Sharing and Automation for
Privacy Preserving Attack Neutralization

(H2020 833418)

D4.3 Approach for capturing incident response and recovery Steps (M15)

Published by the SAPPAN Consortium

Dissemination Level: Public

H2020-SU-ICT-2018-2020 – Cybersecurity

Document control page

Document file: D4.3 Approach for capturing incident response and recovery steps
Document version: 1.0
Document owner: Mehdi Akbari Gurabi (FIT)

Work package: WP4
Task: T4.2
Deliverable type: Report
Delivery month: M15
Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document History:

Version	Author(s)	Date	Summary of changes made
0.1	Mehdi Akbari Gurabi (FIT), Tomas Plesnik (MU)	2020-06-01	Cyber security incident handling and response steps
0.2	Mehdi Akbari Gurabi (FIT)	2020-06-19	Preliminary document outline
0.3	Mehdi Akbari Gurabi (FIT)	2020-07-13	Preliminary Vocabulary, Interview Points
0.4	Mehdi Akbari Gurabi (FIT), Fabian Duenzer (FIT), Jonas Ruelfing (FIT)	2020-07-24	First Draft
0.5	Mehdi Akbari Gurabi (FIT)	2020-07-26	Ready-to-review Version
1.0	Mehdi Akbari Gurabi (FIT)	2020-07-31	Ready for submission

Internal review history:

Reviewed by	Date	Summary of comments
Sarka Pekarova (DL)	2020-07-27	check-reading content and grammar
Mischa Obrecht (DL)	2020-07-28	Content, structure, grammar
Martin Zadnik (CESNET)	2020-07-31	Technical

Executive Summary

The focus of Deliverable D4.3 is on proposing and developing an approach for capturing incident response and recovery steps by human operators. The goal is to understand and capture how cybersecurity analysts classify incidents, respond to them, and what are the recovery steps after an incident. This would be the input for recommending and automating the response and recovery process which will be developed in tasks T4.3 and T4.4 of the project.

This deliverable has a close relation with deliverable D4.1, formal methodology for modeling of response and recovery actions which is submitted at month 12 of the project and deliverable D4.2, vocabulary to express the captured knowledge which will be developed for until month 21.

This deliverable centers on prototype development of an approach for capturing response and recovery steps based on semantic technologies and Semantic Media Wiki platform. To show and test the environment, a preliminary vocabulary has been developed in this stage. Plus, the insertion of sample playbooks and incidents have been exercised. Evaluation of the feasibility and suitability of the approach supporting human operators to input incident-related knowledge, and environments that watch the interactions involve different roles such as security analysts and experts, computer security response incident team (CSIRT) members, and administrators is included in the deliverable as well.

Further, the privacy concern of sharing confidential data between organizations is discussed in the deliverable.

Table of Contents

1 Introduction

Incident response and recovery procedures are used to mitigate the effects of an attack. The incident response and recovery actions can be documented as playbooks as a set of general instructions to deal with a certain type of incident. Playbooks are usually organization-specific and not machine-readable. One of the main objectives of SAPPAN is to suggest a standard for interoperable and machine-readable playbooks, making use of semantic technologies, to enable organizations to share their knowledge. This deliverable is based on the deliverable D4.1 and has the goal to propose a proper approach regarding capturing the incident response and recovery steps.

The following chapter of the deliverable contains an overview of incident handling. The preliminary vocabulary is proposed in the ensuing chapter. After that, the implementation phase of our proposed approach on Semantic MediaWiki is described in the succeeding chapter, followed by a chapter about the evaluation and feedback. Last, a conclusion is given.

2 Overview

2.1 Incident Management Workflow

The incident management process and its workflow are generally organization-specific. Usually, Computer Security Incident Response Teams (CSIRTs) or Computer Emergency Response Teams (CERTs) are responsible for the handling of cybersecurity incidents. Incident management consist of four main phases: Preparation, Detection and Assessment, Mitigation, and Post-Incident Activity. [1]

Three main objectives of incident handling are:

- Strategic protection of the entire infrastructure and mitigate adversary damages
- Finding the adversaries, understanding their behavior and motivations, getting rid of them once they are in, and discovering of their possible return
- Ensure no confidential data is revealed while gaining as much information as possible

Typically, incident mitigations happen in the step after an incident identified. A preparation of cyber exercise playbooks is recommended by MITRE Corporation. [2]. MITRE is an American nonprofit organization working on cybersecurity solutions via federally funded research and development. MITRE also provides MITRE ATT&CK™, a globally-accessible knowledge base of a comprehensive and organized collection of adversary tactics and techniques based on real-world observations. (More information can be found here: https://attack.mitre.org/) A playbook is a set of instructions or steps prepared for handling a particular type of incident. These Playbooks are a formal documentation of mitigation workflows which could be organization-specific. Playbooks provide specific workflows (best practice guidelines) to mitigate certain incidents. Further, vocabulary to express an incident, the mitigation purpose, additional information on further threats, and the potential range of an attack are represented in playbooks. In addition to this incident-specific information, the cyber exercise playbooks should outline organization-specific information to conduct a certain exercise such as a responsible contact person, architecture-specific attacks information and preventions, etc.

These cyber exercise tasks have great potential to be automated, but they become more complex and difficult to automate in case of organization-specific workflows. A very important aspect of automation is the responsibility for automated actions. It consists of risk assessment for the consequences of a false action that can seriously damage an organization. A more detailed discussion about the issue should be considered as part of automation-related deliverables.

2.2 Existing approaches

In this section, we briefly mention some of the existing approaches for the incident response and recovery process. these widely influence the approach that we propose for capturing the incident response and recovery actions.

Approach	Notes
Unit 42 Playbook [3]	• STIX[4] is a language and serialization format for exchanging cyber threat intelligence. (More information can be founded in deliverable D4.1 and here: https://oasis-open.github.io/cti-documentation/stix/intro) • STIX 2.0 does not have a native playbook approach • Unit 42 is a playbook accessed by STIX 2.0 • Created playbooks queryable by elements below Unit 42 playbook sorted by the adversary and filterable by malware, country, and industry • Response and recovery handling: Instructions in Courses of Actions can be added collaboratively, Only details adversaries´ actions in Unit42 playbook, no response or recovery
IncidentResponse [5]	• A community focused on incident response, security operations and recovery process • Provide opensource playbooks for different category of attacks • These playbooks are often too general but useful to derive more specific playbooks
MISP	• MISP is a sharing platform (More information can be founded in deliverable D4.1 and here: https://misp-project.org) • MISP taxonomy [6] has no general playbook • Actions are taken in response to an event linked to it can be added collaboratively in MISP • MISP Galaxy [7] provides simple mitigation actions
VERIS [8]	• Vocabulary for Event Recording and Incident Sharing • Based on risk management • Has estimations for possible impacts to determine occurrence probabilities

2.3 Handling cybersecurity incidents by SAPPAN members

This section describes cybersecurity incident handling and response steps. Below are described different approaches and related issues of handling cybersecurity incidents from different organizations participated in the SAPPAN project. Information is generalized and we avoid mentioning organization-specific approaches due to the confidentiality level of information in different organizations.

2.3.1 Current reporting tools

Organization 1:

- Automated evaluation of URLs
- Enterprise data warehouse feeds going to *Security Information and Event Management Systems (SIEMs)* and events to the data warehouse
- Using an issue tracking software for case management
- Operator creates ticket
- No specific post-incident tool but case management system used.

Organization2:

- Using open-source sharing tools
- Own rule engine
- New process for custom reports in the works

Organization3:

- Detection phase
    - External email and ticket are created and handled
    - Internal reporting of detection via an internal sharing tool
    - Detection with internal IP network traffic monitoring tool and network behavior analysis tool using flow data
- Assessment phase
    - Verified in internal IP network traffic monitoring tool
    - Internal reputation database: future misbehavior probability score, previous reports affiliated with the IP address, presence on blacklists, domain name, autonomous system number, geolocation, tags - prevalent behavior, whois information, passiveDNS
    - Using passiveDNS
- Mitigation phase
    - BGP FlowSpec (Information available at https://tools.ietf.org/html/rfc5575) to block, rate-limit or divert traffic to DDoS mitigation device
- Post-incident
    - Deep analysis of flow data
- Preparation phase
    - Introducing rate limitations for certain combinations of packets - e.g. limit on DNS or NTP fragmented packets (to prevent DNS/NTP amplification attacks)

Organization 4:

- Customized Request Tracker based on Best Practical Request Tracker (Information available at: https://bestpractical.com/request-tracker)
- Integrated other tools used in the incident handling process (whitelist/blacklist databases - IP/DNS/Users/E-mails, WHOIS database, nslookup, etc.)
- Detection and assessment phase
    - Detection methods based on NetFlow data (Scan detection, Brute-force attacks, DDoS detection, User-defined patterns, etc.) and system logs

  - Users and admins also report incidents that they noticed
  - Detect vulnerable machines in the network based on exposed vulnerabilities
- Mitigation phase
  - To mitigate the impact of incidents, block IP and e-mail addresses and users in the network
  - Use a DNS Firewall
  - To fix a vulnerable or abused machine, contact its administrators and give them information about the incident, sometimes with actions that should be taken to fix the machine
- Post-incident activity phase
  - Perform analysis of an incident, the responses of users or admins, and their responses and processes
- Preparation phase
  - Actions that are taken in preparation, e.g., blocking IP addresses used to distribute malware (based on information from warnings)
  - Setting up detection patterns on newly discovered vulnerabilities (under the assumption that they can be detected from NetFlow data)

2.3.2 Current incident response and recovery actions

Organization1:

- Not included, handler read playbook and act.
- Playbooks are editable and dynamic, only for privileged users
- Playbooks are text.
- Playbook for a specific incident, E.g., a malware itself.
- Playbooks to prevent incidents, not vulnerabilities

Organization 2:

- Informing people responsible for information on how to respond.
- Has Playbooks for the recovery phase and shares with customers.

Organization 3:

- Technical response - triggered by organizations (only for their data) or organization's admins/CERT.
- Contact local admin in a subnetwork.
- Automated actions for very simple and specific cases, but there is no automation for complex cases.
- A distributed modular SIEM: receives data from sharing system, compares IP addresses with contact DB, email alert to administrators
- Other incidents: Email-based, handlers are used to do all communication and notes via emails, not going to migrate to a proper ticket system GUI

Organization 4:

- Response and recovery actions are described in Incident Handling (IH) playbooks for cybersecurity incident handlers.
  - There is currently only one textual manual that is edited if necessary.

  - The IH manual is edited and updated when necessary, e.g., a process is changed, or a new process is added.
  - The IH manual is confidential.
  - We can automatically resolve the majority of generic incidents where the adversary is from the outside of the university network. This includes detection, reporting, and mitigation.

2.3.3 Summary of Limitation and pain points

- Time-consuming, includes manual process, e.g., Identification based on the IP: IP → Host Name → Person
- Repetitive manual processes
- Flooding of incident reports from users during large-scaled campaigns (such as phishing)
- Missing playbooks! can be potential action to handlers, now handler should read the manuals and if something beyond his/her skills it should be handled by central domain experts
- General incidents are described in the manual. If there is an entirely new incident or perhaps a different version of a general incident, a senior member is usually required to define the steps that need to be taken to a junior member.

2.3.4 Alternative approaches

- While an organization was looking for alternatives and replacements to their request tracker, they tried the following tools:
  - TheHive (https://thehive-project.org/): a powerful open-source platform with many possibilities – many of those are not interesting to us. It is quite heavy thanks to the integration of the analytical Cortex engine.
  - Django-helpdesk (https://github.com/django-helpdesk/django-helpdesk): rather simple helpdesk tool, not as customizable as Request Tracker
  - OpenProject (https://www.openproject.org/): possible replacement (or rather an extension) of their current workflow. It has better handling of links/dependencies than Request Tracker, and it is fairly customizable.
- Advantages: All of the mentioned tools are written in common programming languages (Python, Ruby). The codebase also seems to be much cleaner than the organization's current request tracking approach – it should be easier to tinker with core functionality if necessary.
- Disadvantages: Organization's current Request Tracking approach is probably slightly more customizable friendly than the mentioned alternative tools (in terms of simplicity and learning curve thanks to the support of custom scrips).

2.3.5 Conclusion on cybersecurity incidents handling in SAPPAN consortium

In conclusion, various incident reporting tools are in use in different organizations. A ticketing system is an important approach for capturing incident reports which have the potential for automation on the process. For the incident response and recovery process, similarly, diverse approaches are in use in different organizations and for different expertise levels. As the main lesson that we learned from the SAPPAN members, the response and recovery approaches are vastly varied organization by organization. There is no standard commonly in use for different organizations to structure their playbooks, and there are very limited approaches to share the playbooks between separated organizations. Lack of sharing methods is mainly due to the sensitive information that reveals in the playbooks. Currently, no proper approach is considered in the SAPPAN members to separate confidential and general information to and share the non-sensitive and general part of playbooks. The main weakness in the response and recovery approaches is the lack of properly structured playbooks which can be used and updated by incident handlers without the requirement of high-level cybersecurity experts involved in the process. Also, the steps of a playbook would be reused by automation tools, accordingly, the results should be machine-readable. Further, existing approaches are time-consuming because of repetitive manual processes that can be majorly automated.

3 Preliminary Vocabulary

For developing the capturing environment, modeling the playbooks, and addressing restrictions, it was necessary to firstly develop a preliminary vocabulary. In this regard, the playbook vocabulary is developed mainly based on the deliverable D4.1 and is made applicable by Semantic MediaWiki (SMW) a framework that is briefly described in the Implementation Phase section. The rest of the vocabulary is not transferred directly from any existing tools, but it is influenced by UCO [9], STIX [4], MISP taxonomy[6]/galaxy [7], and OTX pulses [10] and by trying to transfer Malware-breakdown [11] samples.

The preliminary vocabulary is not expressing every aspect of incident handling and has limited categories for incident reporting. The preliminary vocabulary has used as a mock vocabulary for implementing the capturing approach and has been slightly amended after the feedback from domain experts. It will be extended and amended for deliverable D4.2, based on the domain experts' feedback considering the expression of the playbooks and incidents, and dealing with e.g., restrictions, ambiguousness of semantics, user experience, and privacy requirements.

In the following, you can find tables for each class, a short description of it, properties that can express it, type of each property, and the cardinality level. The types in bold refer to another class in the vocabulary and connect a class to another one by a property. For example class incident has a property "hasIndicator" and it refers to an instance of IndicatorOfCompromise class that expresses an IoC in detail with different types, values, reporting time, etc. The cardinality shows the number of instances of each property can be expressed for an entry. Cardinality "1" shows a mandatory field that should be filled exactly once, "≥ 1" shows a mandatory field with multiple inputs, "≤ 1" is for optional single input fields, "≥ 0" expresses optional multiple input fields, and "≥ 2" describes a field with at least 2 mandatory inputs.

Class	Description	Properties	Type	Cardinality
Incident	This class is the main class to detect and report an incident	hasName	string	1
		hasIndicator	**IndicatorOfCompromise**	≥ 1
		hasPlaybook	**Playbook**	≤ 1
		hasAttackCategory	**AttackCategory**	≥ 1

		hasExploitType	**ExploitType**	≥ 0
		hasAttackConsequences	**AttackConsequences**	≥ 0
		hasAdversaryActor	**AdversaryActor**	≥ 0
		hasAttackMeans	**AttackMeans**	≥ 0
		hasVulnerability	**Vulnerability**	≥ 0
		hasTAG	**Other classes,** string	≥ 0
		hasReport	**IncidentReport**	1
		relatedIncident	**Incident**	≥ 0
		isActive	boolean	1
		hasSnapshot	file	≥ 0

Class	Description	Properties	Type	Cardinality
Adversary-Actor	This class represents the identification and characteristics of the attacker	hasName	string	1
		hasID	string	≤ 1
		hasType	string	≤ 1
		hasMotivation	{Revenge, PersonalGain, OrganisationalGain, Accidental, Unpredictable, Other}	1
		hasFirstReportTime	timeStamp	1
		hasLatestReport-Time	timeStamp	1
		hasLocation	string	≤ 1
		hasRole	string	≤ 1
		hasAliase	string	≥ 0

		hasAlly	**AdversaryActor**	≥ 0
		hasTarget	**Vulnerability**	≥ 0
		causeIncident	Incident	≥ 1

Class	Description	Properties	Type	Cardinality
AttackCategory	This class represents the category of the attack. E.g., Phishing	hasType	{Phishing, Domain Generation Algorithm, Other}	1
		hasName	string	1
		forIncident	**Incident**	≥ 1
		relatedPlaybook	**Playbook**	≥ 0

Class	Description	Properties	Type	Cardinality
AttackMeans	This class represents the method of executing an attack and characterizes the observable details and tactics, techniques and procedures of malicious behavior	hasType	{BufferOverFlow, LogicExploit, SYNFlood, TCPPortScan, Other}	1
		hasSubCategory	string	≤ 1
		hasName	string	1
		hasDescription	string	1

Class	Description	Properties	Type	Cardinality
Vulnerability	This class represents the vulnerabilities or weaknesses in an attack point, e.g., network or end point	hasType	{NetworkVulnerability, EndPointVulnerability, Other}	1
		hasSubCategory	string	≤ 1

		hasName	string	1
		hasDescription	string	1

Class	Description	Properties	Type	Cardinality
ExploitType	This class describes the characteristics of an exploit	hasType	string	1
		hasName	string	1
		hasDescription	string	1

Class	Description	Properties	Type	Cardinality
Attack-Consequences	This class describes the possible results of an attack. E.g., Denial of Service	hasName	string	1
		hasType	{DenialOfService, LossOfConfiguration, PrivilegeEscalation, UnauthorizedUser, Other}	1
		hasSubCategory	string	≤ 1
		hasDescription	string	1

Class	Description	Properties	Type	Cardinality
IndicatorOfCompromise	This class represents a cyber threat indicator regarding the pattern, observable conditions, hash value, etc. of an incident	isIndicatorOf	**Incident**	1
		hasValue	string/etc	1
		hasCreationTimeStamp	timeStamp	1
		hasSize	string	≤ 1
		hasAttackCategory	**AttackCategory**	≥ 0

		hasType	{FileHashMD5, FileHashMD6, FileHashSHA1, FileHashSHA224, FileHashSHA256, FileHashSHA384, FileHashSHA512, Domain, HostName, IPv4, IPv6, URL, Email, YaraRule, Other}	1

Class	Description	Properties	Type	Cardinality
IncidentReport	This class represents the identification of the reporter and meta data regarding date and etc.	hasReporter	**Reporter**	1
		hasCreationTime	timeStamp	1
		hasLocation	string	≥ 0
		hasModified	boolean	≤ 1
		hasModificationTime	timeStamp	≥ 0
		hasRemediated	boolean	≤ 1
		hasRemediationTime	timeStamp	≤ 1

Class	Description	Properties	Type	Cardinality
Reporter	This class represents the reporter of an incident	hasID	string	≤ 1
		hasName	string	1
		hasAge	integer	≤ 1
		hasNationality	string	≤ 1
		hasGender	string	≤ 1
		hasRole	**{**Admin, Security Analyst, CSIRT, LEA, Forensics Expert, Other**}**	1
		hasOrganisation	string	≤ 1
		hasContactInfo	string	≥ 0

Class	Description	Properties	Type	Cardinality
Playbook	This class represents the course of actions of a playbook and contain the steps	forIncident	**Incident**	≥ 1
		forAttackCategory	**AttackCategory**	≥ 1
		toMitigate	**Vulnerability**	≥ 0
		hasInitialStep	**InitialStep**	1
		hasFinalStep	**FinalStep**	1
		hasIntermediateStep	**IntermediateStep**	≥ 1
		hasExclusiveChoiceStep	**ExclusiveChoiceStep**	≥ 0
		hasOptionalStep	**OptionalStep**	≥ 0
		relatedIndicator	**IndicatorOfCompromise**	≥ 0

		hasConfidentialityLevel	{FullyConfidential, PartiallyConfidential, Public}	1
		hasAuthor	**Reporter**	≥ 0

Class	Description	Properties	Type	Cardinality
InitialStep	This class represent the starting point of a playbook	hasName	string	1
		hasDescription	string	≤ 1
		hasNextStep	**{IntermediateStep, ExclusiveChoiceStep}**	≥ 1
		isStepOf	**Playbook**	≥ 1

Class	Description	Properties	Type	Cardinality
FinalStep	This class represent the end point of a playbook	hasName	string	1
		hasDescription	string	≤ 1
		hasPreviousStep	**{IntermediateStep, ExclusiveChoiceStep}**	≥ 1
		isStepOf	**Playbook**	≥ 1

Class	Description	Properties	Type	Cardinality
IntermediateStep	This class represent the intermediate steps of a playbook	hasName	string	1
		hasDescription	string	≤ 1
		hasMean	{Detection, Containment, Remediation, Recovery}	1

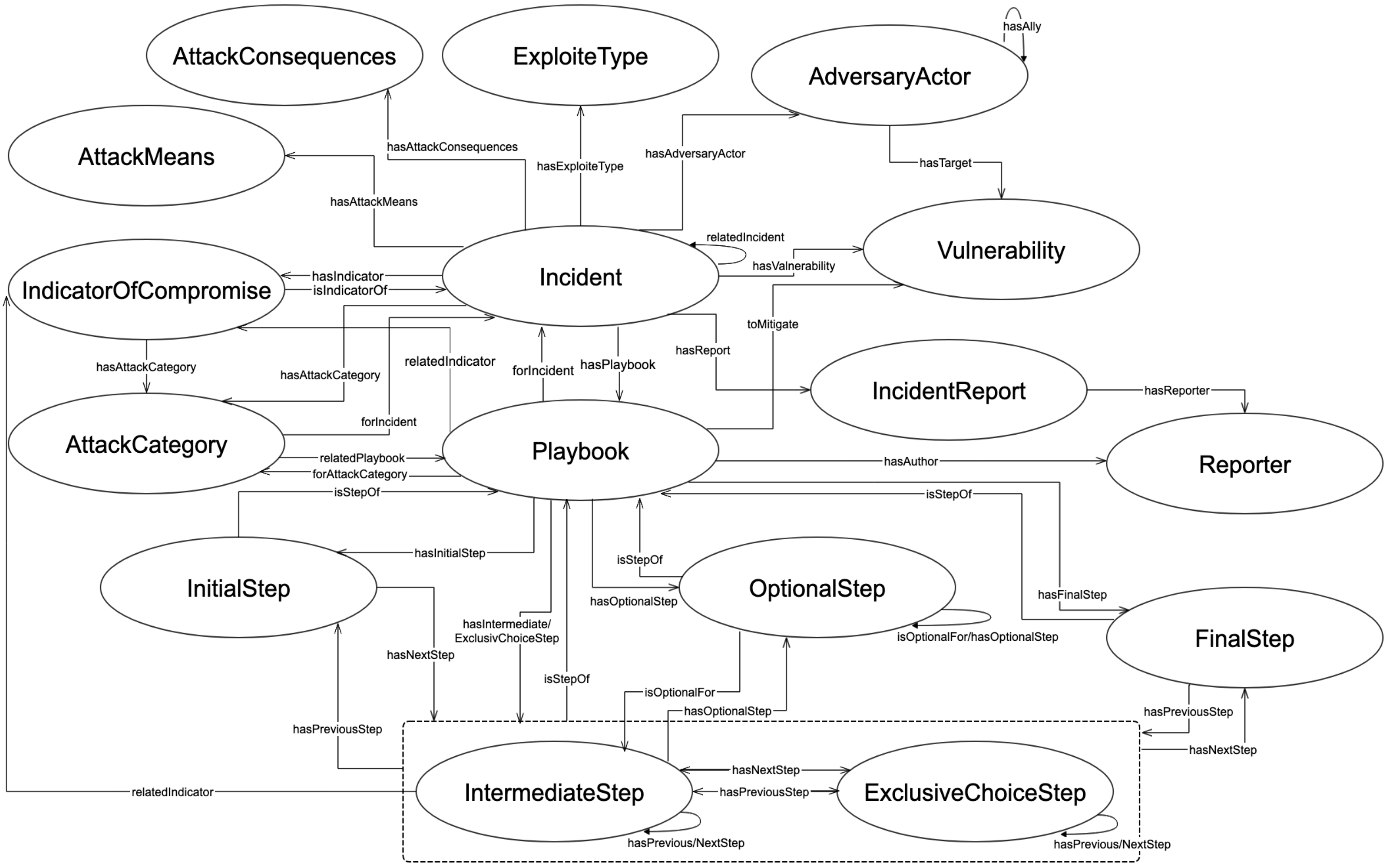
		isStepOf	**Playbook**	≥ 1
		hasPreviousStep	**{InitialStep,** **IntermediateStep,** **ExclusiveChoiceStep}**	≥ 1
		hasNextStep	{**IntermediateStep,** **ExclusiveChoiceStep,** **FinalStep**}	≥ 1
		hasOptionalStep	**OptionalStep**	≥ 0
		relatedIndicator	**IndicatorOfCompromise**	≥ 0
		hasConfidentialityLevel	{FullyConfidential, PartiallyConfidential, Public}	1

Class	Description	Properties	Type	Cardinality
ExclusiveChoiceStep	This class represent the steps of a playbook with an exclusive choice	hasName	string	1
		hasDescription	string	≤ 1
		hasMean	{Detection, Containment, Remediation, Recovery}	1
		isStepOf	**Playbook**	≥ 1
		hasPreviousStep	**{InitialStep,** **ExclusiveChoiceStep,** **IntermediateStep}**	≥ 1
		hasNextStep	{**IntermediateStep,** **ExclusiveChoiceStep,** **FinalStep**}	≥ 2

		hasConfidentialityLevel	{FullyConfidential, PartiallyConfidential, Public}	1

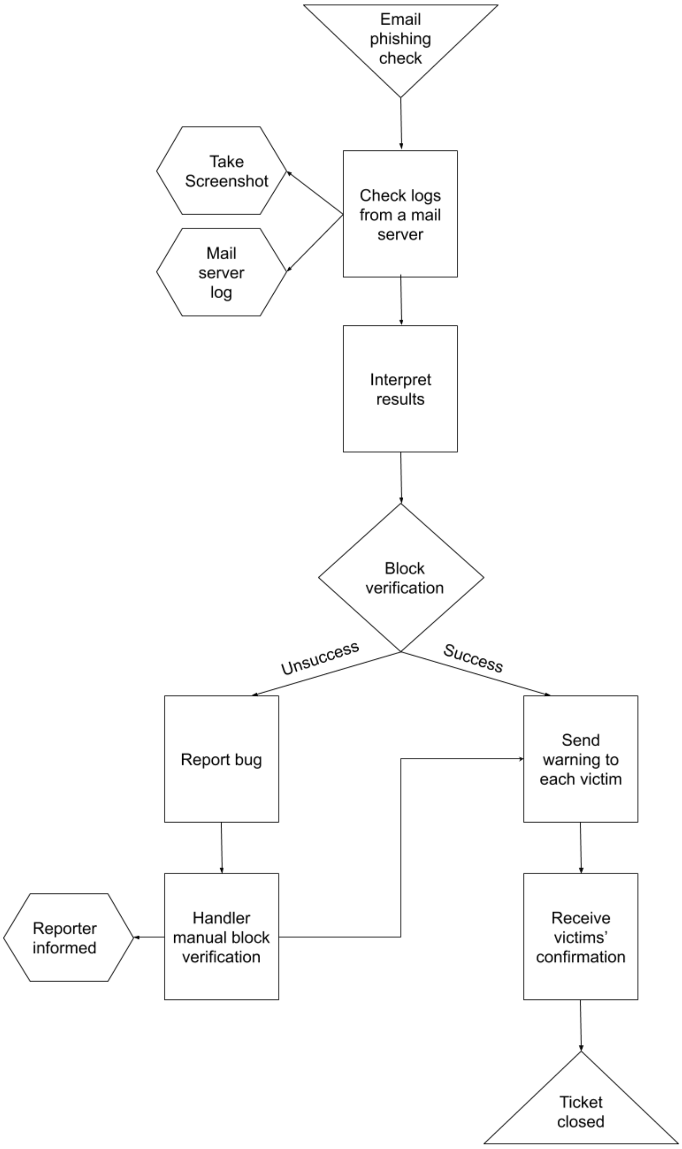
Class	Description	Properties	Type	Cardinality
OptionalStep	This class repre-sent the optional steps of a playbook	hasName	string	1
		hasDescription	string	≤ 1
		hasMean	{Detection, Containment, Remediation, Recovery}	1
		isStepOf	**Playbook**	≥ 1
		hasOptionalStep	**OptionalStep**	≥ 0
		isOptionalFor	{**OptionalStep,** **IntermediateStep**}	≥ 1
		hasConfidentialityLevel	{FullyConfidential, PartiallyConfidential, Public}	1

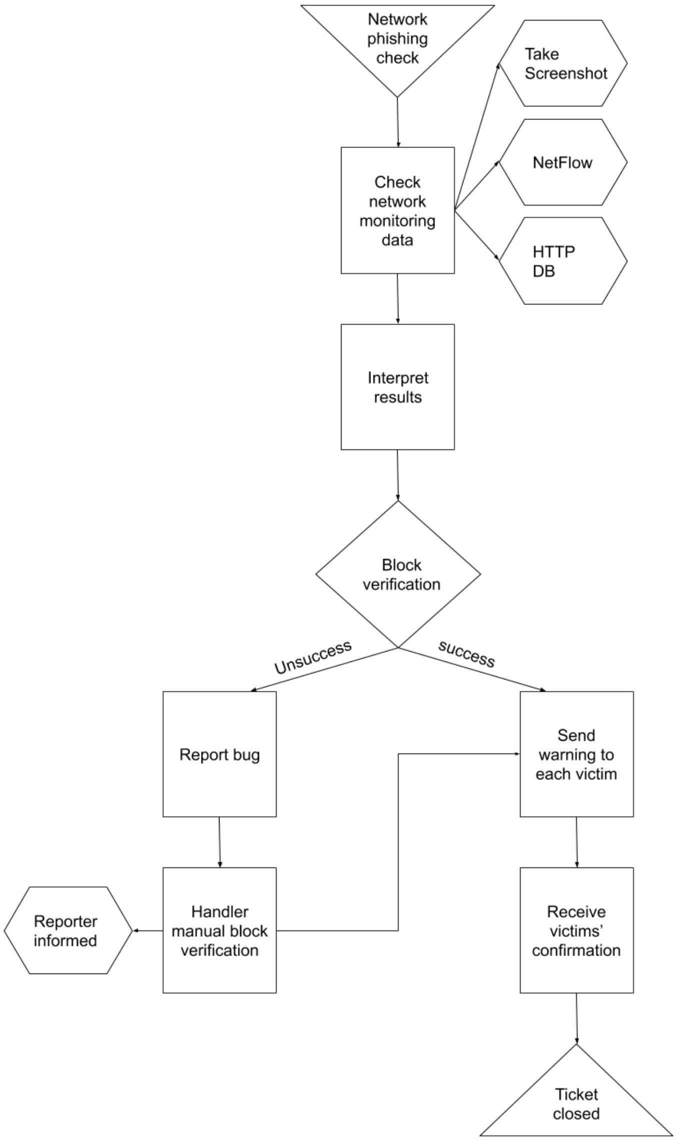
The following figure shows the overall class relationship diagram. The diagram in not including all the properties for each class, but only consists of the properties that connect a class to another one.



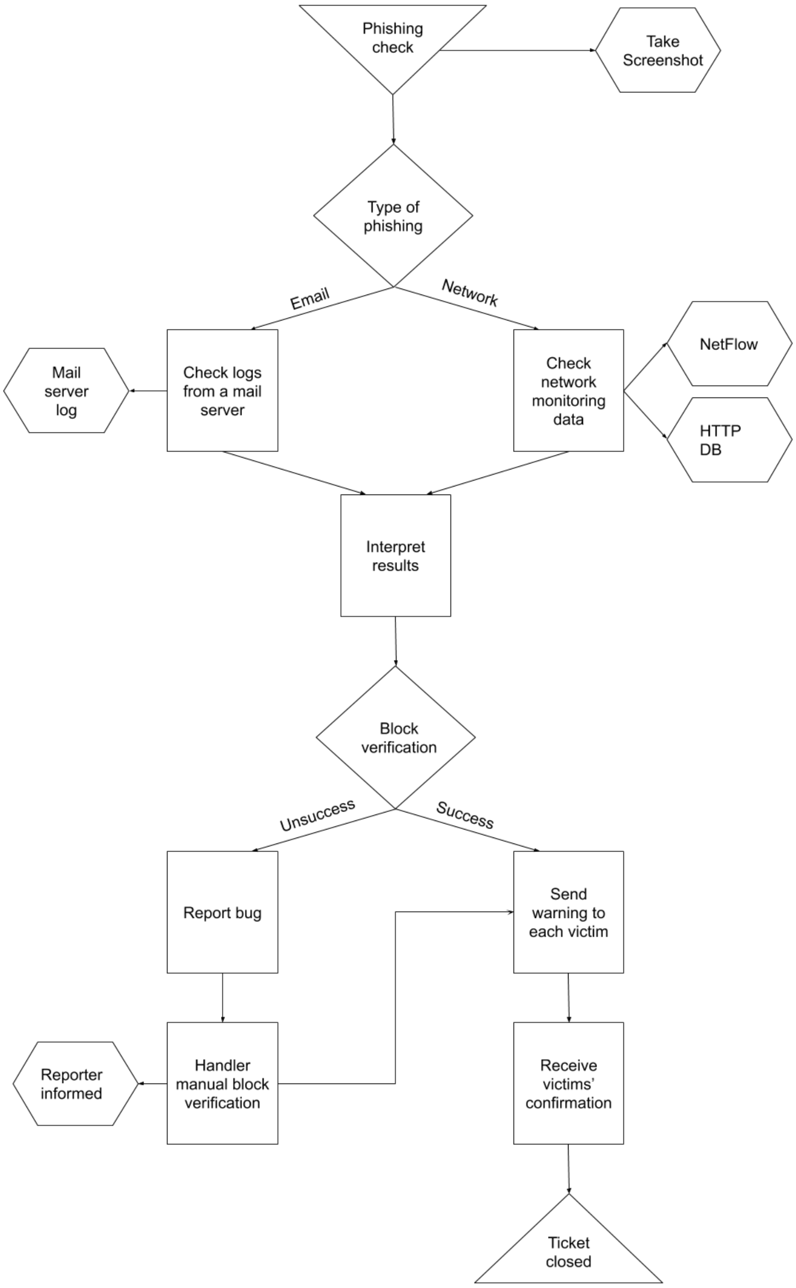
3.1 Sample Playbooks

Sample playbooks are provided as general playbooks for phishing attacks. The diagrams are illustrating the playbooks based on the classes that are defined in the deliverable D4.1. Each playbook has an Initial and a Final step represented by triangle boxes. Separating Initial and Final steps from other steps will help us in the automation process by checking the reachability of the Final step by the Initial step of a playbook to avoid dead ends. Each Initial step links to at least one Intermediate or Exclusivechoice step by a property "nextStep". Also, each Final Step is linked to the previous step(s) by "previousStep" property. Intermediate steps are represented by squares that link to previous and next steps by "previousStep" and "NextStep" properties. Also, OptionalSteps are represented by Hexagonal boxes and are Linked to Intermediate steps by "isOptionalFor" property. OptionalSteps can also link to next OptionalSteps by "hasOptionalStep" property. ExclusiveChoice step is shown by a diamond shape box that links to at least two steps by "nextStep" property. Other than that, the cardinality of "previousStep" and "nextStep" is at least one for all the steps. Multiple assignments to these properties will enable parallelization in the playbook steps. The diagrams are not showing a workflow playbook and they are only visualizing the sequence and semantic class of steps. Each shape represents a different structural element defined in the formal methodology (e.g., an initial step or an optional step) which allows us to map the vocabulary terms to these elements. Visualization of the playbooks will be discussed in "Implementation Phase "and "Lesson Learned and Feedback on Capturing Tool" sections. Two main playbooks for separate phishing categories are defined here: Phishing email attack and phishing URL attack. For each type, a simple playbook is provided as follows.





As it is shown in the sample Email and URL phishing playbooks, most of the steps are the same. It enables merging playbooks by an ExclusiveChoice step which allows the branching for these two types of phishing attacks. The merged playbook can be represented as follow:



3.2 Privacy concerns

In the formal methodology, it is recommended to support two confidentiality levels (Public/Confidential) by having two classes. This could be arbitrarily refined without changing any of the other definitions because it is planned to only be modeled as class membership. For the implementation, based on the slightly different structure of semantic media wiki, the vocabulary is refined and the confidentiality level has been defined via the property "hasConfidentialityLevel" for any Playbook, IntermediateStep, ExclusiveChoiceStep, and OptionalStep classes with three possible values (Fully Confidential, Partially Confidential and Public). These confidentiality level categories can be simply extended by defining more specific confidentiality levels. Access restriction can be applied based on the confidentiality levels. Currently, there is no implementation for the access restriction, but the access control module will be described in the Implementation phase section.

As the issue is mentioned in Deliverable D4.1 when confidentiality levels are applied to steps, if all steps are public, except for a single intermediate step that is fully confidential, it might be non-trivial to remove this step from the playbook (e.g., if an action that is necessary for subsequent steps is associated with this confidential step). A solution to this would be to replace such a step by an "empty step", that does not directly reveal any information, but that lets the receivers of the playbook know that something is missing. For the partially confidential steps, only not sensitive values should be shared. In [1] it is suggested to share only non-sensitive properties of each playbook even in case of public playbooks to avoid sharing of properties that could potentially reveal organization-specific data, e.g., reporter contact info or text description of a playbook.

Regarding the confidentiality level of the playbooks or each step, it is also possible to define different classes for each confidentiality level and assign the resources to corresponding classes. It is consequently possible to use it primarily to information assigned to steps (and not steps themselves). In that case, sensitive information could be removed more easily and playbooks can be shared more flexibly.

4 Implementation phase

The main descriptions and advantages of using the Semantic web and Knowledge graph for the context of cybersecurity incident response and recovery were described and a proposed formal methodology was presented in the deliverable D4.1. In this deliverable, we describe a framework using semantic technologies to define vocabulary and relations in the cybersecurity incident response and recovery domain. We decided to use Semantic MediaWiki extension for rapid prototyping, presenting and evaluating our approach and vocabulary for capturing incident response and recovery actions to offer structured, machine-readable, human-readable, interoperable, and scalable playbooks. The main objective of this prototype is to evaluate the feasibility to apply the proposed formal methodology in deliverable D4.1 to develop our ontology and gain information about specific criteria and domain expert requirements that development would meet.

Based on the next actions and feedback gathered from domain experts, we may reassess using the SMW prototyping approach and switch user interface, extensions, and frameworks to fulfill the future requirements.

4.1 Semantic MediaWiki

Wikis are well-known tools for collecting and sharing human-readable knowledge in communities. However, they are usually not machine-readable and not useful for getting queried or aggregated information. SMW [12] is a free and open-source extension to MediaWiki which applies semantic technologies to a wiki that can make it a knowledge management system with machine-readable relations between the context of the wiki.

Data created within SMW can be exported and published via the Semantic Web which makes reusing the data feasible in other systems. The prototype implementation of the response and recovery approach on SMW is available in the consortium-internal

GitLab repository. For access to this repository, please request login credentials via an email to info@sappan-project.eu.

The development is ongoing and the final version will be provided with the final vocabulary on deliverable D4.2.

4.2 Backend Choices

SPARQL/RDF based backends:

Originally, MediaWiki is using a MySQL backend for data storage. This can limit the advantages of using SMW because a relational database does not represent easily the graph structure which is utilized in SMW. Hence, it is possible to utilize an RDF or SPARQL based backend, which allows a better representation of the data and easier querying. Following, information related to these types of backends can be found here: https://www.semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores

Generally, an RDF store is used next to a normal relational database, i.e. not replacing the original backend. While it is desirable to have an RDF/SPARQL based backend, there are disadvantages as:

- Redundancy of the data and need for maintenance of two systems
- Experience with this type of backends is not mature enough and there might be problems with performance and stability

RDF store is separate from normal relational DB backend and different connectors exist for that different backends.

Following, several SPARQL/RDF based backends that are introduced by SMW are listed [13].

- 4Store (**outdated**)
  - used with SMW (2011)
  - last release 2015, no GitHub activity
  - Connector: https://www.semantic-mediawiki.org/wiki/Help:SPARQLStore/RepositoryConnector/4store

- Virtuoso
  - https://www.w3.org/wiki/VirtuosoUniversalServer
  - Latest release Oct. 2018. But GitHub is continuously updated
  - used with SMW (2011)
  - can hold 58 B triples
  - Connector: https://www.semantic-mediawiki.org/wiki/Help:SPARQLStore/RepositoryConnector/Virtuoso
  - Virtuoso on SMW, last updated Feb 2020

- Blazegraph
  - Connector: https://www.semantic-mediawiki.org/wiki/Help:SPARQLStore/RepositoryConnector/Blazegraph

  - Ongoing releases

- Fuseki
  - Connector: https://www.semantic-mediawiki.org/wiki/Help:SPARQLStore/RepositoryConnector/Fuseki
  - Apache Jena Fuseki, SPARQL server for Jena: https://jena.apache.org/documentation/fuseki2/
  - Ongoing releases

- Sesame
  - Connector: https://www.semantic-mediawiki.org/wiki/Help:SPARQLStore/RepositoryConnector/Sesame
  - Seems to be Eclipse RDF4J now
  - Ongoing releases

It needs to be considered that category/property hierarchies are only supported if RDF store supports: rdfs:subClassOf and rdfs:subPropertyOf features of RDF Schema.

Following, a list of general advantages, disadvantages, and alternatives for the usage of an RDF/SPARQL backend can be found.

Advantages:

- Existing connectors for some of the databases
- Definition of connector is easy in principle
- SPARQL/RDF is better suited to model SMW than a SQL database
- Allows executing of SPARQL/OWL queries

Disadvantages:

- Not sophisticated solutions for SMW with RDF
- The database is mirrored (redundancy, maintaining two databases, ...)
- Not a lot of experience available, the risk about the stability of the solution, no general usage
- The choice of backend might have strong consequences. No native support of backends for SMW

Alternatives:

- RDFIO (https://github.com/rdfio/RDFIO) is an extension that could be used instead of an RDF/SPARQL backend. It allows the import of arbitrary RDF triple, OWL ontologies, and offers a SPARQL endpoint with a relational backend. It utilizes a PHP/MySQL based triple store and SPARQL endpoint and is based on the ARC2 PHP library. It currently supports up to SMW 2.5. RDFIO directly connecting the own SPARQL backend allows more freedom for the creation/handling of the SPARQL database. But also more responsibilities, these connectors could be able to save a lot of headaches by taking care of most functionalities. RDFIO for example seems to be not based on such a sophisticated database and it uses a PHP triple store. More information about RDFIO backend could be found here: https://www.mediawiki.org/wiki/Extension:RDFIO

- ElasticStore is another popular alternative for the backend. This backend is not an RDF compatible backend, but it provides a powerful and scalable query engine and retrieves information from Elasticsearch instead of the default SQL-Store. This introduces as part of Semantic MediaWiki 3.0. [14] Using Elasticsearch is considered for next developments, because it might be selected as the search backend of SAPPAN dashboard.

In conclusion, for the initial version of the approach, we decided to use the default relational DB backend and figure out the limitations and pain points. In a case of complexities or a lack of feasibility, we will migrate to a more suitable backend for developing the complete vocabulary.

4.3 Semantic MediaWiki on Docker

Docker is a free software for isolating applications with container virtualization. Docker simplifies application delivery because containers contain all the packages that can be easily transported and installed as files. It makes the testing and development process faster and less complicated without consideration of the native operating system. For deployment of SMW on docker we use one of the existing recommended docker solutions on the official Semantic MediaWiki website. The solution is a composer based with MariaDB and Nginx which is provided here: https://github.com/toniher/docker-SemanticMediaWiki

4.4 User roles and access restrictions

Natively, MediaWiki is not supporting advanced access control. Different extensions offer advanced functionality, but there is always the risk because of the fundamental design of MediaWiki. More detailed information on the problems arising when trying to restrict access can be found under https://www.mediawiki.org/wiki/Category:Page_specific_user_rights_extensions

Possible extensions with which were experimented are:

- UserGroups
- SemanticAuthProfiling
- Semantic ACL
- Lockdown

Lockdown was showing the most promising results and can restrict access to different namespaces. This would enable the creation of different namespaces for different consortium members and, afterward, assigning specific rights to a single user or user groups.

An example of the definition of a namespace and restriction of reading access to all users is listed below. It has to be added to SMW LocalSettings.php directly or indirectly with an inclusion.

- Define namespace and corresponding talk page.
- Define unique namespace id, has to be even, talk page id+1.

```
define(“NS_MY”, 4000);
define(“NS_MY_TALK”, 4001);

$wgExtraNamespaces[NS_MY] = “My”;
$wgExtraNamespaces[NS_MY_TALK] = “My_talk”;
```

- Important for prevention of inclusion by other page thus enabling read.

```
$wgNonincludableNamespaces[] = NS_MY;
$wgNonincludableNamespaces[] = NS_MY_TALK;
```

- Change permissions for specific user or user groups:

```
$wgNamespacePermissionLockdown[NS_MY][‘read‘] = [ ‘user‘ ];
$wgNamespacePermissionLockdown[NS_MY_TALK][‘read‘] = [ ‘user‘ ];
```

- Pages in this namespace can then be accessed via: $IP/wiki/My:$Pagename

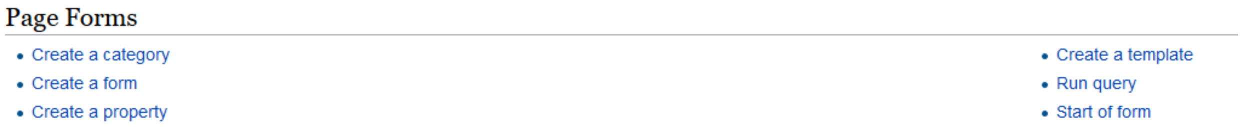
This extension will be useful for the restriction of view for confidential data. Visibility of confidential pages of an organization can be restricted by corresponding namespaces that would be only reachable by organization members.

Another solution to avoid revealing confidential data is to install an instance of SMW in each organization separately. Therefore organizations can share the non-sensitive results in such a format e.g., JSON as reusable data. Available export formats will be discussed in the Export Result Formats subsection.

4.5 Creating Properties, Templates and Forms Based on the Mock Vocabulary

There is an extension to MediaWiki that enables users to create forms, and inserts, edits, and queries on semantic data using forms. It was initially created for Semantic MediaWiki as the so-called "Semantic Forms" extension to edit and store SMW templates parameters. Later, it was extended for other usages for MediaWiki and renamed to the "Page Forms" extension. [15]

In our MediaWiki On the page "Special:SpecialPages" under Page Forms, there are links to create properties, templates, and forms.



Page Forms

- Create a category
- Create a form
- Create a property
- Create a template
- Run query
- Start of form

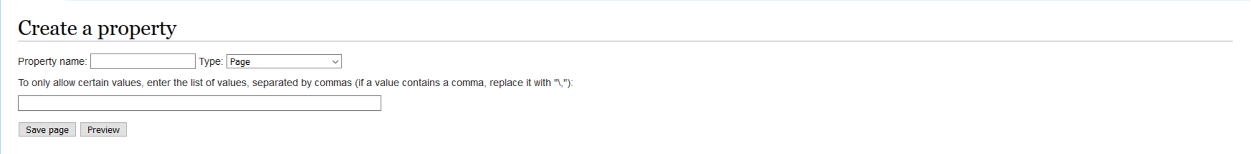
Similarly, after their creation all properties, templates and forms are listed on "Special:Properties", "Special:Templates", and "Special:Forms" respectively.

For every class in the vocabulary the process is as follows:

1. For every property in the class, see if a property in the wiki exists for that type. If not, create that property.

2. Create a template that determines the properties contained in and look of a class' page.
3. Create a form that we fill with the properties for a new page of this class.

A property corresponds to types in the vocabulary. For example, for AttackCategory we needed to create the types Incident and Playbook. As these types indicate classes where an entry gets its page itself, the property's Type is page. For the more basic types (string, boolean, integer, timestamp, etc.) there is a corresponding type in the dropdown menu. For properties that only allow certain values, these can either be set here or later in the form, depending on whether it's a restriction on the type or the class in the vocabulary. For example, the property "hasType" in AttackCategory and other classes doesn't have a specific type and every class allows for different values for this so instead of creating a type property for every class we use a string property and limit the values in the form.



Create a property

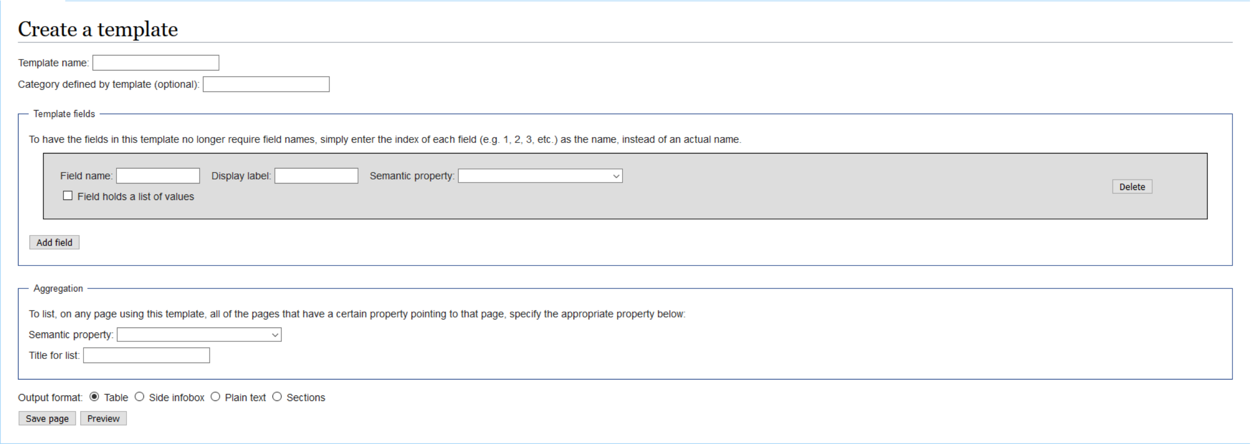
Property name: Type: Page

To only allow certain values, enter the list of values, separated by commas (if a value contains a comma, replace it with "\,"):

There are additional options for properties that are not available on the creation page, such as sub-properties. To set a property as a sub-property of another, the page of the property must be edited manually and [[Subproperty of::<superproperty>]] needs to be added.

When all properties for a class have been created we need a template for a page of this class. The template determines what a page of this class looks like and is required to create a form.

For every property that a class has we need to add a field in the template, set the field name, which is used internally for the semantics, a display label which will be seen on the page, and the property type. Here we can set the cardinality to an extent with the checkbox "Field holds a list of values". Unchecked means cardinality 0-1, checked means it can be anything (a minimum of 1 can be set via the form). We can also select some basic output formats for the created page, but these can be configured in more detail after template creation.



Create a template

Template name:

Category defined by template (optional):

Template fields

To have the fields in this template no longer require field names, simply enter the index of each field (e.g. 1, 2, 3, etc.) as the name, instead of an actual name.

Field name: Display label: Semantic property:

□ Field holds a list of values

Aggregation

To list, on any page using this template, all of the pages that have a certain property pointing to that page, specify the appropriate property below:

Semantic property:

Title for list:

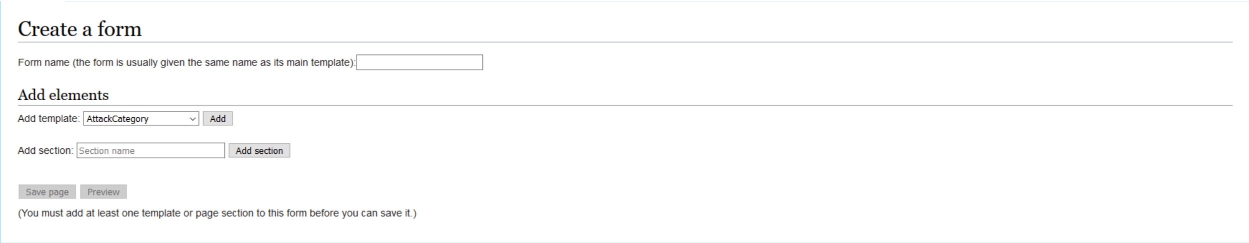
Output format: ● Table ○ Side infobox ○ Plain text ○ Sections

Similar to the property, after a template has been created we can do much more by editing the created template manually.

```
<noinclude>
This is the "AttackCategory" template.
It should be called in the following format:
<pre>
{{AttackCategory
|hasName=
|hasType=
|forIncident=
|relatedPlaybook=
}}
</pre>
Edit the page to see the template text.
</noinclude><includeonly>

'''hasName:''' [[Text::{{{hasName|}}}]]

'''hasType:''' [[Text::{{{hasType|}}}]]

'''forIncident:''' {{#arraymap:{{{forIncident|}}}|,|x|[[Incident::x]]}}

'''relatedPlaybook:''' {{#arraymap:{{{relatedPlaybook|}}}|,|x|[[Playbook::x]]}}

[[Category:AttackCategory]]
</includeonly>
```

Looking at the AttackCategory example, the part that changes with a different output format and that we would edit to change the look of a page of this class is what's inside of <includeonly>. More information about the page forms and template can be found here: https://www.mediawiki.org/wiki/Extension:Page_Forms/Page_Forms_and_templates

While forms allow for multiple templates, so far for our needs, we only need one template per each form, which then adds a field for every property in the form.



Create a form

Form name (the form is usually given the same name as its main template):

Add elements

Add template: AttackCategory

Add section: Section name

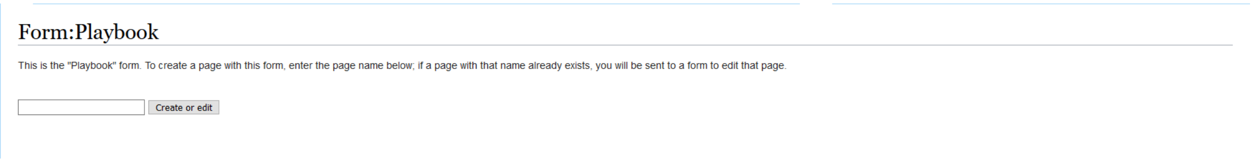
(You must add at least one template or page section to this form before you can save it.)

For most properties, no additional changes are necessary here. The aforementioned cardinality of at least 1 can be set with the parameter "mandatory". Again more options are only available when editing the form manually after creation. The primary one used so far that is not accessible from the form creation page is the input type. For the "hasType" property we allow strings but we only want limited ones, so a dropdown menu is ideal. Therefore we set the input type equal to a dropdown. The values are manually entered here and a default value can be set. Lastly, the mandatory parameter could be set for a dropdown menu, otherwise there will be an empty value that can be set. An overview of all input types can be found here: https://www.mediawiki.org/wiki/Extension:Page_Forms/Input_types.

```
! HasType:
|  {{{field|hasType|input  type=dropdown|values=Phishing,Domain  Generation  Algo-
rithm,Other|default=Phishing|mandatory}}}
```

4.6 Adding Playbooks

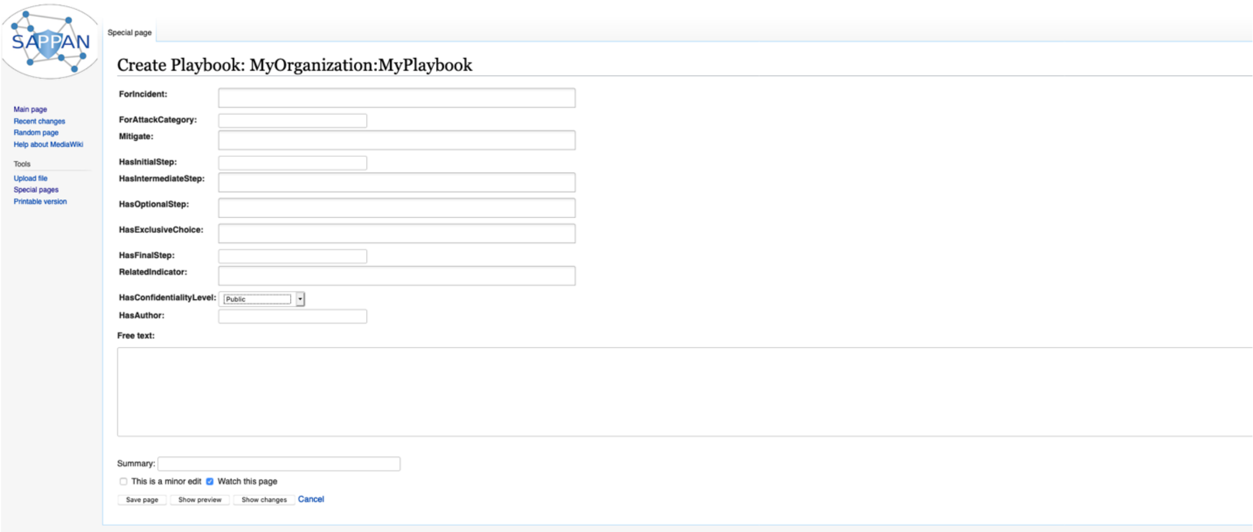
On the page "Special:Forms" forms for all classes used in the wiki can be found. To add a new playbook the "Playbook" form must be selected.



Form:Playbook

This is the "Playbook" form. To create a page with this form, enter the page name below; if a page with that name already exists, you will be sent to a form to edit that page.

Here the name for the playbook is added. If the playbook should only be visible to members of one's organization, the name must look like this: <organization-namespace>:<playbookname> and the visibility restriction should be configured by access restriction extension which is Lockdown extension in our case.



Create Playbook: MyOrganization:MyPlaybook

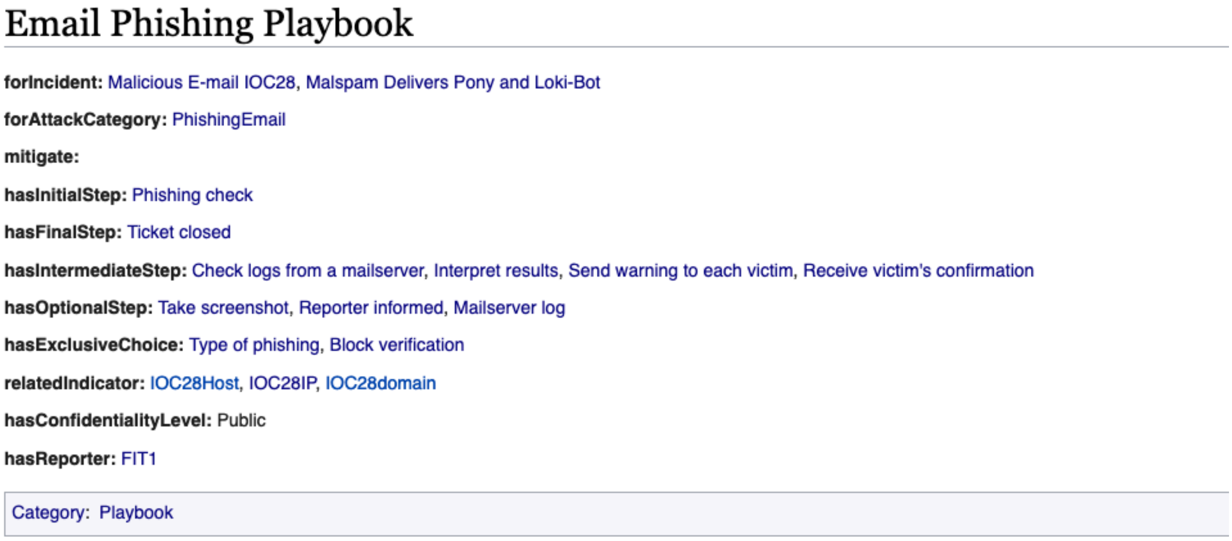
ForIncident:
ForAttackCategory:
Mitigate:
HasInitialStep:
HasIntermediateStep:
HasOptionalStep:
HasExclusiveChoice:
HasFinalStep:
RelatedIndicator:
HasConfidentialityLevel: Public
HasAuthor:
Free text:

Summary:

□ This is a minor edit ☑ Watch this page

In this form, all the properties for your playbook are filled in. All properties here expect a page in the wiki. Pages that don‘t exist yet but will be created later can be filled in. Those pages will be represented as red pages that can be selected and filled later by a corresponding form.

When the playbook itself is all set up the associated incidents, attack category, vulnerabilities, and steps can be created with their respective forms. The order of pages created does not matter, so any of these can be created before the playbook. The result (after adding all the steps) could look like this:



Email Phishing Playbook

forIncident: Malicious E-mail IOC28, Malspam Delivers Pony and Loki-Bot

forAttackCategory: PhishingEmail

mitigate:

hasInitialStep: Phishing check

hasFinalStep: Ticket closed

hasIntermediateStep: Check logs from a mailserver, Interpret results, Send warning to each victim, Receive victim's confirmation

hasOptionalStep: Take screenshot, Reporter informed, Mailserver log

hasExclusiveChoice: Type of phishing, Block verification

relatedIndicator: IOC28Host, IOC28IP, IOC28domain

hasConfidentialityLevel: Public

hasReporter: FIT1

Category: Playbook

4.7 Creating Extensions for Semantic MediaWiki

Extensions are additional functionalities that can be added to the existing functionalities of SMW. There are quite a lot of existing extensions, but it is also possible to create an own one. The following links are the starting points for creating a new extension.

A detailed guide for the development of an exemplary extension for Mediawiki and SMW that discusses the structure of extension, hooks (entry points for the extension), development (PHP-based), composer and class structure can be found here: https://www.semantic-mediawiki.org/wiki/SMWCon_Fall_2014/Mediawiki_Extension_Development

4.8 Querying with SMW Query Language

There are different methods to query SMW, for example by inline queries or the creation of concepts. More detailed information about the semantic search can be found here: https://www.semantic-mediawiki.org/wiki/Help:Semantic_search

SMW is basically providing its own query language for the content of the wiki, which is distinct to SPARQL queries. This query language can be used in the following ways:

- The special page "Special:Ask" provides a direct interface where queries can be created. It also provides partial auto-completion and suggestion of features and, therefore, is recommended when trying to get comfortable with the query language.
- The queries can be included in so-called inline queries in normal documents. This is useful to gather information from corresponding, linked documents that might be of interest. An example could be the page of a country which queries all cities in the country and provides a list of the 20 cities with most inhabitants.

- Concepts are special pages which can be created in the namespace "Concept". Concepts are composed of pages that are automatically filled with information. They can be used when information has to be bundled and can be used by other queries as normal wiki pages.

The general syntax for SMW queries consists of two separate parts. The first is the selection of pages that are of interest. These are identified by semantic links and corresponding values or resources. An example could be:

[[Located in::Germany]]

The second part of the query is the selection of information. It is done by using the "?" symbol and can be used to specify which attributes are selected, which ranges, etc. E.g., "?Population"

Of course, this is only a simple example of a query and much more complicated queries are possible. Inline queries can be specified with the following syntax:

```
{{#ask:

[[Located in::Germany]]

|?Population

}}
```

Similar, concepts are defined with the "concept" keyword instead of "ask".

Following, example queries can be found mapped to the first simple version of the vocabulary. Of course, these queries might have to be restructured if the vocabulary or structure is modified, and there might be other ways to express the same queries.

Playbooks connected to a vulnerability:

This is a simple query, which selects all instances of the category Playbook and prints all the vulnerabilities which are connected to these playbooks.

```
{{#ask:

[[Category:Playbook]]

|?Vulnerability

}}
```

All playbooks with their corresponding steps

This is a simple query, which selects all instances of the category Playbook and prints all the included steps.

```
{{#ask:

[[Category:Playbook]]
|?InitialStep

|?IntermediateStep

|?ExclusiveChoiceStep

|?OptionalStep

|?FinalStep

}}
```

Incidents connected to a playbook

This simple query lists all incidents connected to a specific playbook.

```
{{#ask:

[[$Playbook_name]]

|?Incident

}}
```

For example, using [[Email Phishing Playbook]] as the $Playbook_name will show all the incidents connected to the "Email Phishing Playbook".

Incident reports by a reporter

This concept selects all incident reports which are connected to the specific reporter (for example FIT1). This concept can be accessed via the URI "Concept:Reports_by_FIT1"

```
{{#concept:

[[Category:IncidentReport]]

[[Reporter::FIT1]]

|Reports_by_FIT1

}}
```

4.9 Export Result Formats

SMW provides the semantic search results in different formats. The default results are shown as tables, but it could be simply exported as JSON, CSV, RDF and many other formats by assigning the format type to the parameter "format". The export result can also be more specified easily by the corresponding parameters.

For example:

```
{{#ask:
```

```
"any query"

|format=rdf

}}
```

returns the output of the query as a default format of RDF which is RDF/XML.

```
{{#ask:

"any query"

|format=rdf

|syntax=turtle

}}
```

returns the output of the query as turtle format.

```
{{#ask:

"any query"

|format=json

}}
```

exports the output in JSON serialization format.

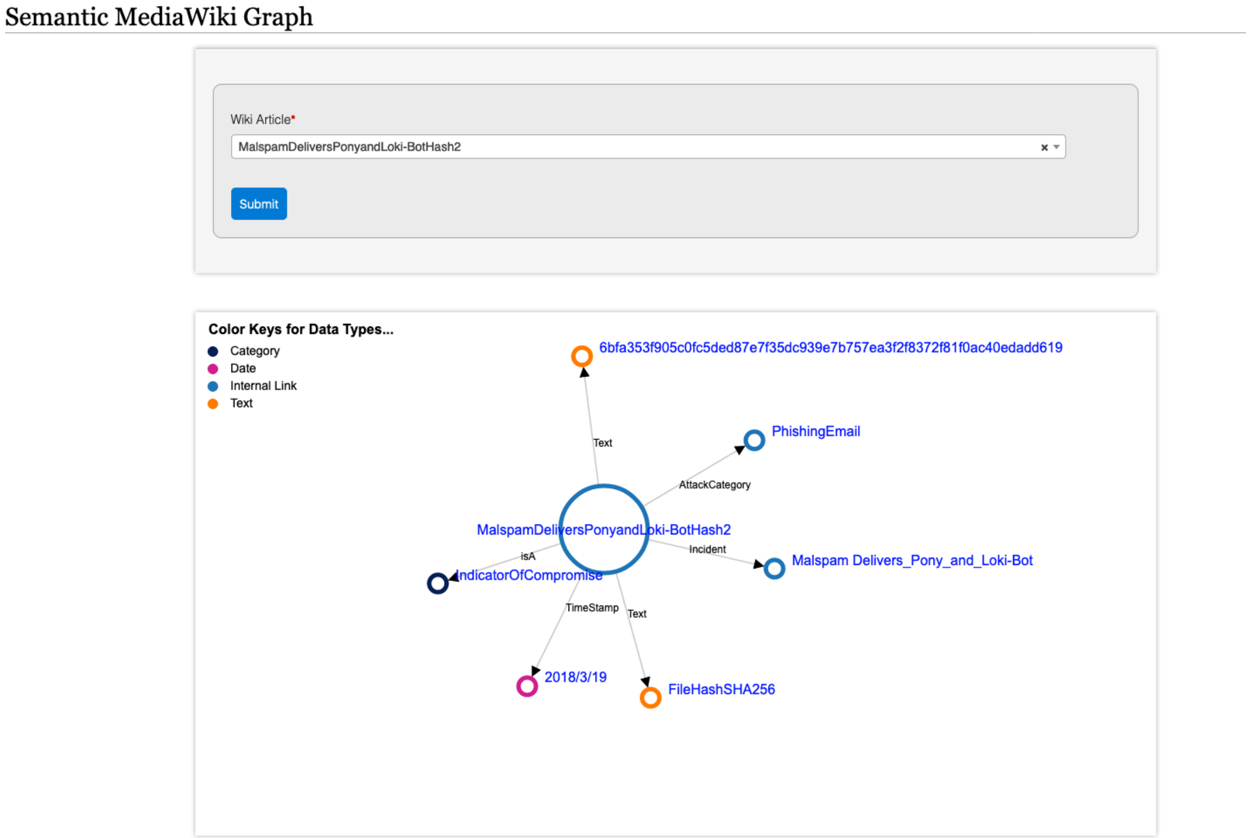
More information about export formats can be found here: https://www.semantic-mediawiki.org/wiki/Help:Export_related_result_formats

4.10 Graph Visualization

The knowledge graph which is representing the data stored in SMW can be visualized. One extension which offers such functionality is https://www.mediawiki.org/wiki/Extension:Semantic_MediaWiki_Graph.

This extension allows the selection of a node (a wiki page) and visualizes the surround connected other nodes. It can be accessed via the special page "Special:SemanticMediaWikiGraph".

This extension is included in the implementation. A graph of a sample Indicator of Compromise node is represented in the following; This is a SHA256 hash file for "Malspam Delivers Pony and Loki-Bot" transferred from the Malware Breakdown website [11].



Semantic MediaWiki Graph

Wiki Article*: MalspamDeliversPonyandLoki-BotHash2

Submit
Color Keys for Data Types...
● Category
● Date
● Internal Link
● Text
6bfa353f905c0fc5ded87e7f35dc939e7b757ea3f2f8372f81f0ac40edadd619
Text
PhishingEmail
AttackCategory
MalspamDeliversPonyandLoki-BotHash2
Incident
Malspam Delivers_Pony_and_Loki-Bot
isA
IndicatorOfCompromise
TimeStamp
Text
2018/3/19
FileHashSHA256

The graph visualization is not fulfilling the requirements for showing playbook workflows to increase the human-readability of the playbooks as it is desired by domain experts. Lack of showing the sequence of the steps is the main disadvantage of the current graph visualization which will be considered in the next phase of implementation.

For the next development phase, "GraphViz" and "Semantic Results Formats" extensions are considered being tested. Further information can be found here:

- https://www.semantic-mediawiki.org/wiki/Extension:GraphViz
- https://www.semantic-mediawiki.org/wiki/Extension:Semantic_Result_Formats

5 Evaluation and Feedback

5.1 Interview with the Domain Experts

For gathering information and feedback from domain experts, interview sessions have been scheduled. The interview questions consist of two main topics. First, information about their current approaches, and second, feedback on our proposed approach based on SMW. The information about current approaches, advantages, disadvantages, and their expectations has been gained from different organizations during the project, but it was included in the interviews to conclude all the aspects consistently for different organizations. The answers for the first part consist of much sensitive and confidential information, but the lesson learned has been and will be considered in the approach development.

General information regarding the first part of the interview is discussed in the "Handling cybersecurity incidents by SAPPAN members" section.

Each interview starts with a brief introduction and describing the main goals. The interviews were held with the incident analysts (cyber defense team) of CESNET, Dreamlab, and HPE. Information about organizations' response and recovery actions also gained from the CSIRT team of Masaryk University and domain experts from F-secure. Moreover, feedback regarding our approach is collected from a SAPPAN dashboard developer affiliated with the University of Stuttgart. Interview points are listed in the following table.

Main topic	Interview points
Introduction to organization's response and recovery tools and approaches	Current reporting tool • Preparation phase • Detection and assessment phase • Mitigation phase • Post-incident activity phase
	Sharing approach
	Privacy issues (confidentiality levels)
	Data Source / APIs / IO formats / Database
	Response and recovery actions • Limitations and pain points • Availability • Adding Playbooks • Editing Playbooks • Querying • Sharing Playbooks • Results for automation of response and recovery process
	Alternative approaches • Advantages and disadvantages
	General Limitations, bottlenecks, obstacles and pain points
	Scalability
	Learning curve • Security knowledge • Computer network knowledge and experience • Programming proficiency • Statistical knowledge

	Roles/ hierarchy • Access management
	General Evaluation • Machine-readability • Human-readability • Unambiguous semantics • Interoperability • Extensibility • Aggregability • Practical application • External dependencies
A brief introduction to our approach on Semantic MediaWiki, showing the prototype implementation (inserting, editing, and querying data) and collecting feedback on our approach	General Impression • Machine-readability • Human-readability • Interoperability • Practical application • Limitations
	Vocabulary • Applicability for incident reporting • Applicability for capturing response and recovery actions • Ambiguousness of semantics • Level of abstraction • Suggestions to improve • Detection and assessment phase • Mitigation phase • Outputs for recommendation and automation of response and recovery process
	User experience
	Roles

5.2 Lesson Learned and Feedback on Capturing Tool

The main feedback from the interviews regarding our approach can be listed as follow:

	CESNET	DL	USTUTT	HPE
General Impression • Machine-readability • Human-readability • Interoperability • Practical application • Limitations	• Going to another page to create indicators for incident cumbersome • Incidents usually created via email and not manually (parser) • Needs proper ticket system GUI (see https://thehive-project.org/, has API to be used by other systems): It should be closely integrated with a ticket system, so it guides the handler through the ticket's lifecycle from its creation to close. • Machine-readability is okay • Human-readability: how information is displayed on pages needs improvement, also querying is not human-readable	• The separate knowledge base from the incident response for SOCs • Information collected not integrated with the incident itself, the incident must be escalated to SOC level • It is not mentioned how the SMW's structure will integrate with case-management and incident management workflow tools • A federated solution, similar to the MISP feed model, where every organization has its own instance and decides what to share can be considered as a model for this solution	• Filling out forms must not be time-consuming, the process must be streamlined (to prevent reluctance of potential data providers) • Presentation lacking for human-readability	• Good first impression, About the structure of playbooks better than non-structured text only playbooks • Machine-readable, more structured way to describe playbooks → better than text-only, non-structured, or semi-structured approaches

Vocabulary • Applicability for incident reporting • Applicability for capturing response and recovery actions • Ambiguousness of semantics • Level of abstraction • Suggestions to improve • Detection and assessment phase • Mitigation phase • Outputs for recommendation and automation of response and recovery process	• Vocabulary looks good in general • Add loops in playbook steps (i.e., re sent mail after some time without response, the interval to check blocks) • Timer in steps for repetition • Steps could belong to multiple playbooks • Can be implemented by branching or using some steps in different playbooks • For IP blocking: address, the decider for a block (blocking on organization level unless severe case) • Show all gathered information in one spot for easier decision making, automation for specific cases (i.e., DDoS attack above the intensity threshold)	• Various cardinalities are debatable and may need amendment. For example, the AdversaryActor has a "hasMotivation" property with exactly one value. In real cases, it might be unknown or with multiple values. • How do risks or possible collateral effects of remedies become part of playbooks steps? • Suggestion: using MITRE ATT&CK framework as vocabulary	• Looks reasonable • Improve visibility of important properties • Classify playbook with tags to connect to organization-specific information, e.g. for triggering alerts	• The link between steps is good, is better to see playbooks as a workflow (Graph visualization) • Suggestions: • Vocabulary for assessment phase → risk → 2 factors: likelihood and impact • Intermediate steps in playbooks are too general, the idea to add different categories: detection--> containment--> remediation--> recovery (it depend, maybe decide to merge containment and remediation, then have all phases in sequence and not mixed up)

User experience	• Query types: by category, age, text in the description, reporter • event: source address, target address, time range • more so required for the researcher rather than the incident handler	• Share playbooks as workflows, (fully) or semi-automated, some flexibility in steps depending on customer needs • It is not mentioned how the SMW would show the event-specific context of a certain workflow instance.	• Consider the visual design of playbook to highlight more important properties • graph of all steps on one page • Queries: alert highly organization-specific (e.g., for an organization: outlook starts command prompts) • flexibility with custom queries • combine exact and fuzzy query (e.g., playbook wherein initial step outlook.exe occurs)	• Query: if the search component works well it is good, about different ways of presenting words (Caps, spaces, and ...)
Suggestions for Roles	• Admins and Handlers • Handlers only access to their own events	-	• Organization-specific view not required in SAPPAN since confidential material won't end up in the wiki	• Read-only for everybody except team lead • Team lead with the read/write access • Changes are expected to be done in a team and can be applied by a team lead

6 Conclusion

Based on the interviews and feedback that we collect from domain experts, we will focus on the ontology development based on the current approach. Additional review round and collect more feedback during the ontology development process will be required. Based on the next actions and feedback we may reassess using this prototyping approach and switch user interface, extensions, and frameworks to fulfill the future requirements and desire of domain experts. The final version of the incident response and recovery tool will be provided with the final version of the vocabulary in the deliverable D4.2. Further, based on the domain experts' feedback, the graph visualization can increase the human-readability of the approach. Therefore, a proper graph visualization of the playbook workflow will be proposed in the future version.

The focus of this deliverable was not on developing the vocabulary that fit all the criteria. Preliminary vocabulary has been developed to test the response and recovery approach, feasibility of its expression, and suitability of the environment. We also get feedback for the development of the final vocabulary. The current vocabulary is not entirely fit all the requirements, e.g., expression of playbook steps, their categories, connections, and means require modification, and response actions and risk assessment vocabulary are still missing.

In addition, the confidentiality level of entries and access management of different roles can be specified and defined in more detail. As the issue is mentioned in the privacy concern section, when the confidentiality levels are applied to steps, it might produce a complicated issue when we cannot share a sensitive step of a playbook. A simple solution that was discussed is replacing such a step by an "empty step" that does not reveal any information of the confidential step, but that insure the structure of the playbook remains unchanged and the receivers can understand that is was some sensitive step that could be defined by their own specifications. Although, more complex solutions can be offered for specific conditions and there is still an open discussion regarding this issue.

7 References

1. Marc Burian. Machine Learning based Handling of Cyber Security Incidents, Master Thesis, RWTH Aachen, 2019
2. Kick, J. 2014. “Cyber Exercise Playbook,” MP140714, The MITRE Corporation, November, 2014. Available at https://www.mitre.org/sites/default/files/publications/pr_14-3929-cyber-exercise-playbook.pdf
3. Unit 42 playbook, project website: https://pan-unit42.github.io/playbook_viewer/
4. STIX (official introduction), available at https://oasis-open.github.io/cti-documentation/stix/intro
5. Incident response playbooks, available at https://www.incidentresponse.com/playbooks/
6. MISP Taxonomies (overview and taxonomy repository), available at https://github.com/MISP/misp-taxonomies
7. MISP Galaxy, available at https://circl.lu/doc/misp/galaxy
8. VERIS project, available at https://github.com/vz-risk/veris
9. Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, UCO: a unified cybersecurity ontology, The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence Artificial Intelligence for Cyber Security: Technical Report WS-16-03. AAAI, 2016.
10. OTX User Guide, available at https://cybersecurity.att.com/documentation/otx.htm?tocpath=Documentation%7CAlien-Vault%C2%AE%20Open%C2%A0Threat%20Exchange%C2%AE%7C_____0
11. Malware breakdown, IOCs and Malware Samples, available at https://malwarebreakdown.com/
12. Semantic MediaWiki, documentations and user guide are available at https://www.semantic-mediawiki.org/wiki/Semantic_MediaWiki

13. Semantic MediaWiki, Using SPARQL and RDF stores, available at https://www.semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores
14. ElasticStore on Semantic MediaWiki, documentation available at: https://www.semantic-mediawiki.org/wiki/Help:ElasticStore
15. Semantic MediaWiki, Page Forms extension, documentation available at: https://www.semantic-mediawiki.org/wiki/Extension:Page_Forms