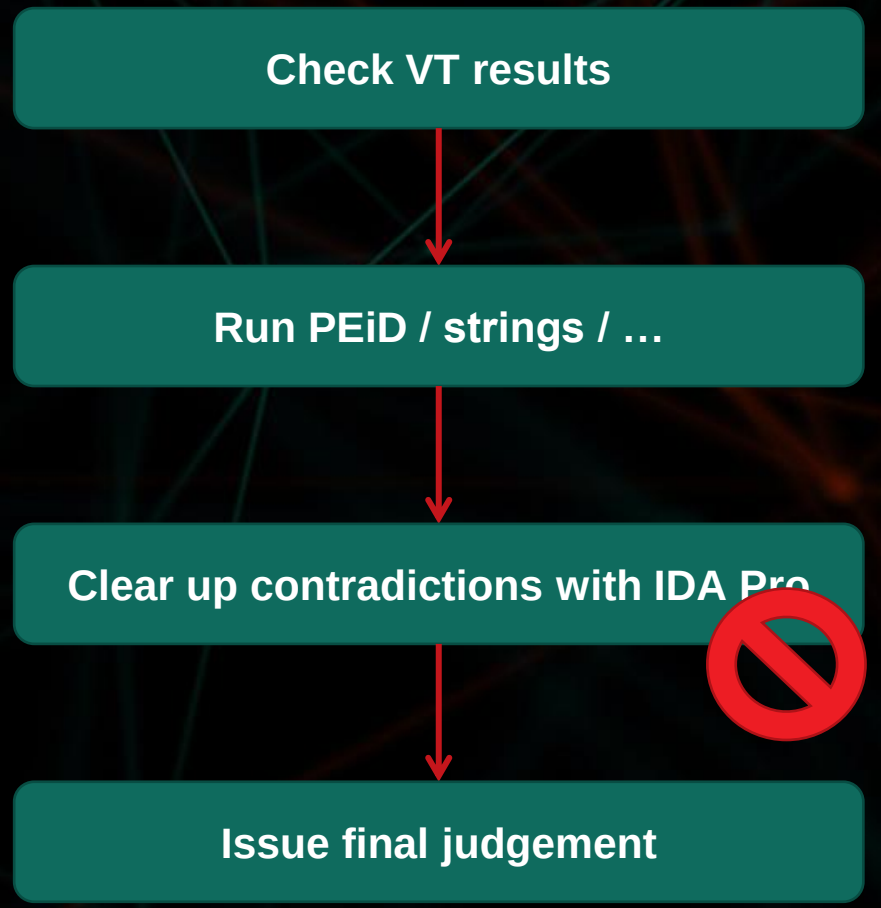


Expected outcome: “is this sample malicious?” → yes / no

When it comes to *static* malware analysis, an ideal scenario is to know if a sample is malicious or not. It relies on the review of contradicting pieces of information to balance whether or not a program is legitimate : this step cannot be easily automated.

Recent malware clustering research with machine-learning showed interesting possibilities. Figuring out the exact features to extract from malwares and trusted clean files to get good results seems a very ambitious expectative.

Stating that in the context of the task automation project, we consider one of the step of the process to hard to code. The project must not be abandoned, the outcomes have to be re-evaluate.

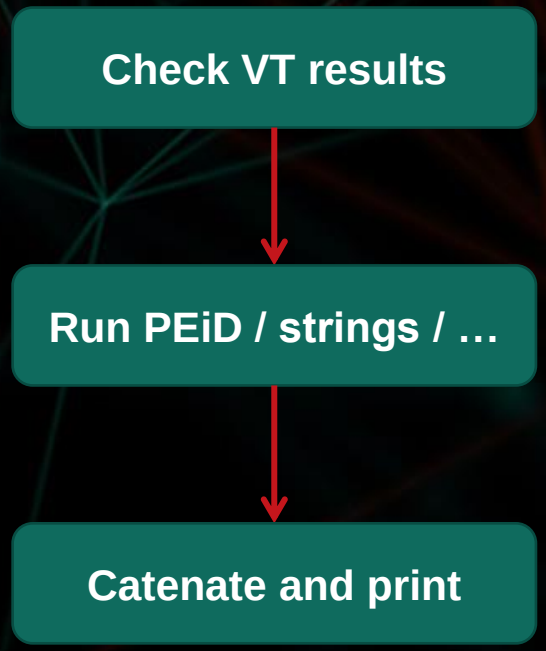


Revised expected outcome: “gather all the info available on this sample”

In this revised version, the tool expectations have been seriously toned down. Automation of contradiction analysis whas out of reach but in the process they are still steps that can be implemented.

For example, identifying bitcoin address in code or listing references to compression library, are two manual tasks that can be realizable and useful to automate.

To focus on the tasks that require actual human intelligence is the purpose of automation. It is to reduce the length of the « chain » of human painful tasks.



Current Methods and Tools in Tactical Threat Intelligence

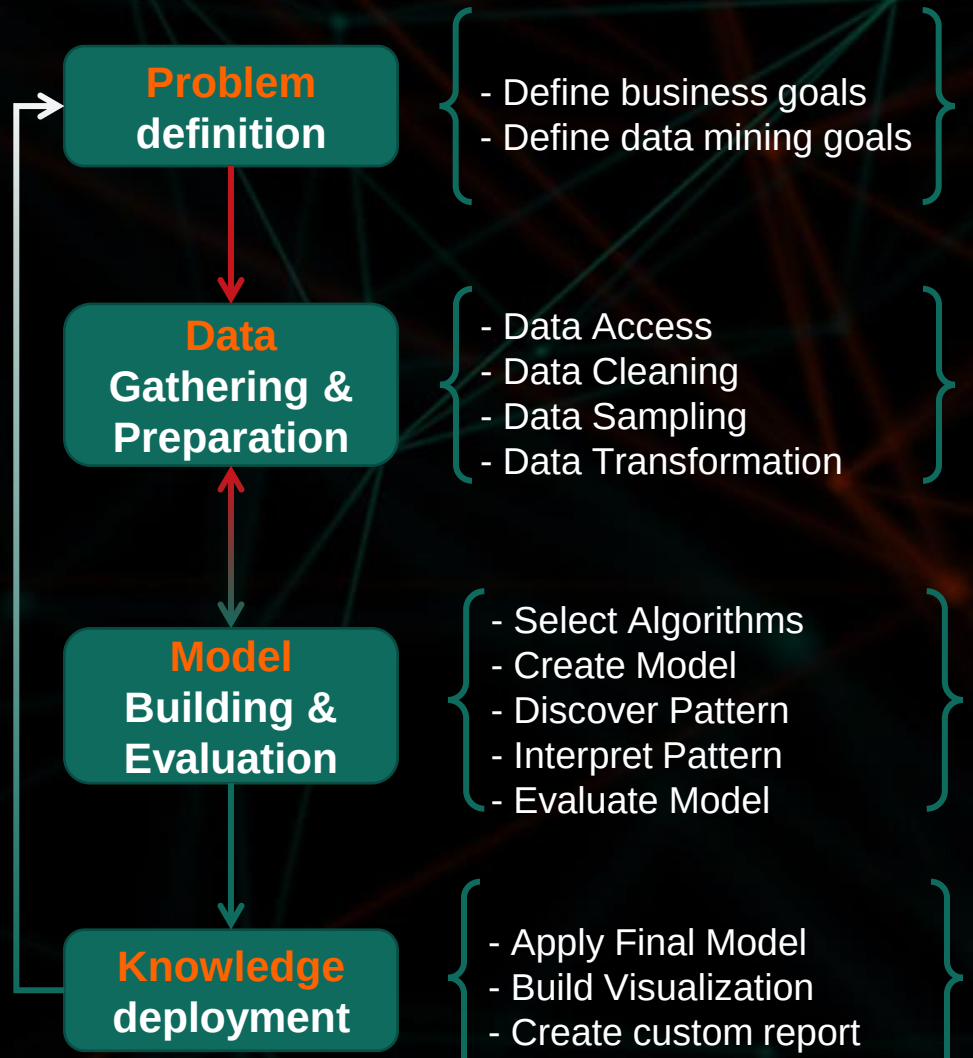
What are limitations of current tools and methods ?

The prevalence of models in Intelligence

Intelligence, as a product, is an operational and contextual knowledge which answer to a question build from technical data.

Intelligence, as a process, collect data from an environment, process data into information, analyse information into knowledge, package knowledge into intelligence.

Intelligence and Data Mining focus on seeing what used to be unseen. They have a very similar methodology. In bot of them models are used to fill the gap between data and knowledge.



The Data Mining Process

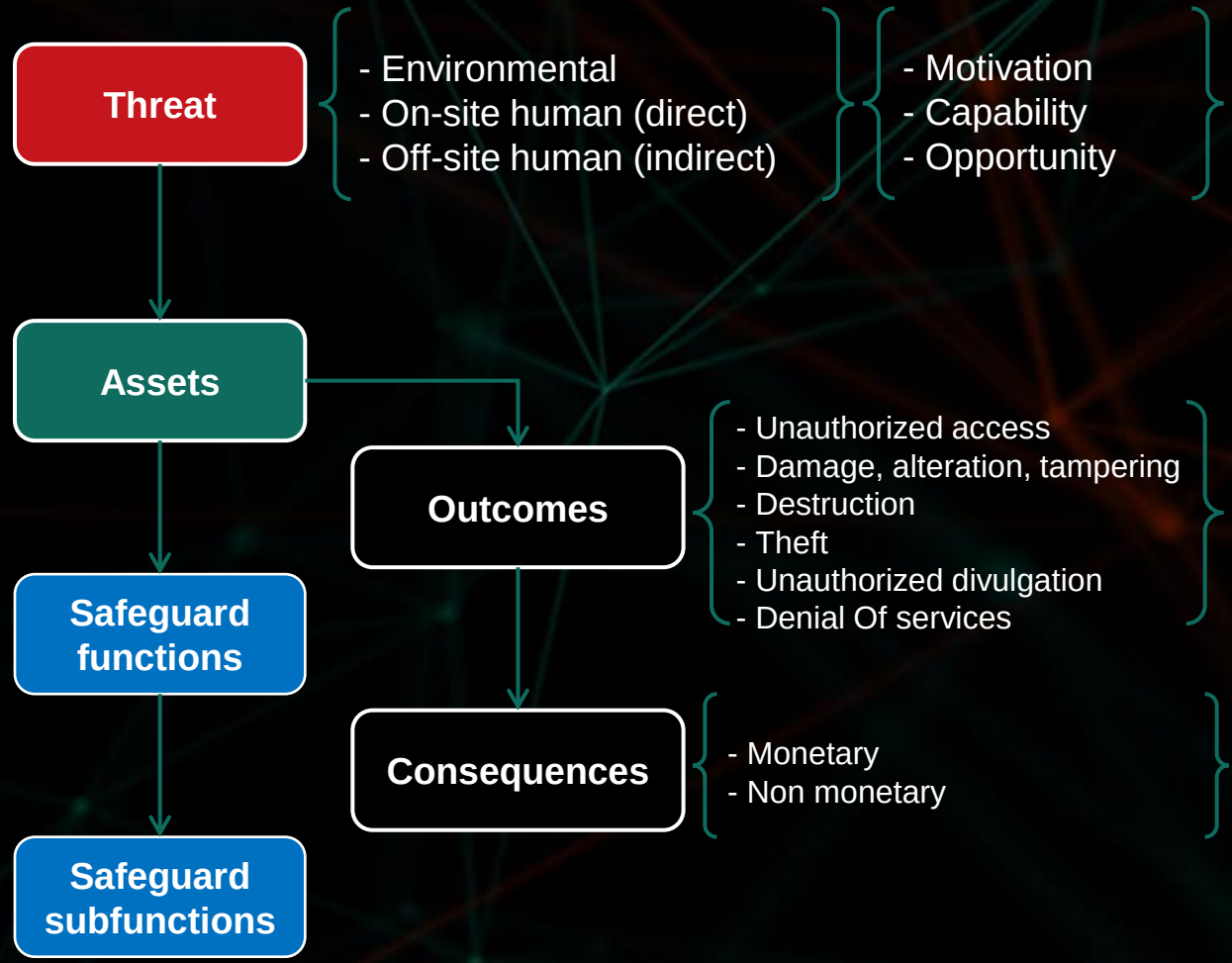
http://www.aaai.org/Papers/KDD/1996/KDD96-014.pdf

LAVA, the root of Dynamic Threat Analysis

The Los Alamos Vulnerability / Risk Assessment System (1983) is the first expert system for risk assessment on complex system.

LAVA has been developed as a sets of hierarchical trees for modeling risk assessment.

The system model is composed of threats, assets and undesirable outcomes to address the ever-changing threats to a system (Dynamic Threat).



LAVA's dynamic threat analysis abstract tree

https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-90-2042

Attack Trees, Adversary Model and the first Kill Chain

The paper "*Toward A Secure System Engineering Methodology*" (1998) result from a NSA funded research to create a methodology for characterizing attacks and choosing rational countermeasures.

This methodology is based on an "Attack Tree" model. An attack tree is a visualization tool to enumerate and weigh different attacks against a system.

This methodology defines an Attack Tree Adversary Model as well as three fundamentals steps for a successful attack (Kill Chain).

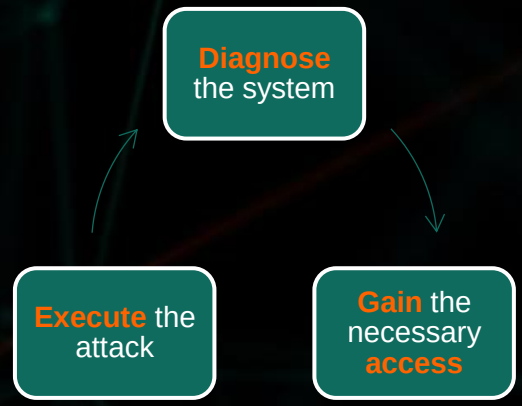
ADVERSARY CATEGORIES

- Hacker
- Insider
- Organized Crime
- Industrial Competitor
- Terrorist
- Intelligence Organization

CHARACTERISTICS

- Ressources
- Access
- Risk Tolerance

Attack Tree Adversary Model



Attack Tree « Kill Chain »

https://www.researchgate.net/publication/221067740_Toward_a_Secure_System_Engineering_Methodolgy

Lockheed Martin's Intrusion Kill Chain

In 2005, UK-NISCC and US-CERT describe a targeted attack that evaded conventional defense capabilities to collect sensitive information : this is the birth of the APT era.

Lockheed Martin leverage the military doctrine F2T2EA for a threat-focused approach to study intrusions from the adversaries' perspective : the Intrusion Kill Chain.

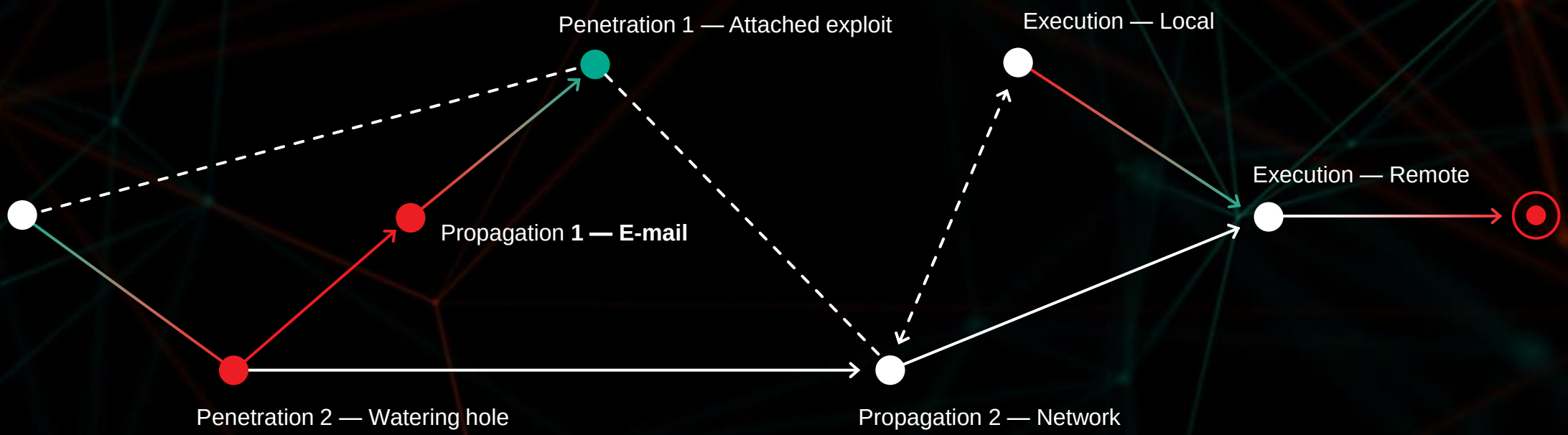
The intrusion kill chain become actionable when defenders align enterprise defensive capabilities with the adversary intrusion process.

Table 1: Courses of Action Matrix

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	"chroot" jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

Intrusion Kill Chain, limitations

In reality… nonlinear path of intrusion:



The Intrusion Kill Chain has been build to address targeted attack, that make use of a malware, to collect information.

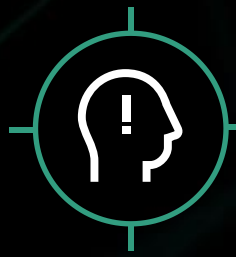
The linear property, reflecting the most effective intrusion path, can only be build *a-posteriori* of the attack.

Modus Operandi versus Tactics, Techniques and Procedures

The study of adversarial tactics is built over the observation of malicious actors' behaviours when planning and performing its intrusion. By empirical manner it also refers to the study and the knowledge of its habits.

The *Modus Operandi* concept comes from the criminology discipline and is mainly based over the 20th century psychological theories.

Modus Operandi Analysis has been in practice mainly applied to serial killers, serial rapists and serial terrorist cells.



Criminal profiling is centred on the repetition of assaults that require a particular state of mind that the society norms would qualify as *deviant*.

***TTP* originate from the US military to test and integrate new technologies in a combat system by examining operational and cognitive impacts in order to adapt existing roles, process and procedure.**

By mirror effect, the US military translate the concept to the review of opponent operations, mainly in the counter-terrorism and the cyber defence fields.

Clandestine or criminals operations involve organized groups performing structured actions that are perform in a rational manner rather than impulse.

Rational versus impulsive actions, this is why we prefer, even if we use the term modus operandi for convenience, to study adversarial actions execution in regards with the military concept of *Tactic, Technique and Procedure* (TTP).

Tactics, Techniques and Procedures, down sides

Based on the general definition of the TTP concept, the idealistic approach is to use this frame to describe how an attacker behave, in order to detect its action at a higher level of abstraction than the technical trails that are betraying its past or present presence.

Tactic : the employment and ordered arrangement of forces in relation to each other.

The tactic is more abstract than the technique that is more abstract than the procedure. The definition of the tactic only rely on the characterization of the technique.

Technique : non-prescriptive ways or methods used by human to perform missions, functions, or tasks.

The technique can only be perceive and detect through technical indicators, which affect the concept of higher-level of detection.

Procedure : standard and detailed steps that prescribe how to perform specific tasks.

The procedure used by the adversary will always be hided or blurred from the defence perspective.

Joint Publication 1-02, US Department of Defence Joint Chief of Staff

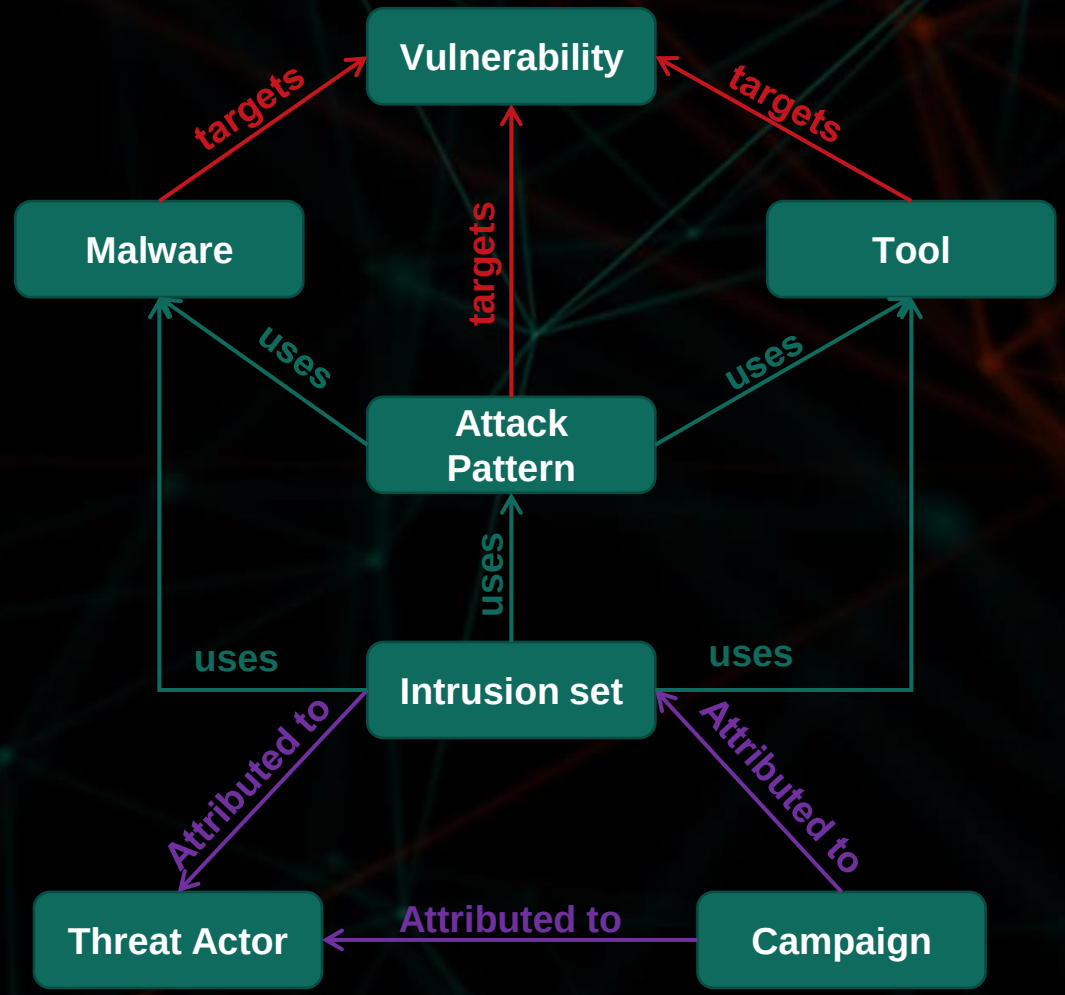
The MITRE Galaxy *

MITRE Corporation is an American NPO created at the end of the 1950's to support US army advancement in the C3I field (Command, Control, Communication and Intelligence). Its more notorious contribution to the information security field is probably the *Common Vulnerability and Exposure* (CVE) base, which references publicly disclosed vulnerabilities.

Nowadays, MITRE is engaged is a number of cyberdefence initiatives, with other organizations. But as a central research hub and leader, we refer to the related projects as the "MITRE Galaxy".

Projects related to cyberdefence knowledge are the CVE base, the CWE base, the MAEC base, the CAPEC base and the MITRE ATT&CK™ base. The projects related to express and communicates information are STIX and TAXII.

In this galaxy, the *STIX* presentation language serves as glue between information of any level of abstraction extracted from observed environment and it provides a grammar and a vocabulary through the other knowledge bases.



Over-simplified view of the STIX2 language for tactical analysis

* This presentation is not affiliated with, sponsored by, or endorsement by MITRE.
This presentation does not represents the views and opinions of MITRE or MITRE personnel.

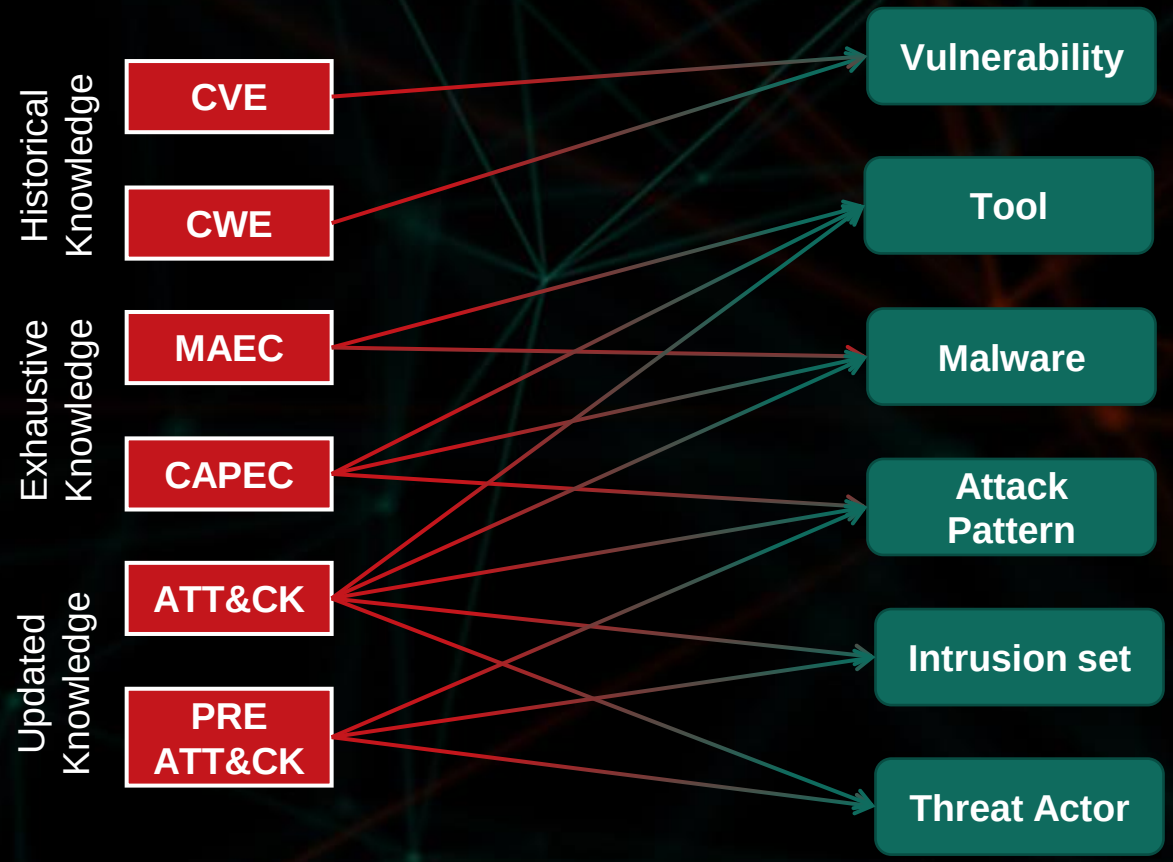
The MITRE Galaxy *

MITRE Corporation is an American NPO created at the end of the 1950's to support US army advancement in the C3I field (Command, Control, Communication and Intelligence). Its more notorious contribution to the information security field is probably the *Common Vulnerability and Exposure* (CVE) base, which references publicly disclosed vulnerabilities.

Nowadays, MITRE is engaged is a number of cyberdefence initiatives, with other organizations. But as a central research hub and leader, we refer to the related projects as the "MITRE Galaxy".

Projects related to cyberdefence knowledge are the CVE base, the CWE base, the MAEC base, the CAPEC base and the MITRE ATT&CK™ base. The projects related to express and communicates information are STIX and TAXII.

In this galaxy, the *STIX* presentation language serves as glue between information of any level of abstraction extracted from observed environment and it provides a grammar and a vocabulary through the other knowledge bases.



Over-simplified view of the « MITRE » Galaxy

* This presentation is not affiliated with, sponsored by, or endorsement by MITRE.
This presentation does not represents the views and opinions of MITRE or MITRE personnel.

The MITRE Galaxy, warning

MITRE Galaxy : is constantly in development and probably perfectible. But it offers a rare capability of rich and structured expression, in comparison with simple visualization and vulgarization brought by models such as the Cyber Kill Chain. On the other hand, we can understand that this richness requires a huge cost in understanding MITRE philosophy and a huge cost in implementing such expression system.

STIX and TAXII : are mostly promoted as a way to store and exchange indicators, but is not provided with methods to integrate or exploit data. STIX is a presentation language and TAXII a transport protocol. You must develop all the environment around. So for indicator sharing and management, open-source solution like MISP, seems easier to deploy. STIX is more suitable for graph-theory applied to CTI and building a supervision, incident response and reporting automated workflow. It is not a native IOC link-analysis system.

ATT&CK : is not design to be an exhaustive list of adversarial techniques or attack vectors.
Changes in listed techniques should occurs quaterly a year to provide an recent-real-case-based adversary categorization system to enable automated cyber threat simulation, both for Blue Team and Red Team.

Limitation of these approaches

There are no silver bullet. Each approach has a purpose. Identifying limitations is needed to avoid application where they are less relevant.

Trees : The branching system do not allow to express dependency or relationship between entities. Too much information on the same branch will broke the visualization ease, which is required to understand the model, interpret results and diagnose issues.

Kill Chain : As adversary scrabble about a network, they move along a graph. This representation does not fit the studied environment. The linear or cyclic property of kill chains only represent the optimal intrusion paths that is visible afterward.

Behavioural analysis : If you are not an intelligence agency able to gain privileged information on the adversary, you will have to infer the human behaviour from technical indicators. The analysis is then done on an abstraction of data that may brings loss of context and simplification that may hide dark corner cases.

Graphs : The visual appealing property of graph can mislead data interpretation by overlooking connection and induce correlation where they are not. Analyst might have the tendency to solely focus on the “big picture”. If complex and incomprehensible amounts of connections can be made clear and structured, graphs should not be used to drawn conclusion : it is an exploratory tool.

Knowledge integration in Tactical Threat Intelligence

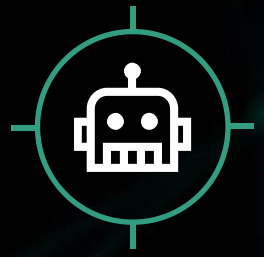
What is the first barrier to automation ?

From ad-hoc to optimized knowledge processing

Technical indicators sharing in cybersecurity within and across organization is effective thanks to an ontology broadly shared and accepted. Ontology limits complexity and organize knowledge with a controlled vocabulary. Ontology is the first step for processing and automation.

Adversary behaviour sharing has been for a long time provided in the form of intelligence report wrote in natural language by investigator. Each investigator may describe the same behaviour with varying words and sentences structures.

Without ontology, correlation or any type of computation is out-of-reach for an automated environment. Each developer creates his own logic to map his knowledge into machine-processable information.



MITRE Knowledge bases provide a structured controlled vocabulary while STIX provide a structured expression logic to describe cyberattack related information.

Maturity Level	Description
Ad-Hoc	Undocumented process, ad-hoc implementation, variable results and no quality standard
Repeatable	Documented process that allows reproductibility
Defined	Process definition, assurance of consistent implementation that provide an understanding of the process. At this point, it is probable that advanced technology can be usefully introduced.
Managed	Process measurement. At this point, most significant quality improvements begin to appear.
Optimized	Process management with deliberate and controlled improvment and optimization.

Characterizing the software process: a maturity framework, IEEE Software (Volume: 5, Issue: 2, March 1988)

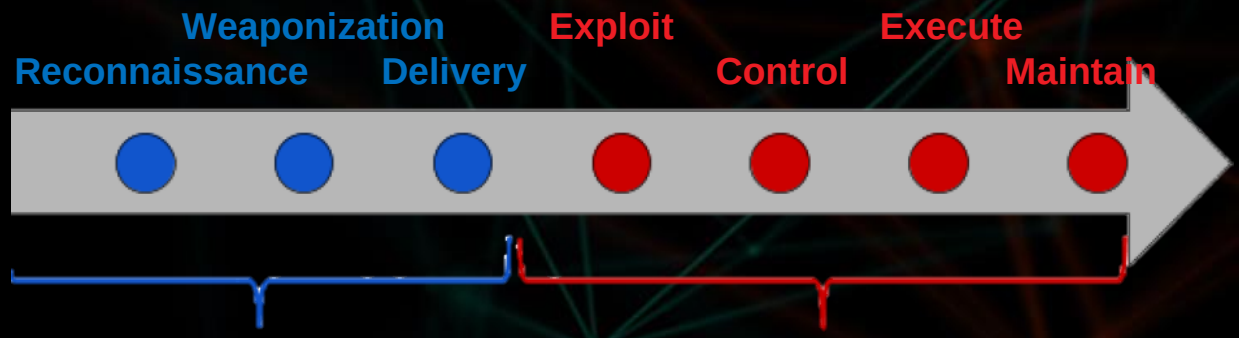
The MITRE ATT&CK™ Framework

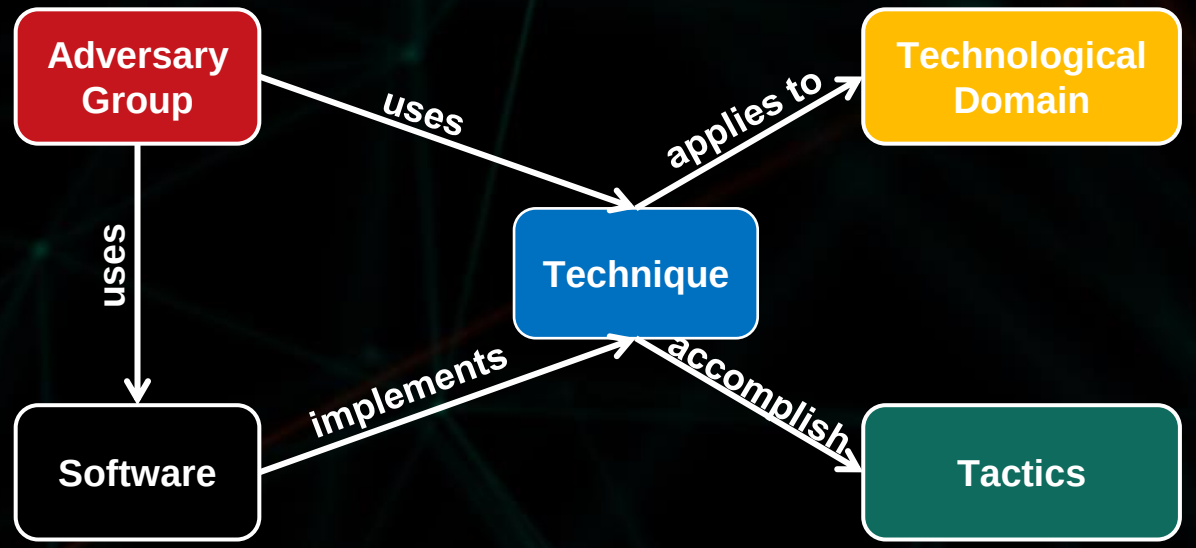
MITRE Adversarial Tactics Techniques & Common Knowledge (ATT&CK) is a peered-review curated knowledge base and model for describing the behaviour adversaries engaged in while planning or performing an intrusion.

ATT&CK articulate around four main components : tactics (short-term tactical objective), techniques (means), adversary usage of techniques, software implementation of techniques.

The main goal is to provide metrics to assess then improve post-exploitation detection capabilities of APT quicker than the manual Threat Hunting process.

Tactic and Technique are articulate around the notion of technological domain. An intrusion append inside a technology domain that have specificities. These specificities bring constraint in the way to perform action and the adversary has to take advantage of them to accomplish its objectives.





https://www.mitre.org/publications/technical-papers/mitre-attack-design-and-philosophy

The MITRE ATT&CK™ Matrices

The ATT&CK display rely on a matrix-based format, with tactics as column and techniques as row. The matrix format has been chosen by MITRE to balance sufficient technical details at the technique level and more abstract information to describe in which context one technique is applied.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control
Spearphishing Attachment	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History
Spearphishing via Service	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing
Supply Chain Compromise	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compiled HTML File

Bypass User Account Control

Windows User Account Control (UAC) allows a program to elevate its privileges to perform a task under administrator-level permissions by prompting the user for confirmation. The impact to the user ranges from denying the operation under high enforcement to allowing the user to perform the action if they are in the local administrators group and click through the prompt or allowing them to enter an administrator password to complete the action. [1]

If the UAC protection level of a computer is set to anything but the highest level, certain Windows programs are allowed to elevate privileges or execute some elevated COM objects without prompting the user through the UAC notification box. [2] [3] An example of this is use of rundll32.exe to load a specifically crafted DLL which loads an auto-elevated COM object and performs a file operation in a protected directory which would typically require elevated access. Malicious software may also be injected into a trusted process to gain elevated privileges without prompting a user. [4] Adversaries can use these techniques to elevate privileges to administrator if the target process is unprotected.

Many methods have been discovered to bypass UAC. The Github readme page for UACMe contains an extensive list of methods [5] that have been discovered and implemented within UACMe, but may not be a comprehensive list of bypasses. Additional bypass methods are regularly discovered and some used in the wild, such as:

- `eventvwr.exe` can auto-elevate and execute a specified binary or script. [6] [7]

ID: T1088
Tactic: Defense Evasion, Privilege Escalation
Platform: Windows
Permissions Required: User, Administrator
Effective Permissions: Administrator
Data Sources: System calls, Process monitoring, Authentication logs, Process command-line parameters
Defense Bypassed: Windows User Account Control
Contributors: Stefan Kanthak, Casey Smith
Version: 1.0

Examples

Name	Description
APT29	APT29 has bypassed UAC.[9]
AutoIt backdoor	AutoIt backdoor attempts to escalate privileges by bypassing User Access Control.[10]
BlackEnergy	BlackEnergy attempts to bypass default User Access Control (UAC) settings by exploiting a backwar
BRONZE BUTLER	BRONZE BUTLER malware xxmm contains a UAC bypass tool for privilege escalation.[12]
Cobalt Group	Cobalt Group has bypassed UAC.[13]
Cobalt Strike	Cobalt Strike can use a number of known techniques to bypass Windows UAC.[14]
Downdelph	Downdelph bypasses UAC to escalate privileges by using a custom "RedirectEXE" shim database.[15]
FinFisher	FinFisher performs UAC bypass.[16][17]

Mitigation

Remove users from the local administrator group on systems. Although UAC bypass techniques exist, it is still prudent to use the highest enforcement level for UAC when possible and mitigate bypass opportunities that exist with techniques such as DLL Search Order Hijacking.

Check for common UAC bypass weaknesses on Windows systems to be aware of the risk posture and address issues where appropriate. [8]

Detection

There are many ways to perform UAC bypasses when a user is in the local administrator group on a system, so it may be difficult to target detection on all variations. Efforts should likely be placed on mitigation and collecting enough information on process launches and actions that could be performed before and after a UAC bypass is performed. Monitor process API calls for behavior that may be indicative of Process Injection and unusual loaded DLLs through DLL Search Order Hijacking, which indicate attempts to gain access to higher privileged processes.

Some UAC bypass methods rely on modifying specific, user-accessible Registry settings. For example:

- The `eventvwr.exe` bypass uses the `[HKEY_CURRENT_USER]\Software\Classes\mscfile\shell\open\command` Registry key [8]
- The `sdclt.exe` bypass uses the `[HKEY_CURRENT_USER]\Software\Microsoft\Windows\CurrentVersion\App Paths\control.exe` and `[HKEY_CURRENT_USER]\Software\Classes\exefile\shell\runas\command\isolatedCommand` Registry keys. [30] [31]

Analysts should monitor these Registry settings for unauthorized changes.

References

1. Lich, B. (2016, May 31). How User Account Control Works. Retrieved June 3, 2016.
2. Russinovich, M. (2009, July). User Account Control: Inside Windows 7 User Account Control. Retrieved July 26, 2016.
3. Microsoft. (n.d.). The COM Elevation Moniker. Retrieved July 26, 2016.
4. Davidson, L. (n.d.). Windows 7 UAC whitelist. Retrieved November 12, 2014.
5. UACME Project. (2016, June 16). UACMe. Retrieved July 26, 2016.
6. Nelson, M. (2016, August 15). "Fileless" UAC Bypass using eventvwr.exe and Registry Hijacking. Retrieved December 27, 2016.
7. Salvio, J., Joven, R. (2016, December 16). Malicious Macro Bypasses UAC to Elevate Privilege for Farelt Malware. Retrieved December 27, 2016.
8. Medin, T. (2013, August 8). PsExec UAC Bypass. Retrieved June 3, 2016.
9. Dunwoody, M. and Carr, N.. (2016, September 27). No Easy Breach DerbyCon 2016. Retrieved October 4, 2016.
10. Settle, A., et al. (2016, August 8). MONSOON - Analysis Of An APT Campaign. Retrieved September 22, 2016.
17. Allievi, A.,Flori, E. (2018, March 01). FinFisher exposed: A researcher's tale of defeating traps, tricks, and complex virtual machines. Retrieved July 9, 2018.
18. Reynolds, J.. (2016, September 14). H1N1: Technical analysis reveals new capabilities – part 2. Retrieved September 26, 2016.
19. Sherstobitoff, R. (2018, March 02). McAfee Uncovers Operation Honeybee, a Malicious Document Campaign Targeting Humanitarian Aid Groups. Retrieved May 16, 2018.
20. Hromcová, Z. (2018, June 07). InvisiMole: Surprisingly equipped spyware, undercover since 2013. Retrieved July 10, 2018.
21. Magius, J., et al. (2017, July 19). Koadic. Retrieved June 18, 2018.
22. Cymmetria. (2016). Unveiling Patchwork - The Copy-Paste APT. Retrieved August 3, 2016.
23. Ash, B., et al. (2018, June 26). RANCOR: Targeted Attacks in South East Asia Using PLAINTEE and DDKONG Malware Families. Retrieved July 2, 2018.
24. Nicolas Verdier. (n.d.). Retrieved January 29, 2018.
25. Faou, M. and Boutin, J.. (2017, February). Read The Manual: A Guide to the RTM Banking Trojan. Retrieved March 9, 2017.

https://attack.mitre.org/matrices/enterprise/

Mapping Manalyzer Outputs to ATT&CK

```
Plugin imports:
===============

MALICIOUS: "The PE contains functions mostly used by malware"
- Code injection capabilities
- Code injection capabilities (process hollowing)
- Code injection capabilities (PowerLoader)
- Code injection capabilities (atom bombing)
- Code injection capabilities (process doppelganging)
- Possibly attempts GINA replacement
- Uses functions commonly found in keyloggers
- Functions related to the privilege level
- Deletes entries from the event log
```

```
Plugin findcrypt:
=================

INFO: "Cryptographic algorithms detected in the bin
- [cryptographic algorithm]

INFO: "Libraries used to perform cryptographic oper
- OpenSSL
- Crypto++
- Botan
- Microsoft's Cryptography API
```

```
Plugin suspicious strings:
==========================

SUSPICIOUS: "Strings found in the binary may indicate undesirable behavior:"
- Contains references to system / monitoring tools
- Contains references to internet browsers
- Contains references to debugging or reversing tools
- Contains references to security software
- Tries to detect virtualized environments
- Looks for VMWare presence
- Looks for Sandboxie presence
- Looks for VirtualPC presence
- Looks for VirtualBox presence
- Looks for Parallels presence
- Looks for Qemu presence
- May have dropper capabilities
- Is an AutoIT compiled script
- Accesses the WMI
- Contains obfuscated function names
- Contains a XORed PE executable
- Contains a base64-encoded executable
- Miscellaneous malware strings
- References the BITS service
```

lation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distrib Compon t Object Model	Clipboard Data	Data Encrypted	Connection Proxy
LLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data Staged	Data Transfer Size Limits	Custom Command and Control Protocol
Ls	Application Shimming	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Information Repositories	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
n Shimmi	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding
ation	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data from Network Shared Drive	Exfiltration Over Other Network Medium	Data Obfuscation
s	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Data from Removable Media	Exfiltration Over Physical Medium	Domain Fronting
	Exploitation for Privilege Escalation	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels

https://github.com/JusticeRage/Manalyze/blob/master/bin/attack.py

Machine-processable information translation from Manalyser to ATT&CK

```
{
# …
"imports": {
    "plugin_output": {
        "Possibly launches other programs": [
            "ShellExecuteW"
        ],
        "Has Internet access capabilities": [
            "InternetConnectW",
            "InternetCloseHandle",
            "InternetReadFile",
            "InternetOpenW",
            "InternetSetStatusCallbackW"
        ],
        "Enumerates local disk drives": [
            "GetVolumeInformationW",
            "GetLogicalDriveStringsW",
            "GetDriveTypeW"
        ]
    },
},
# …
```

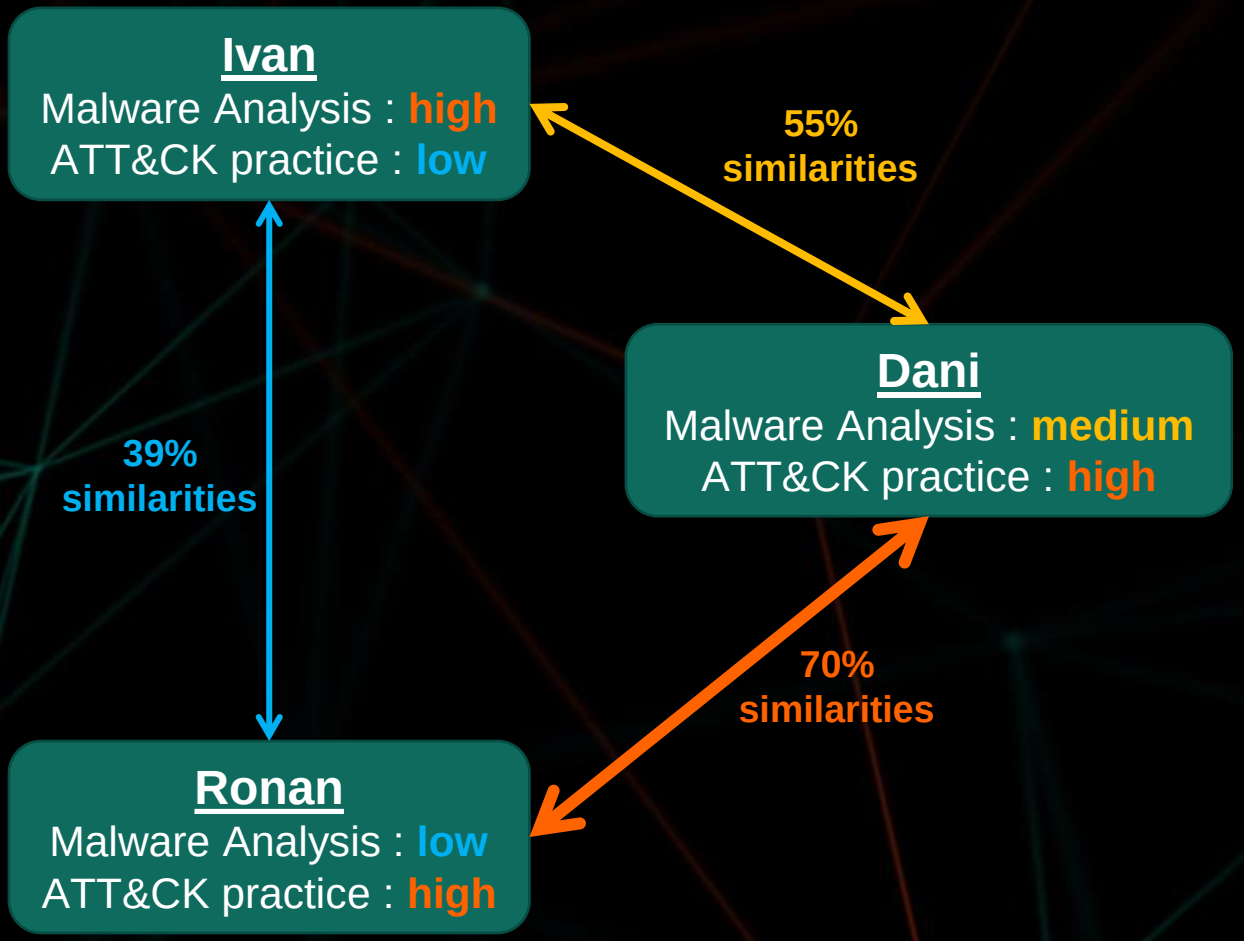
```
{
    #...
    "Execution": [
        "Execution through API"
    ],
    "Discovery": [
        "System Information Discovery"
    ]
}
```

Outputs of Manalyzer's plugins

Outputs of Manalyzer's ./bin/attack.py

ATT&CK mapping Maturity Level, the triple-blind experiment

Separately, without consultation, similarities in Manalyzer to ATT&CK mapping between three GReAT researchers. In this context of static malware analysis mapping, a high skilled malware analyst may have a deeper understanding of the atomic techniques, when a high skilled ATT&CK practitioner may have greater ability to identify how to the full matrix fits the situation.



Triple-blind ATT&CK Mapping Overlap

Maturity Level	Description
Ad-Hoc	Undocumented process, ad-hoc implementation, variable results and no quality standard
Repeatable	Documented process that allows reproductibility
Defined	Process definition, assurance of consistent implementation that provide an understanding of the process. At this point, it is probable that advanced technology can be usefully introduced.
Managed	Process measurement. At this point, most significant quality improvements begin to appear.
Optimized	Process management with deliberate and controlled improvment and optimization.

Estimated triple-blind Maturity Level

Ad-Hoc Ontology implementation, a global issue

Comparing a Kraken Ransomware V2.0.7 (BCD2A924EE16F3A2ED4B77D0C09FC3A0) on two separate sandbox services, with the same execution parameters, that implement mapping to ATT&CK.

Initial Access	Execution	Persistence	Priviledge Escalation
	- CLI - Execution through API		
Defense Evasion	**Credential Access**	**Discovery**	**Lateral Movement**
- Modify Registry		- Process Discovery - Query Registry	
Collection	**Exfiltration**	**C&C**	

APP.ANY.RUN

Initial Access	Execution	Persistence	Priviledge Escalation
	- Service Execution	- Hooking - Kernel Modules and Extension	- Hooking - Process Injection
Defense Evasion	**Credential Access**	**Discovery**	**Lateral Movement**
- File Delection - Modify registry - Process Injection	- Hooking	- Process Discovery - Query Registry - Remote System Discovery - Security Software Discovery	
Collection	**Exfiltration**	**C&C**	
	- Data Compressed		

HYBRID ANALYSIS

Each solution may have different analysis capabilities or visibility. Each solution may map differently outputs to ATT&CK techniques.
Each developpers may have their own understanding of an ATT&CK technique definition.

Conclusion

The Tactical Threat Intelligence Hypothesis

How do we build a human expertise and turn it into tools ?

Human expertise : is the process to produce cognitive strategies that go from single rules, to heuristics, to tactics, to macro-processing.

What are limitations of current tools and methods ?

Current tools and methods are : Kill Chain Like Models; Modus Operandi Analysis; Tactic, Techniques and Procedures Analysis; the « MITRE Galaxy »; that all have a specific purpose and their own limitations.

What is the first barrier to automation ?

Automation kick start are : a shared common practice in mapping the most common Structured Knowledge and Structured Language within and across organization.

Machine-based knowledge mapping to structured language

A tool's output has usually its own ontology. It can be very difficult for someone not familiar with the source code to figure out which ATT&CK technique corresponds to which output.
The ***tool author needs to be involved in the mapping***.

A one-to-one mapping is unrealistic : missing categories, missing techniques, overlaps and ambiguities in definition.

In dynamic analysis, does a sequence reliably indicate what the developper was trying to achieve? Tools look for **actions** or **capabilities**. ATT&CK focuses on **intent**.

Binary Padding

Some security tools inspect files with static signatures to determine if they are known malicious. Adversaries may add data to files to increase the size beyond what security tools are capable of handling or to change the file hash to avoid hash-based blacklists.

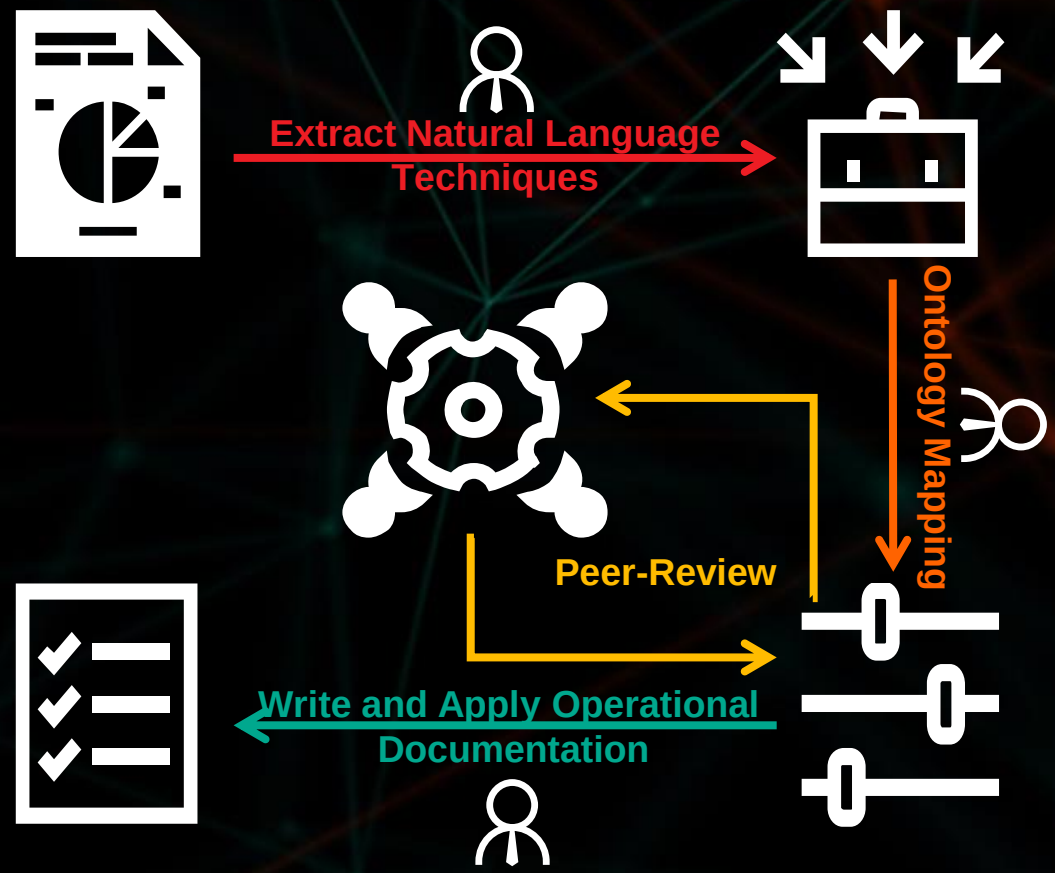
https://attack.mitre.org/wiki/Technique/T1009/

Human-based knowledge mapping to structured language

Adversary behaviour expression mainly provided in the form of intelligence report wrote in natural language by investigator. Each investigator may describe the same behaviour with varying words and sentences structures.

Investigator have a deep to intimate knowledge about the malware or groups they are writing about. They may feel that an ontology do not give enough flexibility while trying to express their observation in a standard manner.

The devil lies in precocious automation. Recommendation System or Natural Language Processing solutions are an optimization that can only be consider after the human-mapping process is repeatable, defined and managed.



Let’s Talk!

Ivan Kwiatkowski and Ronan Mouchoux
Global Research and Analysis Team, Kaspersky Labs
www.securelist.com

KASPERSKY lab