CRUSOE: Data Model for Cyber Situational Awareness

Jana Komárková
Institute of Computer Science
Faculty of Informatics
Masaryk University
Brno, Czech Republic
komarkova@ics.muni.cz

Martin Husák
Institute of Computer Science
Faculty of Informatics
Masaryk University
Brno, Czech Republic
husakm@ics.muni.cz

Martin Laštovička
Institute of Computer Science
Faculty of Informatics
Masaryk University
Brno, Czech Republic
lastovicka@ics.muni.cz

Daniel Tovarňák
Institute of Computer Science
Masaryk University
Brno, Czech Republic
tovarnak@ics.muni.cz

ABSTRACT

Attaining and keeping cyber situational awareness is crucial for the proper incident response, especially in critical infrastructures. Incident handlers need to process heterogeneous data, such as network topology and organisation's missions and objectives, to effectively mitigate the threats. The development of tools for attaining cyber situational awareness often faces the problem of effectively obtaining, correlating, and storing such heterogeneous data. In this paper, we present CRUSOE, an extensible layered data model for attaining and keeping information on cyber situational awareness. We conducted interviews with incident handlers from several security teams and evaluated existing requirements on cyber situational awareness to formalise the requirements on the proposed data model so that can be used in today's common network settings. The CRUSOE data model keeps track of missions, systems, networks, hosts, threats, detection and response capabilities, and access control in a network of an organisation. It is also designed to be filled primarily with the data that can be obtained in a semi- or fully-automated fashion in today's common network environments.

CCS CONCEPTS

• **Security and privacy → Formal security models**; • **Networks → Network security**;

KEYWORDS

situational awareness, data model, attack graph, impact assessment, vulnerability management

ARES 2018, August 27–30, 2018, Hamburg, Germany

ACM ISBN 978-1-4503-6448-5/18/08…$15.00
https://doi.org/10.1145/3230833.3232798

ACM Reference Format:
Jana Komárková, Martin Husák, Martin Laštovička, and Daniel Tovarňák. 2018. CRUSOE: Data Model for Cyber Situational Awareness. In *ARES 2018: International Conference on Availability, Reliability and Security, August 27–30, 2018, Hamburg, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3230833.3232798

1 INTRODUCTION

The complexity and changeability of cyber environment severely hinder the understanding of the current situation by security teams. Especially in critical information infrastructures, it is essential that the security team be aware of the location, function, and dependencies of critical systems. Should they fail to do so, they might not be able to recognise attacks against the components of the system or they might select mitigation that unknowingly severs some of the dependencies, thus harming the mission operation.

More and more the problem of attaining situational awareness by the security teams is studied by researchers. The definition of situational awareness (SA) by Endsley [11] is *"Perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future."* When applied to the cybersecurity field, it is often referred to as cyber situation awareness (CSA) [20, 24]. The specific characteristics of cyber situation awareness as described by Onwubiko [34] are the dynamism and complexity, automation, real-time processing, multi-source data fusion, heterogeneity, security visualisation, risk assessment, resolution, and forecasting and prediction. The current challenges in cyber situational awareness as described by Kott [24] include the complex and fluid system topology, rapidly changing technologies, high noise to signal ratio, time bombs and lurking attacks, rapidly evolving and multi-faceted threats, speed of events, non-integrated tools, data overload and meaning underload, and automation induced SA losses.

Systems for visualisation and analysis based on a central data model could help to handle many of this specific characteristics and challenges. The filling of data model could be automated, the data based on monitoring of the network keeping the model up to date. The formalisation of the data is needed for automated analysis that could help to battle the complexity of the cyber environment. The data model could be filled from many sources, thus be the

point of integration of various otherwise non-integrated tools. The centralised location of data could be used by other tools to visualise the situation efficiently.

Several models have already been proposed for this purpose [16, 17, 33]. However, the models are either too specific or outdated, and none of them is sufficient for describing the information required by security teams. In this paper, we present the CRUSOE[1] data model for cyber situational awareness. The model contains a formalised description of information relevant to cyber defence and can be used as an information base for visualisation and analysis.

The paper contributions to state of the art could be summarised as follows:

- We summarise the requirements on a data model that could be used for capturing cyber situational awareness.
- We propose a data model that fulfils the requirements and describe in details its entities and relationships.
- We describe the data sources that can be utilised to fill the model in fully automated or semi-automated fashion.
- We illustrate how does the proposed data model enhance incident response in common scenarios.

This paper is divided into five sections. Section 2 surveys the data models used for attaining cyber situational awareness. Section 3 summarizes requirements, use cases, and characteristics related to the proposed data model that is described in details in Section 4. Section 5 concludes the paper and outlines further research directions.

2 RELATED WORK

In this section, we survey the existing data models for attaining and keeping cyber situational awareness. The existing approaches are briefly introduced and discussed in terms of their usability and technological readiness in today's networks and fulfilling the requirements on our data model.

M2D2 data model [30] was created for the purpose of alert correlation. It holds four information types and their relations. The included information types are characteristics of the monitored information system, vulnerabilities, security tools used for monitoring, and information about detected events. The information about the monitoring system includes the network topology, hosts, and their configuration (operating systems, programs and their versions running on the host). The model describes vulnerabilities and a which vulnerability affects which configuration, which, in turn, allows identifying all vulnerable hosts in the network. The model accounts for two types of security tools: intrusion detection systems and vulnerability scanners. They generate alerts and scans respectively. Alerts and scans together with events that are the cause of alert constitute the observed events. An event can refer to particular vulnerability or source and destination IP addresses. The model is missing any relations to requirements, function, and mission and is unnecessarily detailed concerning networking part (the model describes all interfaces) and observed events part (there are four different types of events).

[1]The name is derived from the abbreviated name of the project "Research of Tools for Cyber Situational Awareness and Decision Support of CSIRT Teams in Protection of Critical Infrastructure" (CRUSOE for short)

Layered design of a data model was presented in work by Innerhofer-Oberperfler and Breu on using enterprise architectures for IT risk management in [18]. A model consisting of business, application, technical, and physical layer was proposed together with several views on the data. The model is primarily used for threat management, with focus on threat lists and risk matrices for the threats. In our work, however, we aim at creating a data model applicable in incident response rather than in threat management.

Virtual Terrain (VT) [17] is a model created to represent the network situation. The information contained within the virtual terrain includes hosts, servers, running services, known vulnerabilities, intrusion sensor detection locations, firewall rules, and physical connections. It provides a fairly accurate description of the network, however, being designed in 2008 and not maintained since, it does not provide a capability to describe newer technologies, such as virtualisation, clustered computation. Further, it lacks the description of mission and dependencies between components and relation between mission and components.

The Cyber Assets, Missions and Users (CAMUS) [16] is a proof of concept system that allows automatic mapping of cyber assets to the missions and users. The core of the model is relatively simple and has four entities: user, mission, cyber asset, and cyber capability. It is extended by detailed information about each part, such as the role of the user in the organisation or a workstation the user often uses. The detailed structure of the model was described by Buchanan [5], and further developments are presented at Mission Impact Workshop [9]. The workshop addressed how to map the relationships between cyber assets such as network devices and the users, missions, business processes and other entities that depend on those assets. The focus is on populating a model in an automated fashion based on network traffic and logs from servers, distinguishing it from any previous models. The model provides an excellent description of the mission, user, and asset part, however, lacks any information about threat landscape, detection tools, and network topology.

Jakobson [23] focused on mission modelling with mission impact assessment in mind. He used a model of cyber terrain [22] and focused on modelling the mission elements and dependencies between them. He models the mission as a sequential or parallel flow of mission steps. Each step is another flow, another mission, or a single task. The model allows for stating the dependencies of mission steps, the allowed dependency types are AND and OR. The impact dependency graph (IDG) maps the mission elements to services and assets that are part of the cyber terrain. The mission steps express the temporal aspect of the mission. At each point in time, the steps are either in progress, completed or waiting for the start, and the impact of the attack is dependent on the state of each mission step.

Structured Threat Information eXpression (STIX) [3] is an effort by Mitre to define and develop a language for the specification, capture, characterisation, and communication of standardised cyber threat information. The first version of the model was introduced in 2012, and it has been evolving ever since, in 2017 version 2.0 was released. The data model is very detailed, focusing on the description of threats, attacks, actors, indicators, defences, security events and incidents. While it was created for a different purpose, it certainly has an overlap with situational awareness domain.

CyGraph [33] is a system for improving network security posture, maintaining situational awareness in the face of cyber attack and protecting mission-critical assets. The central data model used by the system consists of four layers: mission readiness, cyber threats, network infrastructure, and cyber posture. The mission readiness layer contains the relations between mission objectives, tasks, and information and their dependence on cyber assets. The cyber threats layer describes the exploit and related alerts that are raised by detection systems. The network topology, sensors, and firewall location and configuration are captured in network infrastructure layer. Finally, the cyber posture layer contains the information about vulnerabilities, their properties, the presence of a vulnerability on host and exploits against vulnerabilities.

CyGraph utilises a graph database to store the entities and their relations and introduces its domain-specific query language, CyQL, for queries against CyGraph data model. The database is populated from various sources. TVA/Cauldron tool provides the information about topology, firewall configuration, and vulnerabilities on hosts. They also include data from various sensors and detection tools, vulnerability data from NVD, STIX and attack patterns from Common Attack Pattern Enumeration and Classification (CAPEC). The primary purpose of CyGraph is security posture analysis and visualisation. However, it can be used to build a predictive model of possible attack paths, identify critical vulnerabilities, correlate alerts to attack paths and allow for analysis in the context of mission assurance. Our model is strongly inspired by CyGraph that also uses the layered design. However, in CRUSOE data model, the layers correspond to the areas of expertise, thus making the data model comprehensible by the subject matter experts which in turn enables them to truly describe the actual state in the language of the model. Further, CyGraph description adheres to other formal descriptions in Mitre's Making Security Measurable and assumes the data sources also adhere to these standards. However, this is more often than not the case. CRUSOE data model is more flexible and emphasises filling the model with data. The CRUSOE data model consists only of the data that can be acquired in a typical network environment.

3 REQUIREMENTS AND USE CASES

In this section, we summarise the requirements, use cases, and high-level characteristics related to the data model. First, we follow the use cases for a cyber situational awareness system listed by NATO [31] and select the ones that are related to data modelling. Second, we list the use cases discussed during the interviews with incident handlers from several European security teams. Finally, we sum up key characteristics of the data model that is going to be presented in details in Section 4.

3.1 Selected NATO Use Cases

NATO Cyber Defense Situational Awareness Request for Information (NATO CDSA RFI) [31] states 35 use cases for a cyber defence situational awareness system. These use cases summarise the functionality required by the nations in the area of cyber situational awareness. Following we list the most relevant use cases that would be solved by implementing a cyber situational awareness system based on our proposed data model. The use cases are ordered based on their relevance.

Single authoritative data source (UC10) – Users can consider the system as a single authoritative data source, due to the confidence generated from high quality, complete data, sourced broadly and with conflicts resolved. Other systems and stakeholders can use the authoritative data.

View connections of asset (UC12) – View the dependencies associated with an asset, including such associations as host to component to mission, component to component, and mission to mission dependencies.

Fuse data (UC15) – Data is fused to enrich the data available to the user – where partial host information is captured from various sources, data fusion rules allow these to be associated with one another as information about the same host. Conflicting data can be resolved either by predefined rules (one source is authoritative over another) or presenting them to the user to resolve where rules do not exist.

Drill down / Roll up (UC03) – The user can access additional details on corresponding data elements (drill down). The user can also access higher-level information that relates to detailed information (roll-up).

View asset dependencies (UC06) – Assets are organised within a dependency hierarchy in which missions depend on components, and components depend on hosts. By viewing asset dependencies, users can identify what an asset depends on, and what assets depend on an asset.

View interconnectivity (UC11) – View the logical and physical connections between a host and other entities on the network. Users can view the network as multiple subnets or zones, separated by border devices. Users can see where hosts reside on the network, and which zones are adjacent to each other.

As illustrated in the use cases above, the existence and structure of the proposed data model (assuming it is filled with data) would significantly contribute towards the desired state of the cyber situational awareness information system.

3.2 Interviews with Incident Handlers

In addition to addressing the NATO use cases, we have interviewed incident handlers from several CSIRTs (Computer Security Incident Response Team) from two European countries to gain insights on what they lack in day-to-day operations and security incident response. Following the interviews outcomes, we formulated use cases according to ENISA incident handling workflow [29] that are fulfilled by our data model:

Criticality estimation of attack target – During an incident handling triage (initial assessment of the incident), an incident handler needs to assess the targets' significance for its constituency to assign priority of the incident or to take an immediate action [29]. The incident handler can perform this task with a prepared query to our data model. Every security alert is stored in the model and connected to corresponding nodes of the model figuring in the event. The handler can find detailed information about the target, find all system components dependent on it, and the calculated criticality of the target. Furthermore, the mission requirements on

the target are specified in the model. All these information is crucial to estimate the potential impact of an attack.

Finding responsible person – After the triage, incident handling cycle continues with finding contact information and response coordination. Contacting the right persons within organisation significantly speeds up the incident resolution. In our data model, the security event is connected to IP address and its network subnet, which is further connected to an organisation's unit or a specific user. Choosing the affected machines' responsible administrator contact stored in data model will ensure that the incident will be solved fast and by the most suitable person.

Vulnerability prioritisation and dissemination – Due to a large number of vulnerabilities disclosed every day, there is a need to assess the risks quickly during the vulnerability handling process. However, commonly used methodologies, such as CVSS[2] and TARANIS [8], do not take the situation in the target organisation into account. In our data model, each vulnerability is mapped to a specific host, which helps to discard vulnerabilities not threatening the organisation. Among those, it is easy to find the vulnerable host dependencies, criticality, and requirements, to assess potential impacts. This procedure will help to automatically identify the most important vulnerabilities and consequently alert the responsible administrators.

On the three examples of common problems in incident response, we illustrate the benefits of the proposed data model and, respectively the assumed tools that implement it. Moreover, the interviews with incident handlers provided valuable feedback to our work.

3.3 Data Model Characteristics

The data model is a central point of information required for cyber situational awareness. The role of the model is threefold; it provides the backbone for all relevant information, it specifies the data input format for further analysis, and, last but not least, it defines the desired target for information gathering.

The role as a target for information gathering helps to bridge the gap between incompatible tools. The heterogeneous data format and incompatible tools were identified as one of the leading problems hindering cyber situational awareness [24]. Once the data model is considered as the receiver of the data, it can serve as a point of integration. The tools might be designed to provide data in an appropriate format, which is highly improbable ever to happen. Alternatively, in a more feasible scenario, preprocessor modules might be developed to transform data from each tool to data model format.

There might be other valuable data for analysis and visualisation besides the entities and relations from the model. Example of such data are logs from services and hosts, incident history, or network flow records. The data model should not stand side by side with this data, creating again a situation where many data sources provide partial information. Instead, the additional information should be related to specific entity or relation in the model, so that the model is a backbone and acts as a context for such additional information. The advantage of such approach is that when the operator requires detailed information to some pattern observed in the high-level model, the information is readily obtainable since it is already related to the entities and relations that are part of the pattern.

The data model also serves as a well defined and up-to-date data input for further analysis and visualisation. Together with appropriate visualisation methods, the information can be directly used to increase cyber situational awareness.

Many concepts for intrusion response [14, 35], alert correlation [27, 38], and attack graphs [21, 36] work well in theory, but need a lot of input data. Authors do not often cover the problem of obtaining the required data and as such the methods and algorithms cannot be deployed in production. The existence of a data model that can be reasonably filled with data would help towards the application of such methods.

The characteristics that are required in the data model to fulfil the role of the data model as described above can be summarised as follows:

- **All-embracing** – The model can capture every technology and concept in use nowadays, including concepts such as clustering, virtualisation, and VPN.
- **Comprehensive** – The model captures all aspects of cyber security. Many roles are relevant to cyber security, such as security team members, system administrators, network administrators, security managers, and users, each of them having their perspective. The model has to be capable of modelling all the views and joining these views together.
- **Attainable** – The model should capture the essential information relevant to cyber security. However, it should not be too detailed or contain parameters that are unobtainable.
- **Sustainable** – The model should not have parameters that are difficult to maintain, such as manually inserted parameters that change frequently.
- **Time-conscious** – The model should be able to capture and handle the variability of information in time.
- **Extensible** – The model should allow for its extensions and provide a capability to tie more detailed information to it, that presents a high-level overview. The model should be a corpus, not "yet another model" that stands side by side with other models.

From these requirements, we consider the attainability and sustainability as the most important. Without these two characteristics, the model could be otherwise perfect but useless none the less, since in the typical settings it would never be completely filled with data and up to date. Less emphasis could be put on comprehensiveness and all-embracing aspect of the model. Given it is extensible, these two characteristics can be achieved by gradual enhancements of the model.

4 DATA MODEL PROPOSAL

In this section, we describe the proposed data model, its entities and relations. We define the entities and give reasoning for specific relations and properties. We also outline data sources that can be used to fill the information into the model.

The proposed model consists of seven layers as depicted on Figure 1. The layers are composed in such a way that each of them corresponds to the area of expertise of various roles in a company, such as network administrators, application administrators, and

[2] https://www.first.org/cvss/

security teams. Each layer is described in more details later in this section.

The model is capable of describing a state at given point in time. The model entities and relations are marked as either *fixed* or *timed*. The *fixed* entities and relations do not change often, and there is no value in old entries, such as mission definition and requirements. For the security team, only the current mission and requirements are of importance. On the contrary, the active security incidents change fairly often, and the historical data are used, e.g., for estimating trustworthiness. The *timed* entities and relations are marked by a start and end attribute (or first/last observed attribute).

The model should act as a backbone for additional information. Each entity and relation could point to additional information or historical information of a given type. Example of such may be the relation of a user having access to an application; the additional information is the access logs from the machine coming from that user. We designed the model with such information in mind so that each log, record, and event might be placed as additional info to the model.

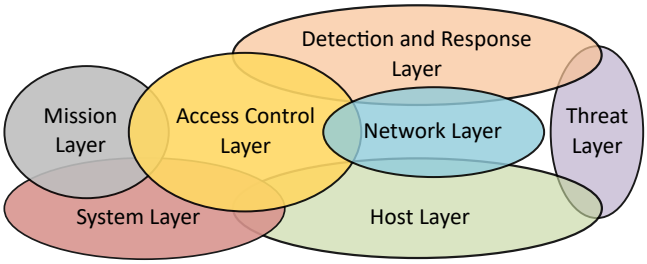

Figure 1: Layers of the data model

The rest of the section is dedicated to a detailed description of each layer. We describe the purpose of each layer, its entities and, if needed, the reasons why are some entities included in the model. Then we describe the relations between the entities in the layer, **intra-layer** relations, and relations between entities from the layer and entities from other layers, **cross-layer relations**, that outline the intersections between various areas of expertise. The data model and a sample database based on it are available at Github[3].

4.1 Host Layer

The host layer, illustrated in Figure 2, describes the host setup and configuration. It should describe a system administrator's point of view. The layer allows for tracking network services and applications on the host, enabling rough estimation of the host role (server, client) and tracking the changes on the host which might be indicators of compromise.

A **host** is an entity describing a general source of computation power. It has two subclasses: **physical host** and **virtual host**. Virtual hosts are hosted on physical hosts, which can be important information in case there is a vulnerability in the hypervisor. A **host cluster** denotes a set of hosts that pool their computational power to host software resources, which is increasingly frequent

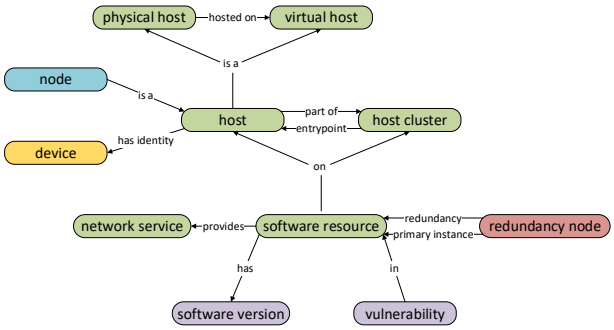

Figure 2: Host Layer

scenario due to the emphasis on scalability and availability. A software resource might be shifted among several hosts. Therefore it is difficult to pinpoint the actual host running it. Host cluster is usually accessed through its entry point that forwards the request to correct cluster host. The typical example is Apache Hadoop cluster[4] with its single master node and multiple slave nodes. A **software resource** is a general software (including OS) running on a host. A **network service** denotes an endpoint capable of accepting connections from the network and providing communication with underlining software resource.

There are five distinct relations between entities in the host layer. A software resource **runs on** a host or a host cluster. A software resource might also **provide** a network service and be capable of communication over the network. A virtual host is **hosted on** a physical host. As mentioned above host can be **part of** host cluster, sharing resources to host software resources. Each host cluster has one **entrypoint**, which is a host serving as a gateway for communication with software resources running on the host cluster.

Beside the intra-layer relationships, entities from host layer have relations to entities from other layers. A software resource may provide **redundancy** or be a **primary instance** of a component, which belongs to the system layer. Each software resource **has version**, the software version entity belongs to the threat layer as well as a vulnerability, which may be **in** an instance of the software resource, i.e. due to the wrong configuration. The relation of software resource to its version helps to immediately find all hosts that are affected by a vulnerability in an application. The host itself **is a** node in a network, an entity from the network layer. A host may also be capable of access control; thus it **has identity**, which defines roles and users that have access to the device in those roles.

Passive network monitoring and active network scanning, such as Nmap[5] or Nessus[6] could partially fill the Host layer by data. The active scanning can give information about network services and their versions. The passive network monitoring can discover network services, client applications communicating through observation point, and their versions using fingerprinting, although the

[3]https://github.com/CSIRT-MU/CRUSOE-Data-Model

[4]http://hadoop.apache.org

[5]https://nmap.org/

[6]https://www.tenable.com/products/nessus-vulnerability-scanner

version determination would be less reliable. The passive network monitoring can also be used for OS fingerprinting [26]. The most reliable information about the network services, application, and OS would give log analysis or host-based agent. However, those data sources might not be available.

The information about the host clusters is available in virtualised environments. However, in most cases, host cluster is not visible to network monitoring or active scans, and thus, the system administrator must insert its presence manually.

4.2 System Layer

The purpose of the system layer is to describe components of information systems and their dependencies as illustrated in Figure 3. This layer also provides information about the data, its location in the system, and about the workflows in the system. This layer should represent the point of view of a person responsible for the information system deployment. The purpose of the layer is to track the dependencies among components, thus enabling the analysis of propagation of impact. The mapping of data to components handling those data is especially useful for sensitive data protection.

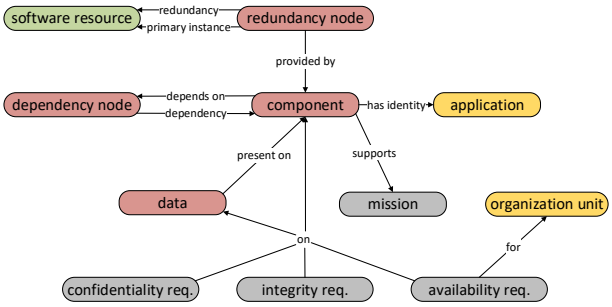

Figure 3: System Layer

A **component** is single part of an information system. It usually corresponds to a single application. Especially in service-oriented architecture (SOA), each service is a component. The components might depend on other components for their functionality. A **dependency node** captures such dependency and its type. We consider three types of dependency: *and*, *or*, and *n-out-of-m*. Components might be realised by one or more instances of software resources, e.g., primary instance and backup instance that can function in case the primary instance is unavailable. **Redundancy node** describes such setup. The redundancy is always *n-out-of-m*, and there are several types, such as round-robin load balancing between several instances, one primary instance and one or more backup instances that receive requests only when the primary instance is unavailable. The important aspect, especially with the latest focus on the privacy of personal information, are the **data** stored in the system. Data can be handled by components and can be an object of requirement on confidentiality/integrity. The requirement on data implies a requirement on the component that handles those data.

The dependencies are expressed by two relations, a component **depends on** a dependency node and this node has **dependent** components. Furthermore, the components are **provided by** redundancy nodes as described above. Data in the system can be stored at different locations but must be **present on** and be handled by at least one component.

The relation between redundancy node and software resource from host layer is the first relation to other layers. The redundancy node can be a **primary instance** of the resource or be one of the **redundancy** nodes. A component can **have identity** of an application, having all of its trust relationships and granting access to all identities having access to the application identity. A component is also required by the mission and **supports** its operation. The mission layer also poses **requirements on** components and data as each part of the system can have different needs regarding confidentiality, integrity, and availability.

The only reliable source of data is a manual insertion by the system administrator. In case of critical information infrastructure, many countries make maintaining such information mandatory.

Automated dependency detection methods could help fill the data. However, they have many drawbacks. The automated dependency detection is either active or passive. The active methods [2, 7, 13, 37] are usually relatively accurate, but require modifications of all monitored systems or communication protocols which can prove problematic in live systems since it can affect the system performance. The passive systems [1, 6, 32] were introduced to work in an environment where modification of existing software or traffic injection is unacceptable. They are usually based on packet sniffing and traffic monitoring and infer the dependency by analysing packets arrival time distribution or from flow correlation and statistical analysis of network traffic.

The automatic detection has the advantage that it also detects the dependencies that tend to be forgotten, such as dependencies on DNS service for domain name resolution.

4.3 Network Layer

The main purpose of the network layer illustrated in Figure 4, is to describe the topology of the network as well as routing and filtering information. The model is limited to the Network layer of OSI model. The Network layer should allow answering questions about the relative position of two hosts, whether given communication passes through observation points and which nodes need to be reconfigured in order to enable/disable given communication.

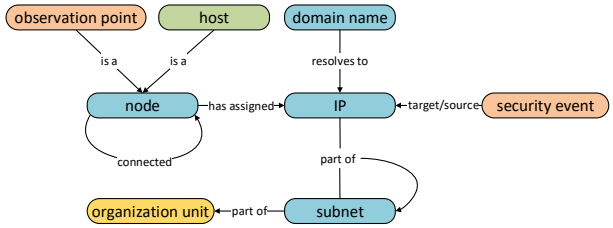

Figure 4: Network Layer

A **node** can be any element in a network, such as a host, router, switch, and gateway. A node can have routing and filtering capability and can be assigned one or more **IP** addresses. The address

can be both IPv4 or IPv6 address. The IP addresses are grouped into **subnet**. A **domain name** holds information about the domain name and its mapping to an IP address.

The fundamental relation is the **connection** between two network nodes. Each **has assigned** an IP address, where one node can have assigned multiple IP addresses (especially IPv6 addresses). As this assignment might be dynamic in time, the relation has a start and an end timestamp to express the exact moment of the assignment. Similarly, the IP address **resolves to** a domain name. Also, this relation can often change in time and needs to carry timestamps since when and to when was this mapping valid. An IP address can further be a **part of** subnet which is transitively **part of** a bigger one.

A host from the host layer **is a** subclass of a node in a network. The virtual and physical subclasses of the host also inherit this relation. The network node can have traffic monitoring or anomaly detection capabilities and then **is an** observation point which server data for a detection system. Another cross-layer relation is the connection of security event from Detection and Response layer which can **target** an IP address or the address can be the **source** of the incident. An administrative connection to the Access Control layer is the subnet being **part of** an organisation unit.

The mostly static division of the network into subnets can be initially inserted by network administrators and then verified and maintained with network topology discovery tools. Combination of those approaches ensures the topology is complete and up to date. Similarly, passive DNS tools can monitor the changes of IP-domain name relations in time and fill the model with relevant data.

The mapping of current routing and filtering rules must be done via active probing of the nodes. The routing rules are essential for impact assessment as they might serve for blocking of an attacker, e.g., by blackholing [25], and they can change rapidly so that the monitoring system must work continuously. The same holds for firewall rules but firewalls are usually proprietary, and we need to implement connectors for them to gather data for the model.

4.4 Detection and Response Layer

The detection and response layer, illustrated in Figure 5, describes the observation capability, current and past incidents, and countermeasures in place. The layer corresponds to an incident response team's point of view. The information included in the layer can be used for further analysis in order to correlate security events to incidents and tracking the current state of an attack. It also allows for response efficiency analysis.

A **security event** is any occurrence with potential security implications, i.e. IDS alert. A **detection system** is any system capable of raising security events, i.e. anomaly detection system. If a detection system is based on observing network traffic, it can consume data from one or more **observation points**. A security **incident** denotes a violation of computer security policies. A **response** is a reactive action to given incident that should mitigate the damage incurred due to the incident and prevent its continuation.

The relations within detection layer are quite straightforward. Observation point server as **data input** for detection system which processes them and **raises** security events. These events are then

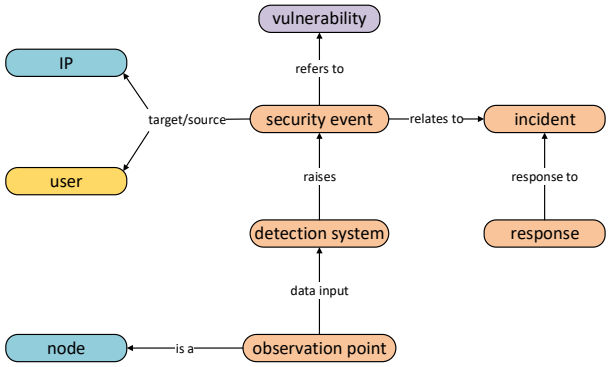

Figure 5: Detection and Response Layer

related to an incident in the system. Finally, a response is deployed in **response to** the incident.

A security event can **refer to** a vulnerability from threat layer or it can **target** an IP address from network layer or user from Access Control layer. The other way around, an IP address or a user can be the **source** of the event. Finally, an observation point **is a** node in the network layer context.

The information about the location and capabilities of intrusion detection systems is held by the security team and does not change often, therefore can be maintained easily. Most detection systems provide capabilities to query the generated security events, which can be utilised to fill the data in an automated fashion. The incidents are usually created after analysis of security event. In case of automated analysis, the incident could be created as result of the analysis (however this strongly depends on the actual implementation). In case of manual analysis, most security teams utilise some tracking systems (an example of such is RTIR[7]) which also support querying for information. Responses can be added as a part of reconfiguration if the reconfiguration is not performed manually (which in most cases is not, the changes in configuration are usually automated).

4.5 Access Control Layer

The access control layer describes the relations between user accounts, applications, and devices, as illustrated in Figure 6. It is described in accord with the most prevalent identity models, such as Microsoft Active Directory, and is capable of modelling role-based access control, which is a proposed standard for modelling access control [12]. The purpose of the layer is to track the permissions of users and groups which is especially important should a user account be compromised when looking for users with privileged access to devices or applications.

An **application** is an identity under which a component operates and a **device** is an identity for a physical device capable of access control, such as PC or printer. There might be specific **permissions** to each application or device, which define a set of operations that can be performed on the application or device. A **role** is an abstract description of a function of a user in that role.

[7] https://bestpractical.com/rtir/

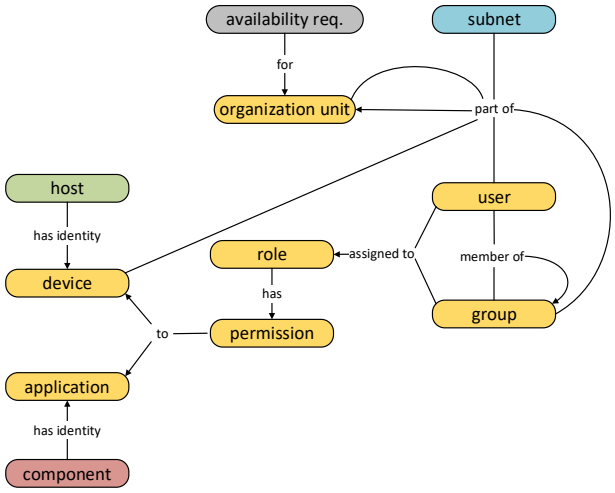

Figure 6: Access Control Layer

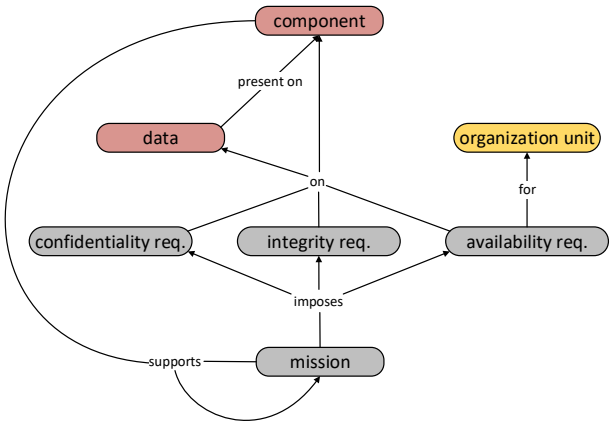

Figure 7: Mission Layer

A **user** is an identity of a person. One person can have one or more user accounts. A **group** is a set of users and groups. Groups themselves can contain other groups. An **organisation unit** is a set of users, devices, and subnets belonging to a given unit in the organisation.

A user, as well as a group, might be **member of** another group. A group, a user, a device, and an organisation unit itself can be **part of** another organisation unit. A user or a group can all have the same function in the organisation and therefore can be **assigned to** a role corresponding to the function. A role can **have** one or more permissions **to** an application or device.

Besides the already described intra-layer relations, there are several relations to other layers. A component capable of access control **has identity** application and similarly a host capable of access control **has identity** device (the system layer and the host layer respectively). A subnet from the network layer may be **part of** an organisation unit and an availability requirement from the Mission layer may be **for** specific organisation unit.

The information about access control layer can be extracted from any central identity management system. Microsoft Active Directory is the most common example of such system. Since the current trends in identity management and access control are federated identities [15] and single sign-on [4], the information can be almost complete. If only incomplete or no information is available, access logs can be used to gain at least partial information.

4.6 Mission Layer

The Mission layer, illustrated in Figure 7, defines the mission elements, their relations, and the requirements that should be satisfied to execute the mission. The purpose of the layer is to track which components support which missions and state the conditions required for mission operation.

A **mission** is the purpose of organisation's operation. The mission can consist of several submissions that specify the operation in higher granularity. A mission can be supported by components. Each mission states the requirements for components and data. Those requirements are divided into three categories based on CIA triad: confidentiality requirements, integrity requirements, and availability requirements. An **availability requirement** offers a way to define for whom should the component be available and how much, i.e. the accounting system component should not be unavailable for accounting department for more than 5 minutes. There can be more than one availability requirement on component based on the audience. A **confidentiality requirement** can be targeted on component or data. The confidentiality of data implies that every component handling the data should be confidential. An **integrity requirement** can also be targeted on component or data and works the same as confidentiality requirement.

There are, in essence, only two relations between entities in the Mission layer. A mission can **support** other missions meaning that the supporting mission needs to be completed for the supported mission to be fulfilled. A mission might **impose** a confidentiality, integrity or availability requirement. Such a requirement must also be fulfilled for the supported mission to be fulfilled.

There are also very few relations to entities from other layers. It ties most with the system layer. A component might also **support** a mission, meaning its functioning is necessary for mission operation. A confidentiality, integrity or availability requirement is imposed **on** either data or component. The availability might be required only **for** users and devices belonging to a particular organisation unit, an entity from the access control layer.

The information about mission and requirements must be maintained and inserted manually. Especially for the purpose of situational awareness in a network of critical information infrastructure (CII), creation and maintenance of the documentation about systems belonging to CII are often required by law. Further, many standards, such as ISO/IEC 27005 [19] from ISO/IEC 27000 family, require determining the mission statement and requirements as an input to the risk assessment process. Therefore there is a reasonable chance that this information will be up to date and available. Another aspect is that the mission statement does not change very often and maintaining it manually is realistic. Medeiros et al. [10]

also discussed the options of manual and automatic capturing of the artefacts, which would complement our proposed model.

4.7 Threat Layer

The threat layer, illustrated in Figure 8, describes the exposure of the network through existing vulnerabilities. It should allow for simple vulnerability tracking, providing information about the location of critical vulnerabilities and helping efficient vulnerability patching.

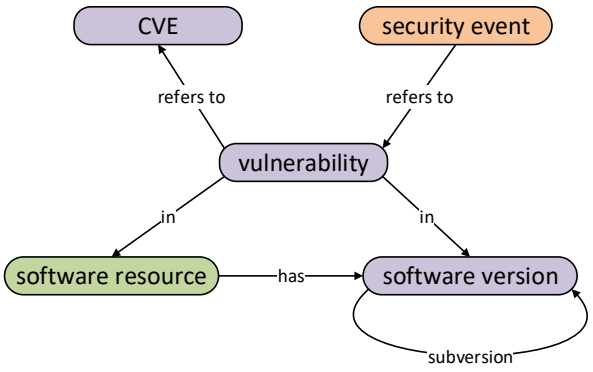

Figure 8: Threat Layer

A **vulnerability** refers to the specific type of vulnerability that is detected in the network. There can be multiple instances of the vulnerability. A vulnerability can be caused both by a misconfiguration or a bug in a software resource. A **software version** denote a name and version of a software in the network, although the version might be only high level. A software version might have several subversions allowing for more grained version system. Since some fingerprinting systems might not be capable of determining exact version of the software, the higher level software versions allow for including their outputs in the model. It enables tracking of all vulnerabilities in the network that are present in a given version of the software. **CVE** refers to a description of a vulnerability in Common Vulnerabilities and Exposures system, including its CVSS rating.

A vulnerability in the system may **refer to** its CVE description and can be present **in** a specific software version. The software version can be a **subversion** of another one.

A connection to host layer a represented by software resource which can **have** a specific software version. Also, a vulnerability can be present **in** the software resource. A security event from Detection and Response layer can **refer to** a vulnerability, i.e. exploit of such vulnerability.

The methods for software version detection range from passive fingerprinting to active network scans and host-based discovery. The accuracy and invasiveness of such sources vary. The most common tool for active network scan and service fingerprinting is Nmap [8]. The vulnerability descriptions can be found for example in National Vulnerability Database (NVD)[9] or in official vendor feeds. The vulnerabilities can be discovered by vulnerability scanners. There are two types of vulnerability scanners: network vulnerability scanners and host vulnerability scanners. The network vulnerability scanners, such as Nessus[10], OpenVAS[11], and Retina[12], use their vulnerability databases and fingerprinting methods to assess the presence of vulnerabilities on a system through remote communication with the host. Such detections are not invasive, but they are less accurate. Host vulnerability scanners [28] are more accurate but require an agent installed on each host. They discover the exact versions and configurations of the software installed on a host and compare it to the list of known vulnerabilities.

[8] https://nmap.org/
[9] https://nvd.nist.gov/

5 CONCLUSION

In this paper, we presented a data model for cyber situational awareness that is to be used by security teams for incident response. The model consists of seven layers, each layer also represents a view on a system from a certain perspective, from network topology to mission statements. Thus, the data model enables better comprehension of a situation by experts in different domains and can serve as the information exchange point for otherwise incompatible tools. The data model is also designed so that the information held by the model could be obtained in a reasonable manner, preferably in an automated fashion. Extensibility was also taken into consideration so that the data model can serve as a backbone for additional information sources. The proposed data model was thoroughly evaluated and discussed with incident handlers from several security teams to reflect the needs of future operational deployment.

The future work will be conducted in two directions. First, we are going to implement an information system for keeping track of the data required for cyber situational awareness. We are going to further examine available data sources and the methods for deriving the data. In addition, methods for merging information from more data sources need to be examined, since multiple heterogeneous data sources can give the same kind of information. Second, we are going to utilise the proposed data model to further analyse the cybersecurity situation. Wide area of tasks can be conducted using the tools based on the proposed data model, such as automated attack impact analysis, cybersecurity situation visualisation, decision support for security teams, and attack mitigation recommendation.

ACKNOWLEDGMENTS

This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20172020070 Research of Tools for Cyber Situational Awareness and Decision Support of CSIRT Teams in Protection of Critical Infrastructure. Martin Laštovička is Brno Ph.D. Talent Scholarship Holder – Funded by the Brno City Municipality.

REFERENCES

[1] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A. Maltz, and Ming Zhang. 2007. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *ACM SIGCOMM Computer Communication Review*, Vol. 37. ACM, 13–24.

[10] https://www.tenable.com/products/nessus
[11] http://www.openvas.org
[12] https://www.beyondtrust.com/products/retina

[2] Paul Barham, Austin Donnelly, Rebecca Isaacs, and Richard Mortier. 2004. Using Magpie for Request Extraction and Workload Modelling. In *OSDI*, Vol. 4. 18–18.

[3] Sean Barnum. 2012. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). http://www.standardscoordination.org/sites/default/files/docs/STIX_Whitepaper_v1.1.pdf. *MITRE Corporation* Version 1.1, Revision 1 (2012).

[4] Luis Barriga-Caceres, Jesus Angel de Gregorio-Rodriguez, Avelina Pardo-Blazquez, and John Michael Walker-Pina. 2007. System, method and apparatus for federated single sign-on services. (May 22 2007). US Patent 7,221,935.

[5] Laurin Buchanan, Mark Larkin, and Anita D'Amico. 2012. Mission assurance proof-of-concept: Mapping dependencies among cyber assets, missions, and users. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*. IEEE, 298–304.

[6] Xu Chen, Ming Zhang, Zhuoqing Morley Mao, and Paramvir Bahl. 2008. Automating Network Application Dependency Discovery: Experiences, Limitations, and New Solutions. In *OSDI*, Vol. 8. 117–130.

[7] Yen-Yang Michael Chen, Anthony J. Accardi, Emre Kiciman, David A. Patterson, Armando Fox, and Eric A. Brewer. 2004. Path-based failure and evolution management. (2004).

[8] Cosmin Ciobanu, Don Stikvoort, Miroslaw Maj, Tomasz Chlebowski, Roeland Reijers, and Mirko Wollenberg. 2013. Alerts, Warnings and Announcements. Best Practices Guide. https://www.enisa.europa.eu/publications/awa/at_download/fullReport. (Nov. 2013).

[9] Anita D'Amico, Laurin Buchanan, John Goodall, and Paul Walczak. 2010. Mission impact of cyber events: scenarios and ontology to express the relationships between cyber assets, missions and users. In *International Conference on Cyber Warfare and Security*. Academic Conferences International Limited, 388. 33 cit.

[10] A. K. A. de Medeiros, W. M. P. van der Aalst, and A. J. M. M. Weijters. 2003. Workflow Mining: Current Status and Future Directions. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Springer Berlin Heidelberg, Berlin, Heidelberg, 389–406.

[11] Mica R. Endsley. 1988. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society annual meeting*, Vol. 32. SAGE Publications Sage CA: Los Angeles, CA, 97–101.

[12] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. 2001. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4, 3 (2001), 224–274.

[13] Rodrigo Fonseca, George Porter, Randy H. Katz, Scott Shenker, and Ion Stoica. 2007. X-trace: A pervasive network tracing framework. In *Proceedings of the 4th USENIX conference on Networked systems design & implementation*. USENIX Association, 20–20.

[14] Bingrui Foo, Y.-S. Wu, Y.-C. Mao, Saurabh Bagchi, and Eugene Spafford. 2005. ADEPTS: adaptive intrusion response using attack graphs in an e-commerce environment. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*. IEEE, 508–517.

[15] Martin Gaedke, Johannes Meinecke, and Martin Nussbaumer. 2005. A modeling approach to federated identity and access management. In *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 1156–1157.

[16] John R. Goodall, Anita D'Amico, and Jason K. Kopylec. 2009. Camus: automatically mapping cyber assets to missions and users. In *Military Communications Conference, 2009. MILCOM 2009. IEEE*. IEEE, 1–7.

[17] Jared Holsopple, Shanchieh Jay Yang, and Brian Argauer. 2008. Virtual terrain: a security-based representation of a computer network. In *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*. 69730E. 12 cit.

[18] Frank Innerhofer-Oberperfler and Ruth Breu. 2006. Using an Enterprise Architecture for IT Risk Management. In *ISSA*. 1–12.

[19] ISO 27005:2011(E) 2011. *Information technology – Security techniques – Information security risk management*. Standard. International Organization for Standardization, Geneva, CH.

[20] Sushil Jajodia, Peng Liu, Vipin Swarup, and Cliff Wang. 2010. *Cyber situational awareness*. Vol. 14. Springer.

[21] Sushil Jajodia, Steven Noel, Pramod Kalapa, Massimiliano Albanese, and John Williams. 2011. Cauldron mission-centric cyber situational awareness with defense in depth. In *Military Communications Conference, 2011-MILCOM 2011*. IEEE, 1339–1344.

[22] Gabriel Jakobson. 2011. Extending situation modeling with inference of plausible future cyber situations. In *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*. IEEE, 48–55.

[23] Gabriel Jakobson. 2011. Mission cyber security situation assessment using impact dependency graphs. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE, 1–8.

[24] Alexander Kott, Cliff Wang, and Robert F. Erbacher. 2015. *Cyber defense and situational awareness*. Vol. 62. Springer.

[25] W. Kumari and D. McPherson. 2009. *Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)*. RFC 5635.

[26] Martin Laštovička, Tomáš Jirsík, Pavel Čeleda, Stanislav Špaček, and Daniel Filakovský. 2018. Passive OS Fingerprinting Methods in the Jungle of Wireless Networks. In *2018 IEEE/IFIP Network Operations and Management Symposium (To appear)*.

[27] Zhijie Liu, Chongjun Wang, and Shifu Chen. 2008. Correlating multi-step attack and constructing attack scenarios based on attack pattern modeling. In *Information Security and Assurance, 2008. ISA 2008. International Conference on*. IEEE, 214–219.

[28] Mingchao Ma, Daniel Kouřil, Michal Procházka, Cyril L'Orphelin, Olivier Lequeux, Pierre Veyre, Christos Triantafyllidis, Christos Kanellopoulos, and Paschalis Korosoglou. 2012. EGI Security Monitoring. In *International Symposium on Grids and Clouds proceedings*, Vol. 1.

[29] Miroslaw Maj, Roeland Reijers, and Don Stikvoort. 2010. Good Practice Guide for Incident Management. https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management. (20 Dec. 2010).

[30] Benjamin Morin, Ludovic Mé, Hervé Debar, and Mireille Ducassé. 2002. M2D2: A formal data model for IDS alert correlation. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 115–137. 299 cit.

[31] Tamsin Moye, Reginald Sawilla, Rodney Sullivan, and Philippe Lagadec. 2015. Cyber Defence Situational Awareness Demonstration/Request for Information (RFI) from Industry and Government (CO-14068-MNCD2). *NCI Agency Acquisition* (2015).

[32] Arun Natarajan, Peng Ning, Yao Liu, Sushil Jajodia, and Steve E. Hutchinson. 2012. *NSDMiner: Automated discovery of network service dependencies*. IEEE.

[33] S. Noel, E. Harley, K. H. Tam, M. Limiero, and M. Share. 2016. CyGraph: Graph-Based Analytics and Visualization for Cybersecurity. *Handbook of Statistics* 35 (2016), 117–167. 2 cit.

[34] Cyril Onwubiko. 2009. Functional requirements of situational awareness in computer network security. In *Intelligence and Security Informatics, 2009. ISI'09. IEEE International Conference on*. IEEE, 209–213.

[35] Sven Ossenbühl, Jessica Steinberger, and Harald Baier. 2015. Towards automated incident handling: How to select an appropriate response against a network-based attack?. In *IT Security Incident Management & IT Forensics (IMF), 2015 Ninth International Conference on*. IEEE, 51–67.

[36] Xinming Ou, Sudhakar Govindavajhala, and Andrew W. Appel. 2005. MulVAL: A Logic-based Network Security Analyzer. In *USENIX Security Symposium*. Baltimore, MD, 8–8.

[37] Ali Zand, Giovanni Vigna, Richard Kemmerer, and Christopher Kruegel. 2014. Rippler: Delay injection for service dependency detection. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2157–2165.

[38] Bin Zhu and Ali A. Ghorbani. 2006. Alert correlation for extracting attack strategies. *International Journal of Network Security* 3, 3 (Nov 2006), 244–258.