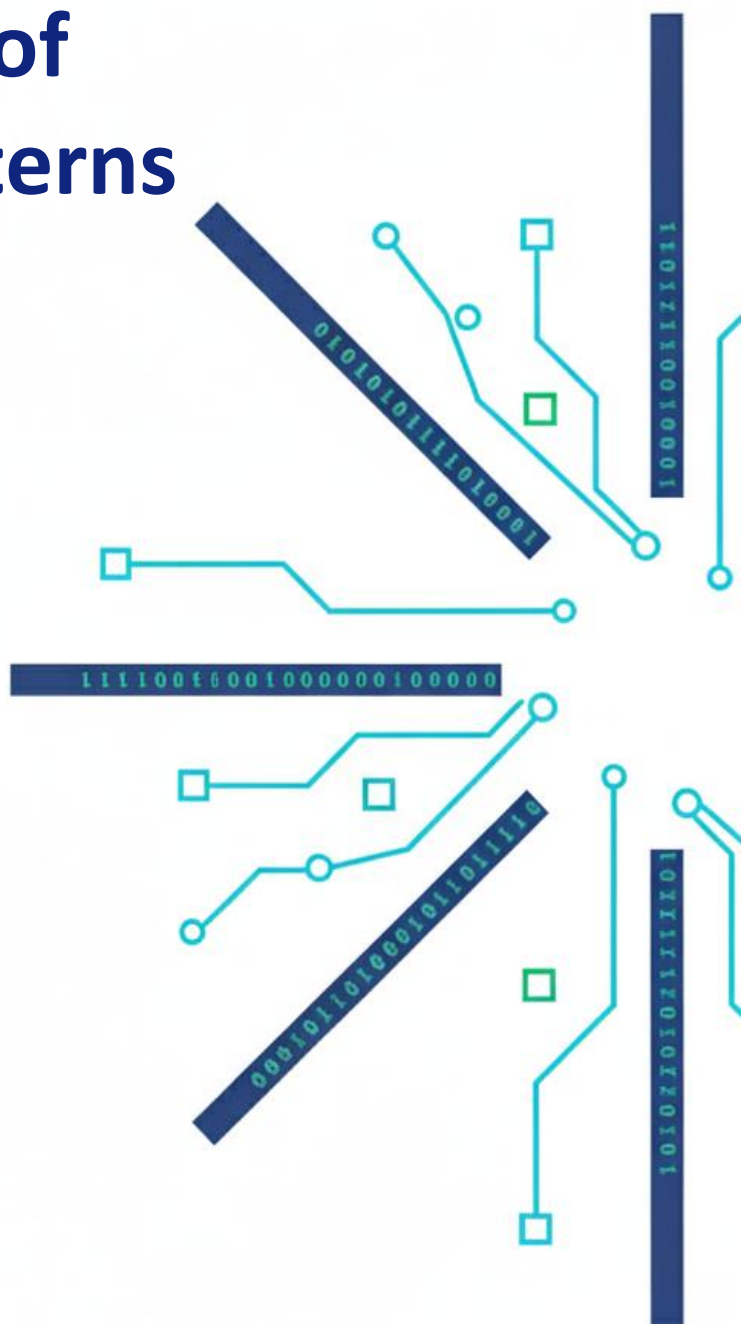


D22 – Database of behavioural patterns



The project Automatization of digital forensics and incident response (ADFIR) funded by the European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic under project No. č. 09I05-03-V02-00079.

CONTENTS

1 Project description	3
2 Introduction.....	4
3 Structure of deliverable – description of columns.....	5
3.1 MITRE	5
3.2 Event Type.....	6
3.3 Trigger Condition	6
3.4 Brief.....	7
3.5 Description	7
4 How to read a rule (example)	9
5 Why is this structure essential for automatic forensic analysis	10
6 Notes on Catalog Maintenance and Expansion	11
7 Appendix	12

1 Project description

The project **Automatization of digital forensics and incident response** (hereinafter referred to as “**ADFIR**”) is funded by the **European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic** under project No. č. 09I05-03-V02-00079. This project addresses one of the key challenges in cybersecurity and information security – how to process the massive volume of digital evidence generated during cybersecurity incidents or forensic investigations. Currently, this process is highly demanding in terms of human resources and time. Therefore, automation using machine learning methods can significantly **improve the quality of digital forensic analysis** and reduce the time required to perform it. Overall, this enables security teams to respond more effectively to cyber threats. Main benefits of this project are:

- **Accelerated Resolution of Cybersecurity Incidents.** The project ADFIR introduces automated approaches to collecting, processing, and analyzing digital evidence. As a result, security teams can identify the causes of incidents more quickly and adopt effective measures to address them.
- **Reduced Workload for Forensic Analysts.** Routine and time-consuming tasks involved in processing digital evidence will be replaced by automated methods. This will allow analysts to focus on more complex cases and strategic decision-making.
- **Higher Quality and Consistency of Outputs.** The use of unified methodologies and tools ensures that the processed digital evidence will be more accurate, consistent, and easily verifiable. This significantly reduces the risk of errors caused by human factors.
- **Potential Use in Criminal Proceedings.** The project outputs will be developed in compliance with legal requirements and standards, allowing the digital evidence to be accepted as relevant evidence for investigations and court proceedings.

2 Introduction

Deliverable is a catalogue of detection rules. Each rule describes one specific, observable trace of the activity. The rule is based on one of three types of telemetry:

- **process creation events** (ProcessCreate) - e.g. the launch of a suspicious or abusable binary file - typically Event ID 4688, Sysmon Event ID 1 or EDR telemetry.
- **records in event logs** (Windows Event Log / ETW) - e.g. PowerShell events, security logs, etc.,
- Registry change events - e.g. creation or modification of a persistence application.

The rules are **static** in the sense that they do not use machine learning or probabilistic scoring. Each rule has a clearly defined trigger that evaluates to true or false. As a result, every hit is fully explained, audited and reproduced - the analyst always knows exactly why the rule has responded.

Each rule is also mapped on a technique from the **MITRE ATT&CK** framework, which is a common taxonomy of attacks allows to measure the coverage of the current techniques.

3 Structure of deliverable – description of columns

Below is an overview of all the columns, followed by a detailed description of each of them.

Column	What's included	Value example
MITRES	Identifier of the MITRE ATT&CK technique to which the rule is mapped	T1113
Event Type	Rule telemetry/trigger event type	OnProcessCreate
Trigger Condition	The exact condition under which the rule responds	process.name == 'psr.exe'
Brief	A short, humanly read the name of the rules	Misuse of screen capture Psr.exe
Description	A detailed description of the detected activity and its context	(See below)

3.1 MITRE

The MITRE **column** contains the number of the technique identifier from the MITRE ATT&CK framework to which the rule is mapped, e.g. T1113 (Screen Capture) or T1547 (Boot or Logon Autostart Execution). MITRE ATT&CK is a publicly available knowledge base of real-world attackers tactics and techniques, which functions as a common, standardized vocabulary for cybersecurity teams.

Mapping each rule to a specific technique has several practical benefits:

- **Context for the analyst** - when the rule is hit, it is immediately obvious what phase of the attack and what type of behavior it is, without the analyst having to further research the context.
- **Coverage analysis** - from the mapping, it is possible to calculate which techniques the rule set covers and where there are detection gaps.
- **Reporting and communication** - enables reporting of findings and detection capabilities in a language that can also be understood by other teams as well as customers.
- **Link to threat intelligence** - findings can be correlated with the known behavior of specific malicious groups, that are mentioned in MITRE ATT&CK through their techniques.

3.2 Event Type

The Event Type **column** determines what type of telemetry the rule is based on, and also when its condition is evaluated. It's one of the six values:

- **OnProcessCreate** - the rule is evaluated every time a new process is created (typically a resource such as Sysmon Event ID 1 or Windows Security Event ID 4688). Fields such as process name, command line, parent process, hash, etc. are available. Useful for detection of execution of malicious binaries, legitimate but abused software (LOLBins), or unusual process chains.
- **OnEvent** - the rule is evaluated when a specific record occurs in the event log / ETW (e.g. PowerShell operating log, security log). It is the broadest type that covers various sources of events and their fields (payload).
- **OnRegistryValueChange** - the rule is evaluated when a value is created or modified in the registry (typically Sysmon Event ID 13). Suitable for detecting persistence, configuration modifications, and tampering with system settings.
- **OnFileCreate** – the rule is evaluated when a file is created or exists within the file system (e.g. SACL Event ID 4663 or MFT). Suitable, for example, for detecting malicious files of an attacker.
- **OnConnectionEstablished** – The rule is evaluated when a network connection is established (typically Sysmon Event ID 3). It allows you to detect connections to the attacker's malicious infrastructure, or legitimate – but abused websites.
- **OnDnsRequest** – the rule is evaluated at the DNS request (typically Sysmon Event ID 22). It allows you to detect translations of the domains of the attacker's malicious infrastructure, or legitimate – but abused websites.

This column also serves as an abstraction over the data source: it allows the same evaluation engine to process different types of telemetry in a uniform way.

3.3 Trigger Condition

The Trigger Condition **contains** the exact condition that will cause the rule to respond. It is a logical (Boolean) expression that is evaluated in terms of a specific event and returns true/false. It is the core of the rule - it defines its detection logic.

The condition consists of three basic parts:

1. Field references (by event type), e.g.:

- process.name, process.cmd, process.parent.name, process.hash - for OnProcessCreate,
- etw.id, etw.fields.<name>, etw.fields.payload - for OnEvent,
- registry.path, registry.value_name, registry.value_data - for OnRegistryValueChange.

2. Operators to compare values, for example:

- == (equals), != (not equal to),
- Contains
- Starts with / Not starts with,
- matches (regular expression match).

3. Logical conjunctions to combine multiple conditions: AND, OR, NOT and parentheses to determine priority.

Examples:

- process.name == 'psr.exe' – The rule responds when the process named psr.exe starts.
- etw.id == 'Microsoft-Windows-PowerShell.4103' A etw.fields.payload contains 'CopyFromScreen' - the rule responds when event 4103 occurs in the PowerShell operating log and its payload also contains the string CopyFromScreen.

The condition is intentionally machine-readable and unambiguous so that it can be automatically evaluated, versioned, tested and, if necessary, tuned (by narrowing it due to false positives).

3.4 Brief

The Brief column contains a short, human-readable name of the rule - basically a "sticker" that summarizes what the rule detects. Used for quick information: in the list of detections or in the dashboard, the analyst can see what is going on at first glance, without having to read the entire condition or detailed description. A well-formulated Brief significantly improves the sorting and prioritization of detections.

3.5 Description

The column Description contains a more detailed description of the rule. It explains what exactly the rule detects why a given case is suspect or relevant from the point of view of the user, in which context (tactics, techniques) it fits, and if necessary, false positives as well as

recommendations for further investigation. This column gives the analyst context right during the investigation and reduces the need to switch between tools and to search for information elsewhere.

4 How to read a rule (example)

To illustrate how the individual columns together form one complete rule, we present two examples focused on detection of screen capture.

Example A - Misuse of Psr.exe

Columns	Value
MITRES	T1113 (Screen Capture)
Event Type	OnProcessCreate
Trigger Condition	process.name == 'psr.exe'
Brief	Starting Psr.exe (Problem Steps Recorder)
Description	It detects the execution of the legitimate Windows tool "Problem Steps Recorder" (Psr.exe), which can automatically capture screenshots of the user's actions. Attackers can exploit it as a "living-off-the-land" tool to record the victim's activities without the need for their own malware.

How to read the rule: When a process is created (OnProcessCreate), the system compares the process name with a value of psr.exe. If they match, the rule responds, the range is referred to as the T1113 technique, and the analyst is shown a Brief and a Description as context.

Example B – Screen Capture via PowerShell

Columns	Value
MITRES	T1113 (Screen Capture)
Event Type	OnEvent
Trigger Condition	etw.id == 'Microsoft-Windows-PowerShell.4103' AND etw.fields.payload contains 'CopyFromScreen'
Brief	Screen Capture with PowerShell
Description	Detects an entry in the PowerShell operating log (event 4103 - module/pipeline logging), whose payload contains a call to the CopyFromScreen method. This .NET method is used to copy the contents of the screen into an image, which is a typical feature of a script performing a screen capture.

How to read the rule: when you check an event in the event log (OnEvent), the engine verifies that it is a PowerShell event with ID 4103 **and** that its payload contains the string CopyFromScreen. Both conditions must be met (AND) for the rule to respond.

5 Why is this structure essential for automatic forensic analysis

The set of rules designed in this way is a key tool for automated forensic analysis for several reasons:

Scalable, and consistent. Static, deterministic rules allow large volumes of telemetry from many systems to be processed repeatably and uniformly. The tool applies the same logic in the same way every time, regardless of the experience and fatigue of a specific analyst.

Encoding knowledge into machine-readable form. Each rule represents a piece of know-how about a known attackers technique, converted into a form that the machine can automatically evaluate. The knowledge of "how to do X" does not remain only in the head of the expert, but is captured, versioned and reused.

Explainable and auditable. Unlike a black box (e.g. statistical models) each of detection is an unambiguous action - a specific condition that is met. This is essential in forensic analysis, because findings must be defensible and demonstrable, and it is also appropriate to use them as evidence material.

Precision and fine-tuning. The Trigger Condition column defines the precise, testable logic. The rules can be narrowed or broadened in a targeted manner according to the environment, which is governed by the ratio between the detection of actual threats and the level of false alarms.

Uniform processing of different data sources. The Event Type column abstracts the type of telemetry (process, event log, register) so that the same engine can apply rules in the same way across different sources. Adding a new rule does not require a major rewrite, only the addition of a row to the catalogue.

Faster forensic analysis. The Brief and Description columns provide context directly at the moment of analyst's reaching it, so that the analyst does not have to keep an eye on the knowledge and can concentrate on the sorting and decision-making. Mapping to the MITRE ATT&CK reduces the results of the structure and makes it possible to measure the coverage of the technology.

Expandability. The Rules Catalog is a live document. With the discovery of new techniques, it is possible to add more rules, and the capabilities of the tool are constantly expanding without the need to rework its core.

6 Notes on Catalog Maintenance and Expansion

In order for the rule set to retain value, it is recommended to focus on:

- **consistent mapping to MITRE ATT&CK** - assign each new rule to a technique (or subtechniques) so that it remains valid for the coverage analyses,
- **clear and concise Brief names** - so that they remain usable for a quick triage,
- **documented false alarms** – state known false positives in the Description column,
- **versioning and testing of conditions** – test changes in Trigger Condition against known samples, so that no increase in false positives or false negatives is created,
- **periodic revisions** - the rules are continuously updated according to new knowledge and changes in telemetry (e.g. changes in the form of event logs).

7 Appendix

ADFIR-D22-KPB4-04-Appendix.xlsx