Establishing National Cyber Situational Awareness through Incident Information Clustering

Florian Skopik, Markus Wurzenberger, Giuseppe Settanni, Roman Fiedler
Safety and Security Department
AIT Austrian Institute of Technology
A-1220 Vienna, Austria
firstname.lastname@ait.ac.at

***Abstract*—The number and type of threats to modern information and communication networks has increased massively in the recent years. Furthermore, the system complexity and interconnectedness has reached a level which makes it impossible to adequately protect networked systems with standard security solutions. There are simply too many unknown vulnerabilities, potential configuration mistakes and therefore enlarged attack surfaces and channels. A promising approach to better secure today's networked systems is information sharing about threats, vulnerabilities and indicators of compromise across organizations; and, in case something went wrong, to report incidents to national cyber security centers. These measures enable early warning systems, support risk management processes, and increase the overall situational awareness of organizations. Several cyber security directives around the world, such as the EU Network and Information Security Directive and the equivalent NIST Framework, demand specifically national cyber security centers and policies for organizations to report on incidents. However, effective tools to support the operation of such centers are rare. Typically, existing tools have been developed with the single organization as customer in mind. These tools are often not appropriate either for the large amounts of data or for the application use case at all. In this paper, we therefore introduce a novel incident clustering model and a system architecture along with a prototype implementation to establish situational awareness about the security of participating organizations. This is a vital prerequisite to plan further actions towards securing national infrastructure assets.**

I. Introduction

Information and communication technology (ICT) is a vital part of today's critical infrastructures, including energy and water supply, finance, transportation and health care management systems, and underpin their smooth operation and high reliability. ICT is not only used to enable remote monitoring and maintenance, but also to connect systems and services that have been operated isolated in the past [1]. The application of Cloud computing and mobile accessibility additionally increases the complexity of those systems. As a consequence, also attack surfaces and channels have multiplied. Furthermore, software products that are developed under time- and cost-pressure have led to an enormous amount of publicly known (and most likely even more unknown [2]) vulnerabilities. In recent years, the number of long lasting targeted attacks with high impact [3], so-called advanced persistent threats (APTs) [4], has significantly increased.

Appreciating this, a number of organizations have developed standards and recommendations for tackling this serious security situation. Specifically, NIST [5], ITU-T [6] and ISO [7] have proposed information sharing mechanisms, possibly with centrally coordinating entities (i.e, national or industry sector-specific cyber security centers), to which affected organizations shall report incidents, exchange network monitoring data and status information [8] of critical services across organizational boundaries. The main goal of these efforts is to create an extensive *situational awareness picture* about potential threats and ongoing incidents, which is a prerequisite for effective preparation and mitigation in large-scale incidents. Eventually, the successful operation of such a cyber center can potentially decrease the required budget for cyber security (since reported incidents and exploited vulnerabilities will result in early warnings to others) and increase the effectiveness in terms of timeliness and success of mitigation measures.

National initiatives, such as Computer Emergency Response Teams (CERTs), as well as open and commercial Internet platforms, e.g., Internet Storm Center or Arbor Networks, perform invaluable work by collecting relevant security information, running awareness campaigns and providing up-to-date information on cyber security incidents and their mitigation. However, these information is usually quite generic, not shaped to particular industries and often lacks in depth knowledge. In order to make such platforms more effective, sector-specific views along with rich information and experience reports are required to provide an added value to professional users; e.g., What are the most important holes to close in industrial control systems software? What are currently the most common attack vectors in the energy domain? Which malware has caused the biggest trouble in the last week in the banking sector? What new phishing types are currently under way? In order to provide timely answers to such quite specific questions concerning national security, many countries around the world are building up cyber security centers.

In this paper we therefore deal with an approach and prototype system providing support to the most elementary task of a cyber center: establishing cyber situational awareness to issue warnings and derive further steps, being proactive security measures or mitigation strategies. For this purpose, we assume efficient information sharing procedures, such as [9], [7] are already in place and information on past and ongoing incidents are stored centrally. In detail, this paper comprises the following contributions:

- *Incident Clustering Model.* We introduce a theoretical model for incident classification and clustering used to create a situational overview from massive amounts of incident reports.
- *System Architecture and Implementation.* We outline the system architecture of a prototype implementation used to evaluate our approach to establishing situational awareness.
- *Evaluation and Findings of Incident Clustering.* We study our approach in terms of both, scalability with respect to the size of input data sets and applicability in realistic settings.

The remainder of this paper is structured as follows: Section II deals with background and related work. Section III introduces the conceptual model of incident clustering and the basic mode of operation. Then, Sect. IV shows the system architecture of the current prototype. After that, the evaluation results and some general findings are provided in Sect. V. Eventually, Sect. VI concludes the paper.

II. Background and Related Work

Cyber-attacks are becoming increasingly sophisticated, targeted and coordinated, resulting in so-called Advanced Persistent Threats (APTs) [4]. Consequently, new paradigms are required for detecting and mitigating this kind of attacks, and eventually to establish situational awareness [10]. Many of these tasks are currently performed within individual organizations only, and – apart from the important works that national CERTs do – there is little cross-organizational security information sharing. However, information sharing is a crucial step to acquiring a thorough understanding of large-scale cyber-attack situations; and eventually cyber situational awareness is necessary to warn others against threats and make informed decisions. In that sense, cooperative cyber defense [8], [11], [12] has been studied in the recent years, yet their broad adoption is still missing. Besides the involved risks for reputation damage, different data formats to describe cyber incidents and related information make interoperability a serious issue.

Central to taking an informed and coordinated approach to cyber security incidents is determining situational awareness (SA). A number of models of SA exist [13][14][15], but arguably the most pervasive is that proposed by Endsley [13], which describes three increasing levels of awareness: *perception*, *comprehension*, and *projection*. As one advances through these levels, decision making capabilities are improved. In the paper at hand, we mainly deal with the first two levels, perception and comprehension.

Standards bodies and the like have produced volumes about how to establish security information sharing networks and situational awareness on a large scale - the canonical examples being the NIST guideline 'Framework for Improving Critical Infrastructure Cybersecurity' [5], the ENISA documents 'Good Practice Guide on Information Sharing' and 'Cybersecurity cooperation: Defending the digital frontline' (just to name two of the many available guidelines from ENISA), or the ISO/IEC standard 27010 'Information technology - Security techniques - Information security management for inter-sector and inter-organizational communications' [7]. Whilst representing important work, these recommendations are not the complete picture and important pieces are still missing. For instance, current recommendations largely take an architectural (and partly organizational) view on the problem, and omit guidance on operational aspects of enabling security information sharing. Little attention is given to the technologies and processes that are needed to maintain situational awareness for these potentially complex cyber systems.

III. Incident Clustering Model

We assume, a reporting policy of organizational security incidents to a (national) cyber security center is in place (as discussed in numerous national cyber security strategies like [5] and [16]), and this entity is collecting incoming reports over a long time span. Since with the expected large amounts of data it will be hard to keep track of overall trends and manual processing is time- and resource-intensive, there is a strong need for an automatic machine-supported approach to incident management in order to guarantee fast reaction and quick decision making. Therefore, the heart of our work to establishing cyber situational awareness is a sophisticated clustering mechanism of incident messages from multiple organizations, which provides an overview about incident classes and effectively groups them according to common properties.

The concrete implementation is based on the MANTIS Cyber-Intelligence Management Framework[1] (MANTIS) as the storage backend for incident reports. On top of that, our approach implements the classification and clustering mechanisms, as well as a visualization and statistics components. Moreover, a trend analysis of cyber attacks is enabled by applying filters on datasets. The building blocks of the proposed system, which are run through continuously, are: (i) *reporting interface and database*, to receive incident reports from organizations; (ii) *clustering algorithm* to perform the actual data processing; (iii) *report generation*, summarizes essential statistics on the calculated clusters and

[1] `http://django-mantis.readthedocs.org`

forms the actual cyber situational awareness picture; (iv) *visualization* provides an overview of the generated report, which is vital for fast decision making; and (v) *trend analysis support* allows to enrich the situational awareness picture with temporal properties and latest developments, such as emerging types of incidents.

A. Reporting Interface and Database

According to the state of the art, there are numerous data exchange formats for cyber incidents, such as IODEF and STIX. In order to allow interoperability between these formats, MANTIS makes use of a number of importers which enable the integration of incident reports in various formats, and stores them accordingly in one harmonized database schema.

The illustrative applied schema foresees entries consisting of 13 facts, as summarized in Table I. We selected these 13 facts, because they describe the properties of a cyber incident which are most important for obtaining a comprehensive situational awareness picture; they include the affected (technical) system and business area in which it appeared, a time stamp and the severity of the incident. Using the common vulnerabilities and exposures (CVE) ID incidents can be linked to technical vulnerabilities which were exploited. Furthermore, there are enough numerical attributes, to be utilized to organize the incidents in clusters.

B. Clustering Algorithm

To identify incident reports clusters, we argue that an *Agglomerative Hierarchical Cluster Analysis* (HCA) (see e.g., [17]) is the most suitable choice for our approach. HCA proved to have higher performance compared to other existing clustering methods. Most of the other algorithms, such as the *k-means* [18], have the disadvantage that the number of clusters must be set before running the algorithm. Since the number of cyber incidents increases continuously, and so might the number of obtained clusters, it is not appropriate for our approach to adopt such a clustering method. Furthermore, other algorithms, which determine the number of clusters themselves, often calculated an inappropriately small number of clusters in our tests. The reason for this is the high density of the input data, i. e. the elements are concentrated within a small range of values.

As linkage criteria we picked the common complete linkage method [19], which means in each step the two clusters with the smallest maximum distance d between all elements $D_{complete}$ (cf. Eq. 1) are merged.

$$D_{complete}(X, Y) := \max\{d(x, y) | x \in X, y \in Y\} \quad (1)$$

According to the data facts presented in Table I, there are 3 numerical facts which are suitable for clustering: *Time Until Discovery*, *Time Until Report* and *Damage Euro*. Since *Damage Score* is of type enumeration and thus a classification with exactly 10 classes, it makes no sense running a clustering algorithm concerning this attribute. Hence there is the possibility to cluster for one, two or all three attributes mentioned before.

In our approach, every data base entry corresponds to a point in space, whereby the space dimension N depends on how many attributes have been selected for clustering. Therefore, it is possible to calculate a symmetric distance matrix $Dist$ (Eq. 2) of the euclidean distances [20] (cf. Eq. 3) between all entries. Due to the symmetry of $Dist$ the calculation can be reduced to a lower triangular matrix. Additionally, $d(x_i, x_i) = 0$ holds for all $x_i \in X$.

$$Dist = \begin{pmatrix} 0 & d(x_1, x_2) & \cdots & d(x_1, x_n) \\ d(x_2, x_1) & 0 & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ d(x_n, x_1) & d(x_n, x_2) & \cdots & 0 \end{pmatrix} \quad (2)$$

$$d(x, y) = \sqrt{\sum_{j=1}^{N} (x_j - y_j)^2} \quad (3)$$

Afterwards we run an HCA based on the distance matrix $Dist$ for calculating a cluster tree. This works as follows: First, every data base entry forms a cluster by itself. Then, in every step two clusters, selected by using the linkage criteria presented in Eq. 1, (where d is the the euclidean distance from Eq. 3), are merged. The procedure is repeated until there is just one cluster left.

The next step is to determine the optimal number of clusters. Therefore, we first set a condition (cf. Eq. 5), which every cluster C_i (i is the cluster number) has to fulfill: the

Table I
DATA FACTS AND THEIR DESCRIPTION.

Fact	*Type*	*Description*
Action	enumeration	the kind of incident, e.g., hacking, misuse, etc.
Actor	enumeration	differentiates internal and external actors
Company	free text	affected company name and type
CVE ID	unique	identifier, which links the exploited vulnerability in the Common Vulnerabilities and Exposures[2] (CVE) data base
Damage Euro	number	financial loss (in Euro)
Damage Score	enumeration	an integer between 1 and 10 which indicates the severity of the incident for the affected organization
Incident Date	number	date on which the incident has been reported
Incident ID	unique	identifier of the incident
System	free text	affected (technical) system
Time Until Discovery	number	time between the occurrence and the detection of the incident (in seconds)
Time Until Report	number	time between *Vulnerability Date* and *Incident Date* (in seconds)
Vulnerability Date	number	date on which the vulnerability, which was exploited, has been reported
Business Area	enumeration	*company's* field of business

[2] https://cve.mitre.org

Table II
CLUSTERING STATISTICS AND THEIR DESCRIPTION.

Metric	*Type*	*Description*
number of entries	number	number of entries
number of clusters	number	number of clusters
clusters size*	vector	number of elements per cluster
clusters diameter*	vector	maximum within cluster distances
clusterwise within cluster average dist.	vector	average distance between the elements of one cluster
clusterwise within cluster median dist.	vector	median distance between the elements of one cluster
separation vector*	vector	minimum distance of a point in the cluster to a point of another cluster
separation matrix	matrix	matrix of separation values between all pairs of clusters
average toother	vector	average distance of a point in the cluster to the points in other clusters
mean dissimilarities	matrix	matrix of mean dissimilarities between points of every cluster
average distance between clusters	number	average distance between all clusters
average distance within clusters	number	average of within cluster distances of all clusters
gaps	vector	widest within cluster gap

distance d between all elements inside one cluster C_i has to be smaller than α (cf. Eq. 4), which is a proportion of the maximum distance between all elements. In order to find the optimal number of clusters, the tree has to be cut until all clusters fulfill the condition in Eq. 5. This means we first check if the condition in Eq. 5 is respected having only one cluster. If not two clusters are considered and the condition is tested within each of them. This step is repeated until all clusters fulfill the condition in Eq. 5.

$$\alpha = a \cdot \max(Dist), \qquad a^3 \in (0, 1) \tag{4}$$

$$d(x, y) < \alpha, \qquad \text{for all } x, y \in C_i \tag{5}$$

In order to generate situational awareness security operational centers need to process vast amounts of incident reports collected from different sources. In some isolated cases, running the described clustering algorithm, one incident assigned to a cluster can happen to fit better into a neighbor cluster. We argue that given the large size of the data set, these unusual cases do not significantly affect the clustering quality and are tolerable in favor of simplicity and performance of the approach.

C. Report Generation

The application of the clustering approach presented before enables us to gain broader knowledge from collected incidents. This knowledge is modeled as two kinds of statistical reports:

1) overall dataset clustering properties,
2) individual cluster properties, i.e., statistics about each single cluster C_i.

The first report comprises the statistics given in Table II; these statistics convey information on how the elements of the data-set are distributed among the generated clusters. In the second report for each cluster, the maximum value, the minimum value, the mean and the median value of *Damage Score*, *Time Until Discovery*, *Time Until Report* and *Damage Euro* are calculated. These values provide basic knowledge on the incidents composing each cluster C_i.

D. Visualization

An essential component of the proposed approach is the clusters visualization. Visualizing the obtained clusters graphically eases analysts' tasks by allowing to instantly have an overview on the considered incident information and by enabling graphical-supported analysis. Depending on the number of clustered attributes it is possible to create a 'one-dimensional', two-dimensional or three-dimensional graph. Up to three spatial dimensions can therefore visualize up to three incident attributes. Moreover, additional non-spatial dimensions can be represented by changing the color, the size, the shape, the transparency, or other characteristics of the data points. In Figure 1 five dimensions are visualized: *Damage Euro* is represented on the x-axis, *Time Until Discovery* on the y-axis and *Time Until Report* on the z-axis; we set the same color for all entries belonging to the same cluster and map the node size to the *Damage Score*. Analysts can inspect the clusters by rotating the graph and by zooming in or out; a click on one node redirects the user to the incident report page, where detailed information on the selected incident report are listed.

E. Trend Analysis Support

Trend analysis can be performed by employing the provided filtering function. Users can filter the incidents data set by *Vulnerability Dates* and *Incident Dates* to select a given time interval, and obtain clusters and statistics pertaining incidents occurred in that specific time frame. Graphs and statistics related to consequent time frames can be analyzed to deduce trends. For instance, the creation of time series enables the detection of change patterns in incident reports and can significantly contribute in the prediction of future trends.

Trend analysis is a valuable method to also outline possible mitigation procedures in the attempt of solving incidents

*Additionally, the minimum, the maximum, the mean and the median are calculated

[3]Empirical studies have shown that $a = 0.2$ is a good option.

belonging to the same cluster or to a group of neighboring clusters.

IV. System Implementation

We implemented our system inside a virtual machine using Oracle VM VirtualBox. As operating system we used Ubuntu 12.04, which is recommended by the developers of MANTIS[4] platform utilized as backend.

A. Data Backend with MANTIS

The proposed system is embedded in MANTIS (Model-based Analysis of Threat Intelligence Sources), which is based on Django[5]. For this reason, Python code can be added to MANTIS quite easily. On the one hand, MANTIS can be used as data backend and on the other hand, it provides a browser framework, which can be employed for querying the database. A big advantage of MANTIS is that it is open source and can be simply customized.

There are already importers implemented for common incident data formats, e.g., STIX and OpenIOC. For our approach, we implemented a custom importer to populate the MANTIS database tables with data following the structure mentioned in Section III-A. The designed importer acquires the incident data stored in an XML file and stores them into the MANTIS database.

B. Algorithm Implementation

The main functional blocks of the clustering algorithm we proposed are the following:

1) calculating the distance matrix $Dist$ (cf. Eq. 2)
2) hierarchical clustering (complete-linkage)
3) calculating the optimal number of clusters
4) generating reports, i.e. calculating statistics
5) generating a clustering graph.

Calculating the Distance Matrix: Before calculating the distance matrix $Dist$, the logarithm of the data attributes to be clustered, is calculated. This is necessary in order to be able to compare data spanning a wide range of values. For calculating $Dist$, the `pdist` function, which is provided by the SciPy[6] library, is used.

Hierarchical Clustering (Complete-Linkage): For calculating the HCA the `linkage` function, which is provided by the fastcluster[7] library, is used. For efficiency the core of this library is implemented in C++. Hence, this cluster calculation provides higher performance than the corresponding Python implementation. The distance matrix $Dist$ serves as input to the linkage function.

Calculating the Optimal Number of Clusters: A big advantage of the SciPy library is that it already includes a function which allows to cut the clustering tree by using a condition such as Eq. 5. The function is named `fcluster` and forms so called *flat clusters* using the hierarchical clustering, defined by the linkage matrix, which is calculated by the `linkage` function, i. e. the output of the HCA. Furthermore, it is possible to choose between several criteria for cutting the tree and obtain the right number of clusters. A threshold has therefore to be defined. In our approach, the chosen cutting criterion is the *distance*, and $0.2 \cdot \max(Dist)$ serves as threshold.

Report Generation: The cluster elements' properties are calculated using the `min`, `max`, `mean`, `median` functions provided by NumPy[8]. The easiest way to calculate the statistics which describe the cluster properties, is to use the rpy2[9] library. It enables R[10] functions for Python. Thus, we are able to use the powerful R function `cluster.stats`, which is provided by the R package `fpc`. Finally all statistical metrics mentioned in Section III-C can be calculated by using the `cluster.stats` function.

Cluster Graph Generation: Based on the results of the HCA and depending on the selected number of attributes for the clustering, a '1D', 2D or 3D plot is generated. As MANTIS provides the means to host our visualization subsystem, the plot is generated after loading the clustering tab on the MANTIS web-page. For obtaining the graph we adopt the JavaScript library *CanvasXpress*[11]. Besides its straightforward handling, CanvasXpress library offers simple ways to implement mouseover and mouseclicking events.

C. Operator View Implementation:

Since MANTIS was designed with great extensibility in mind, our incident clustering approach is embedded as a so-called MANTIS application. In the current implementation, the user can pick one to three metrics[12] for clustering, which are Time Until Discovery, Time Until Report, and Financial Loss (*Damage Euro*). Since the visualization of the clustering results is usually the first artifact a human operator is interested in, plots are placed in the upper part of the web-page; detailed clustering statistics are placed below the graph. The *Incident ID* and the values of the clustering attributes are visualized when moving the mouse cursor over a data-point inside the graph. The full incident description related to one data-point is shown in a new window by clicking on the point. Moreover, it is possible to choose between several filtering options and run the clustering algorithm only on a specific subset of incidents selected through multiple-step filtering. According to the filtering results, also the number of selected elements is visualized.

[4] `http://django-mantis.readthedocs.org/en/latest/`
[5] `https://www.djangoproject.com`
[6] `http://www.scipy.org`
[7] `https://pypi.python.org/pypi/fastcluster`
[8] `http://www.numpy.org/`
[9] `http://rpy.sourceforge.net`
[10] `http://www.r-project.org`
[11] `http://canvasxpress.org`
[12] In this paper we select only 3 illustrative metrics, but this list is easily extendable. The algorithm allows the introduction of any number of measurable metrics relevant for the reported incident.

Below the plot the previously selected clustering statistics are printed. Figure 1 shows a screen-shot of the clustering view including the clusters graph, the incidents filter menu, the available clustering options and the selected clustering statistics. A filter on *Vulnerability Date* is applied on a 5000 incidents dataset obtaining only (the 1228) incidents related to vulnerabilities discovered after the 1st of February 2014. These 1228 incidents are clustered, according to the 3 selected attributes, in 17 clusters; the clusters graph is depicted in the corresponding 3D plot alongside the clustering statistics table.

V. Evaluation and Application

In order to generate illustrative examples and evaluate our approach, we deployed our system inside a virtual machine having 4 GB base memory, running Ubuntu 12.04 (as suggested in the MANTIS installation guide). The host machine is an Intel(R) Core(TM) i7-3540M CPU @ 3.00GHz, 3001 Mhz, 2 Cores, 4 Logical Processors with 8 GB physical memory (RAM) running Microsoft Windows 7 Enterprise. Moreover, we recommend to use Google Chrome as web-browser for visualizing the MANTIS page, because it has proved to be faster than other web-browsers in generating detailed graphics, such as the CanvasXpress plot, and run some of the CanvasXpress function, including mouseover and zooming.

A. Data Set Generation

Since it is generally difficult to get access to real cyber-incident data we decided to create a realistic synthetic data-set. First, there is only access to real incidents on a large sale. Second, buying data from a single company would prevent us from having diversity in the incident statistics, due to the lack of information sharing across organization boundaries.

For creating our synthetic incident data (that follows the schema in Table I), we implemented a short Java procedure that creates data-sets of any size. The origin of every synthetic incident is a real vulnerability represented by a CVE database entry[13]. We create on average 3.3 entries for each vulnerability, e.g., a synthetic data-set consisting of 1000 entries is based on 300 real vulnerabilities. Next, we generate the *Damage Scores*, using a number generator, based on a geometric distribution[14] with parameter success probability $p = 0.25$. The *Time Until Discovery*, the *Time Until Report* and the *Damage Euro* are generated by using a number generator based on a normal distribution, where the mean and the variance depend on the *Damage Score*. The *Business Area* is randomly chosen out of a given set of business fields. After that, depending on the *Business Area*, the *Company* is also randomly selected within a given set. Finally to every synthetic incident is assigned an *Incident ID*.

[13] https://cve.mitre.org/

[14] $\mathbb{E} = \frac{1-p}{p}$ and $\mathbb{V} = \frac{1-p}{p^2}$

Since we only choose the distribution for generating the numeric attributes, our algorithm is also suitable to work on future (real) data-sets. Indeed, it is realistic that most of the incidents have *Damage Score* around 3 and 4 and there are only a few incidents with *Damage Score* close to 10. Additionally, as the *Damage Score* is an indicator of the severity of an incident, the values of *Time Until Discovery*, *Time Until Report* and *Damage Euro* depend on it.

B. Performance and Scalability

Incident handling is one of the principal activities that a Security Operational Center (SOC) has to constantly carry out. Operators and analysts employed at SOCs need to have a comprehensive picture of the monitored infrastructures in order to timely and efficiently develop mitigation strategies suitable to the reported incidents. This implies that automated incident analysis procedures (e.g., incident clustering) supporting the analysts must assure high performance and cannot be time-demanding.

We tested the implemented system with 4 synthetic data-sets of different size in order to assess its performance and scalability.

Specifically, we measured the time (in seconds) the different steps of the algorithm need to be carried out, and we measured then the average time required for:

1) calculating the distance matrix
2) performing the HCA
3) calculating the optimal number of clusters
4) executing the whole clustering (consists of points 1-3)
5) calculating the statistics
6) visualizing the clustering results
7) running the overall algorithm
8) running the overall algorithm, without calculating the statistics.

We performed the test 5 times for each data-set, and calculated then the average of the measured durations. Table III summarizes the results of such performance assessment and scalability tests.

The shortest time is consumed for calculating the distance matrix, calculating the optimal number of clusters and for the visualization of the results. These three functional components present the highest scalability degree. We noticed that the overall clustering time increases rapidly from 5000 data-set elements on. The most time consuming step of the clustering algorithm is the HCA. We can observe that the step that calculates the clustering statistics is the one having the worst scalability.

Looking at the overall running time without calculating the statistics and considering that the system runs inside a virtual machine, we can conclude that the system performance is still promising for 10000 data-set entries, indeed we are able to obtain the clustering results and to represent them in less than 6 seconds.

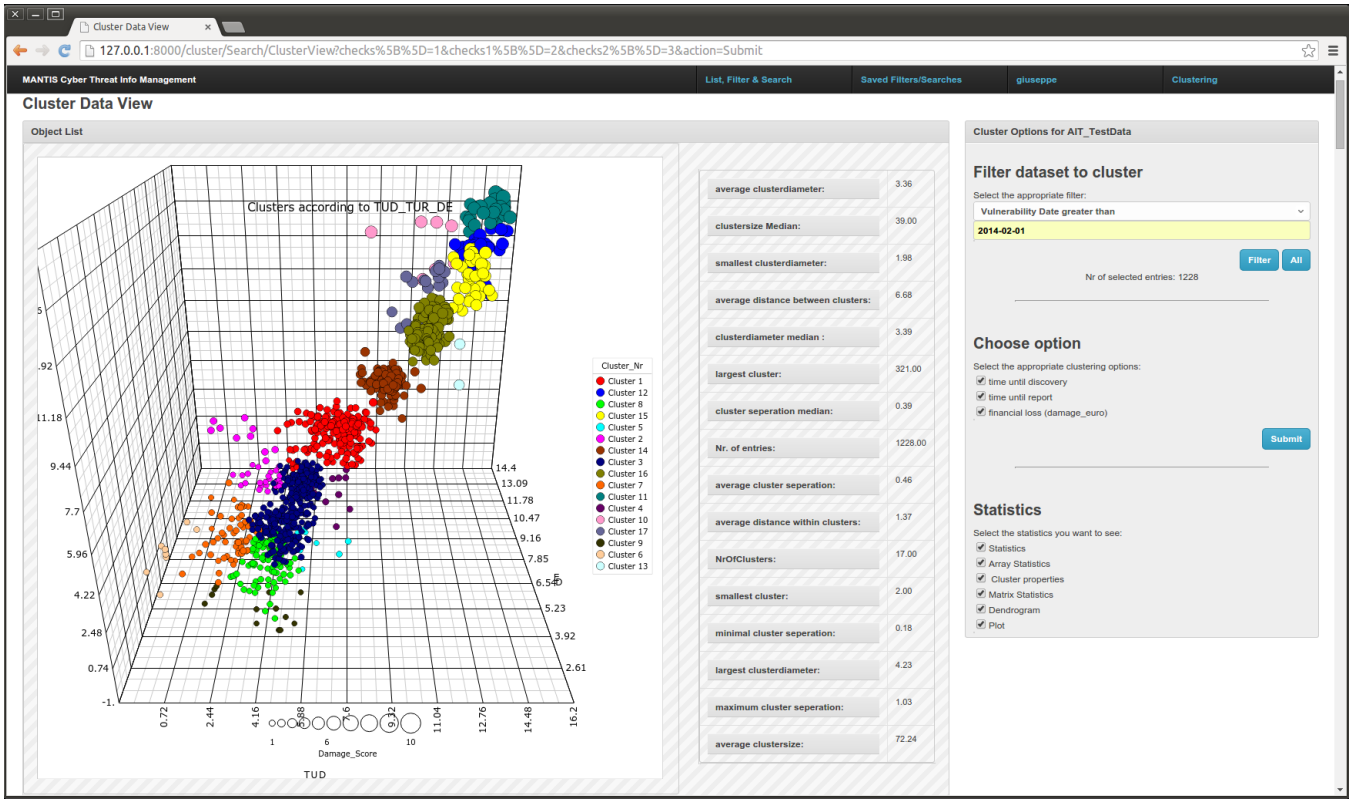

Figure 1. Clustering view: incidents filter, clustering options and statistics can be set in this view. Clustering according to *Damage Euro* (x-axis), *Time Until Discovery* (y-axis), *Time Until Report* (z-axis), performed on 1000 incidents. All points in one cluster have the same color. The size of the nodes indicates the *Damage Score*.

Table III
PERFORMANCE RESULTS IN SECONDS.

Step / Num. of Entries	*100*	*1000*	*5000*	*10000*
cluster overall	0,001215	0,028427	0,977875	4,613557
distance matrix	0,000241	0,006355	0,135251	0,527426
HCA	0,000417	0,018681	0,813758	3,993861
optim. num. of clusters	0,000557	0,003392	0,028865	0,092270
statistics	0,024852	0,894656	19,725982	-
visualization	0,000331	0,010292	0,009924	0,030215
overall	0,375326	1,354036	21,516662	-
overall (w/o stats)	0,350474	0,459380	1,790680	5,980122

C. Output Interpretation in an Application Scenario

Going through a massive amount of incident reports and addressing them individually within a limited time-frame will usually not be feasible for a SOC operator. To ease this function, our system calculates a list of statistics related to the overall clustered data-set, and also about the characteristics of the elements populating each cluster. There are literally hundreds of ways to exploit the calculated statistics to make informed decisions in a cyber security center. Here[15], we only describe some potential interpretations to illustrate the use of the proposed system.

The **number of clusters** provide valuable information on how the massive amount of reported incidents can be grouped according to common properties. Bundled incidents do not necessarily need to be addressed individually but can be addressed as groups and thus more efficiently.

The **size of cluster** provides information on the most common incidents and their ground vulnerabilities respectively. Addressing concerned vulnerabilities, e.g., through campaigns on software quality management, can significantly reduce the overall number of (future) incidents. The size is also an important metric to decide on where to invest spares mitigation resources.

The **cluster diameter** provides information about the diversity of incidents. For small diameters we can assume rather similar incidents and thus expect common mitigation strategies, such as information campaigns on how to counter certain types of attacks, be rather effective. Clusters with large diameter mean that concerned incidents have high variability of time until discovery, time until report or financial loss. So, either reporting needs to be enforced better in the given cases or specific help on discovering incidents must be provided by the cyber center. In these cases our system can be further employed to perform a deeper inspection of the large-diameter clusters. Firstly consecutive filters on the different incidents' attributes can be applied in order to

[15]Also compare the given metrics with Table II and Figure 1.

select all the incidents belonging to the cluster of interest. Then a further clustering process can be executed on the obtained incidents subset. The resulting sub-clusters and the respective statistics can be very useful in better identifying groups of similar incidents and suggest specific mitigation measures for the most prominent ones.

Statistics according to the **distances between clusters**, such as the **separation** and the **separation vector** are indicators for the similarity of two clusters. Hence, if the values of this statistics are low, the statistics which describe the properties of the elements of these clusters are similar. Same sort of information provide the average toother and the matrix of mean dissimilarities (tough on another level of abstraction). These metrics are thus indicators on how many different strategies to counter related attacks are required (e.g., strategies to detect incidents in case of high variability of the time until discovery; or strategies on enforcing reporting in case of highly varying time until report etc.)

Further interpretations are straight forward and omitted here due to space limitations.

VI. Conclusion and Future Work

The approach proposed in this paper provides an important building block for a (national or sector-specific) cyber security center to establishing situational awareness. We presented a model that consists of relevant and important incident metrics (facts), proposed an algorithm to efficiently cluster incident data, outlined a prototype implementation with state of the art technologies and discussed its applicability. In detail, we highlighted some useful statistics, which are calculated on top of collected and clustered incident data, and their possible interpretation in a real cyber center in order to derive concrete measures on a higher level.

Our approach is a groundwork which identifies and evaluates the main elements comprising a systematic cyber situational awareness model based on incident clustering mechanisms. It is not intended to provide a comprehensive *ready-to-use* software solution for incident analysis at security operational centers, neither it targets at replacing existing analytical tools, it rather aims at improving the efficiency and support some of the tedious tasks of incident handling that are currently carried out manually by SOC operators. The development of our prototype is still ongoing and evaluation tests are currently being conducted to further validate our approach and extend the system with additional functionalities.

Future work is twofold: On the one side we have recently started to discuss our solutions with the national CERT and will further elaborate on application scenarios in the near future, specifically the exploitation of our work in the currently implemented national cyber security initiative. On the other side, there are some possible technical improvements in our software prototype. The most urgent ones are the flexible and smooth integration of new incident data on arrival without the need to recalculate the whole model, the extension of the basic incident model with more properties (i.e., facts), and the periodic calculation of clustering trends, i.e., the changes of cluster sizes, diameters etc. over time.

Acknowledgements

This work was partly funded by the Austrian FFG research program KIRAS in course of the project CIIS (840842) and the European Union FP7 project ECOSSIAN (607577).

References

[1] S. M. Rinaldi, "Modeling and simulating critical infrastructures and their interdependencies," in *System sciences*. IEEE, 2004, pp. 8–pp.

[2] C. Miller, "The legitimate vulnerability market: the secretive world of 0-day exploit sales," in *Workshop on the Economics of Information Security*, 2007, pp. 1–10.

[3] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.

[4] C. Tankard, "Advanced persistent threats and how to monitor and deter them." *Network Security*, vol. 2011, no. 8, pp. 16–19, 2011.

[5] NIST, "Framework for improving critical infrastructure cybersecurity," 2014-02-12.

[6] ITU-T, "Recommendation itu-t x.1500 cybersecurity info. exchange tech." 2012.

[7] ISO, "Iso/iec27010: Info. tech.: Security techniques - information security management for inter-sector and inter-organizational communications," 2012-03-20.

[8] J. L. Hernandez-Ardieta, J. E. Tapiador, and G. Suarez-Tangil, "Information sharing models for cooperative cyber defence," in *Cyber Conflict*, 2013, pp. 1–28.

[9] ENISA, "Good practice guide for incident management," European Union Agency for Network and Information Security, Tech. Rep., 2010.

[10] S. Jajodia, P. Liu, V. Swarup, and C. Wang, *Cyber Situational Awareness: Issues and Research*. Springer, 2009.

[11] K. Harrison and G. White, "Information sharing requirements and framework needed for community cyber incident detection and response," in *Homeland Security (HST), 2012 IEEE Conference on Technologies for*. IEEE, 2012, pp. 463–469.

[12] W. Zhao and G. White, "A collaborative information sharing framework for community cyber security," in *Homeland Security (HST)*. IEEE, 2012, pp. 457–462.

[13] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 1, pp. 32–64, 1995.

[14] M. Fracker, "Measures of situation awareness: Review and future directions," Wright-Patterson Air Force Base, Tech. Rep. AL-TR-1991-0128, 1991.

[15] N. Sarter and D. Woods, "Situation awareness: A critical but ill-defined phenomenon," *International Journal of Aviation Psychology*, vol. 1, pp. 45–57, 1991.

[16] European Union, "Eu cyber security strategy," http://eeas.europa.eu/top_stories/2013/070213_cybersecurity_en.htm, 2013.

[17] C. C. BRIDGES, "Hierarchical cluster analysis," *Psychological Reports*, vol. 18, pp. 851–854, 1966.

[18] J. MacQueen, in *Fifth Berkeley Symposium on Mathematical Statistics and Probability*.

[19] B. Everitt, *Cluster analysis*. London New York: Arnold Oxford University Press, 2001.

[20] M. Deza and E. Deza, *Encyclopedia of Distances*, 2009.