Supporting An ISMS Through Passive Network Inventory

Stephan Schwinger, Alexandra Meyer, Jeremy Perez, Arnold Krille
genua GmbH, Germany

Abstract

Modern networks keep growing in complexity and are rather dynamic by nature. On the other hand, due to legal requirements on information security, appropriate protective measures must be identified, implemented, sustained, enforced, and documented. To this end, network administrators are confronted with the effortful task of gaining an overview over their network, dividing the communicating devices into meaningful groups, and tracking changes. Hitherto existing research approaches usually suffer from a lack of readily available and used domain knowledge, fail to obtain acceptance of the derived device classes from the users or require either active network scans or agents running on managed devices. In our approach, this classification shall be guided by the pre-specified technical and infrastructural aspects of a methodology for information security management systems, namely the modules of the German IT Baseline Protection, and achieved by passive observation of the network traffic. This paves the way to a continuous control over the network.

1. Introduction

The German Federal Office for Information Security (BSI for Bundesamt für Sicherheit in der Informationstechnik) defines an information security management system (ISMS) as "a planned and organized course of action to achieve and maintain an appropriate level of information security" [1]. Its aim is both to prepare for critical situations such as cyberattacks as well as to guarantee compliance with legal requirements, as the violation of the latter may lead to loss of reputation or fines.

The IT Baseline Protection (ITBP) [2] is a general guideline for building an ISMS published by the BSI and defines an ISMS in accordance with the ISO27001 certification. It is divided into modules with a proper definition on when to apply them to a certain device or network segment. Besides, it raises awareness of the threats posed to these systems and defines requirements to counter them.

Usually, ISMSs enforce the definition, implementation, and fulfillment of both organizational and technical requirements. As an actual-theoretical comparison forms its initial step and networks are subject to constant changes, building and maintaining an ISMS is a time-consuming process.

The job of the administrator that we want to address in this contribution is the identification and labelling of systems running in his network. This includes in particular the task to determine components that are safety-critical, highly relevant for the business processes, or vulnerable. Wide-spread methods of device and software classification often require active scans or agents that are deployed on the respective devices. For instance, information on the used operation system may be retrieved using port scans, and type and version of the installed software may be obtained using banner grabbing. Documentation systems, that inventory present hardware and software or describe them in the course of a certification, generally have to be set up and maintained manually. Adaptions to the description of the systems usually do not occur automatically if their behavior changes. For that reason, the present documentation is outdated rather fast and does not represent the current state of the network. This has to be considered critical from both compliance and security concerns.

2. Related Work

Network segmentation, that is, its division into groups, may be driven by distinct characteristics, like roles within the networks or applications running on the devices. In [3] the authors attempt to divide the network according to two distinct scenarios, namely client-server-disambiguation and separation of infected from non-infected hosts. More complicated roles may both be predefined or computed from observations. [4] models the behavior of the hosts of a university network and shows reasonable results to recover the network segmentation by unsupervised learning and to assign new hosts to these segments using supervised methods. Similarly, [5] utilizes Random Forests to assign newly observed hosts to departments respectively the central server infrastructure within an intranet. [6] presents an approach that uses role vectors that have been hand-crafted by domain experts to both drive the clustering process and to assign new hosts to the computed clusters. In [7] a general approach to assign a role to each node within a network according to their topological properties is developed and evaluated on computer networks.

The usage of standard NetFlow features to derive a network segmentation or characterize host behavior can be found in a number of works. Often the former are enriched by specially tailored features such as entropy measures [8], detection of connection spikes and bursts [9] or aggregated features for split time

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

intervals [4]. In an IoT context, [10] extracts the MAC address of a device passively from ARP requests and searches a database for devices with a similar MAC address and known product type to determine the product type of this device. The passive scanner described in [11] exploits among other indicators URL patterns to determine the operating system of the network participants.

3. Labelling of Assets

Any automatized labelling of network devices must benefit the users. In the following, we want to shed some light on their requirements and justify our choice for satisfying the latter.

3.1. Motivation and Approach

From our experience, administrators lack a standardized labelling taxonomy. We identified a number of reasons for labelling devices:

i. I*nventory*: The user might use labels to keep an overview about the type and number of certain devices, like workstations or printers.

ii. *Filtering*: For reporting of traffic related information, on may want to select or aggregate in-formation for particular device types.

iii. *Firewalling*: Users may want to utilize a label within a firewall rule, for instance as a precondi-tion or an override for a certain flow action.

iv. *Localization*: The user may want to track the whereabouts of the device. This includes infor-mation on the building, facility, or business site as well as subnet-like divisions of the network.

In the following, we want to focus on the classification implied by the modules of the IT Baseline Protection. These modules can be divided into system-related modules and process-related ones. Examples for the former are *APP.3.1 Web Applications and Web Services*, *APP.3.2 Web Servers* and for the latter *ORP.3 Awareness and Training in Information Security*. The modules from the APP-, NET-, and SYS-categories are particularly suitable for the choice of labels, as they happen to overlap to a large extend with network observables.

Obviously, not all modules can be observed from the traffic, but as we will see, some may be derived implicitly.

3.2. Benefits

Grouping systems according to the taxonomy of the IT Baseline Protection developed by the BSI is well-suited for several reasons. Not only does the administrator get an overview about the types and properties of the systems that communicate in the network, but also valuable information may be provided that assures the compliance for instance in the context of an IT Baseline Protection or an ISO27001 certification of the IT landscape. A modern micro-segmentation of a network can be prepared and supported and thereby, the total security of the network can be increased by labelling devices with user-defined labels and using them to enforce highly adaptive and fine-grained security policies for the communication between these devices.

A uniform and at least partially automatized labelling procedure will boost the efficiency but requires domain knowledge not available to every organization in the required profundity. At this point, using expert systems, we want to alleviate the entry and daily routine of operating networks and securing them according to the requirements of the organization.

By recognizing changes in the number, functionality, and usage of devices through observation of changed behavior within the network, deviations from the documented state may be spotted. Thus, the administrator is enabled to keep the inventory up-to-date or to intervene by other actions, if an indicated change was not desired or points towards misbehavior, misconfiguration, or threats. Through the continuous usage in the daily operation, the documentation does not only cover the status in the moment of the certification best possible but is adjusted and scrutinized regularly. By this, efforts to generate or update this documentation on behalf of an audit for a certification, are reduced significantly. Furthermore, usability of the documentation for situational assessment and recovery of systems in the case of a security incident is supported.

By utilizing predefined and expert-validated labels to describe devices and their roles, the training period required to use them will be shortened and the lack of specialists may partially be compensated. Moreover, in the context of the IT Baseline Protection both the anticipated threats as well as the necessary and recommended measures for securing these assets and classes of assets are defined and updated regularly by the BSI.

4. Utilization of Domain Knowledge

In this section, we disclose the gap between observed user behavior and contemporary scientific approaches. Subsequently we sketch our proposed solution to overcome this gap.

4.1. User Behavior

We performed an initial experiment using a visualization of the network as a communication graph where network assets have been colored both according to their position within the network and the observed network protocols and applications, respect-

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

tively. The behavior of the target audience revealed the usage of the following techniques:

i. Graph based techniques: The user interpret-ed high nodal degrees as measure for central systems and services. His subsequent identification of systems was not necessarily correct. Besides, single-element colorings have been considered important as well, but interpretation was again a hard task.

ii. Subnet based techniques: Although not directly given by the user interface, the user tried to recover subnets assuming that they have been configured to contain systems with similar properties.

iii. Rule based techniques: Identical protocols used in the communication between different clusters were used as a trigger to mentally merge multiple source or target clusters.

iv. Protocol based techniques: The user at-tempted to find context to a certain cluster by filtering for dominant protocols, specific combination of protocols, or protocols that are typical for the use by a human.

4.2. State of the Art

Although the latter two techniques emphasize the high relevance of domain-specific information, recent scientific research focuses on rather abstract features derived from network data, like volumetric or timing information [12].

Jakalan [8] for instance captures, among other metrics, the mean packet size to distinguish signaling traffic from data exchange, or the number of communication partners to get an insight into the popularity of the IP node and to differentiate one-to-one from one-to-several and one-to-many communication.

In this sense, a certain intention can be provided for each extracted characteristic, but in general, a direct interpretation of the used features is hard to make accessible to the user.

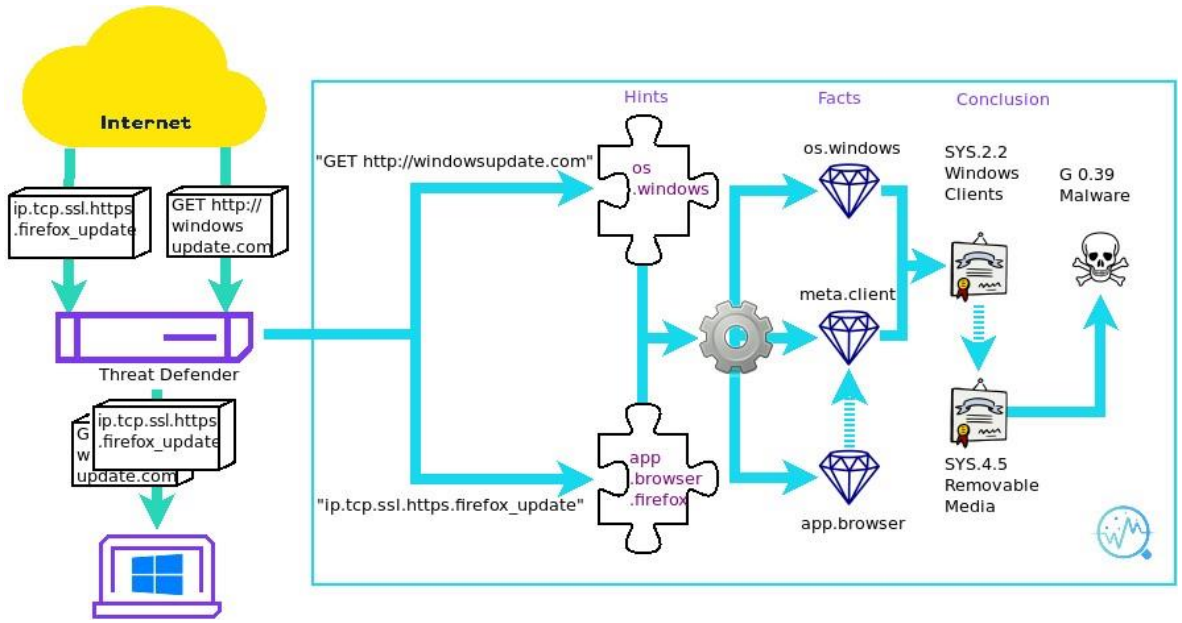

Figure 1. Steps to derive labels from network observations

4.3. Expert System Approach

Instead, we opt to interpret observations directly using domain knowledge. We assume that the network probe is deployed at an appropriate observation point, e.g. a central switch, and capable of extracting domain names from requests and classifying network traffic, e.g., by Deep Packet Inspection (DPI). In a first step we extract for each device in the internal network its possible properties by mapping observations to so-called hints. Note that the latter do not have to be accurate yet.

Every such mapping rule consists of a matching pattern, a client/server indication, and a condition indicating whether a reverse packet flow must be observed. The matching patterns apply to either FQDNs, MACs, or the application recognized by the DPI.

The hints are organized in a hierarchical manner such as *app.browser.firefox* and divided into the five top-level categories operating system (*os*), recognized installed software (*app*), vendor (*vendor*), device type (*device*) and meta information (*meta*).

From the collections of hints, for each device we derive facts through heuristics, that for example

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

consist of majority counts and combination of hints. Thereby, possible contradictions in the observations shall be resolved. Once the facts are extracted, they can be mapped or combined to the respective ITBP modules, which we regard as an explicit label for that device. Moreover, through the knowledge graph implicitly defined in the BSI Baseline Protection Compendium [2], we may assign implicit labels as well. In a final step, threats applicable to the device may be determined by evaluating predefined "cross tables".

A simplified illustration of this procedure is in order. Assume that, for a given device, we observed a flow classified as *ip.tcp.ssl.https.firefox* update and furthermore, a GET-request to *http://windowsupdate.com*. From the first observation we conclude a hint to an installed Firefox browser, this is, *app.browser.firefox*, while the second may indicate the device is running Windows, *os.windows*. After collecting a multitude of such hints, one may conclude the device is really a Windows-based machine having installed Firefox, which implies that it is most likely a client, and we additionally assign the fact *meta.client*. We can now derive the explicit ITBP-based label *SYS.2.2 Windows Clients*, and from that, the implicit label *SYS.4.5 Removable Media*. The latter entails the threat *G 0.39 Malware*. The procedure is depicted in Figure 1.

5. Clustering

ISMSs usually enforce compliance on groups of assets, not on single hosts. This motivates a clustering of the network, i.e., a microsegmentation that is not necessarily aligned with the subnet structure. A self-evident approach towards this would be a clustering of the communication graph itself, this is, the graph of assets that are linked if a communication occurred between them. However, according to [13] the latter graphs lack the assumptions underlying the methods from social network analysis and renders the application of the latter doubtful. To overcome this issue, Jakalan [14] constructs a "social behavior similarity graph" of internal network nodes by connecting them if and only if they communicated with the same external resources. Afterwards, communities of hosts are computed using this graph instead.

Independent of this, in the context of asset-clustering based on the derived facts, we conjectured that a graph formed from the observed network hosts exhibits the structure of a social graph if one would link assets that share a common fact after the above analysis. After initial experiments, we refined the idea and construct a weighted, undirected graph using Algorithm 1.

Algorithm 1 Construction of a social network graph

Input: a set of internal network nodes N, and the collected hints $H(n)$ for each of these nodes $n \in N$
Output: social network graph G

1: initialize an empty, undirected graph G
2: Add all internal network nodes $n \in N$ to the graph G
3: From $\{H(n), n \in N\}$, compute the set of facts for each node $F(n), n \in N$
4: For each subset $\{n, m\}$, $n \neq m$ of nodes, add an undirected edge $e(n, m)$ between them to G, weighted by the number of shared facts
$w(n, m) := |F(n) \cap F(m)|$, if $w(n, m) > 0$.
5: For each node n, add an undirected self-edge $e(n, n)$ weighted by the number of facts for this node, that are not shared with any other node,

$$w(n, n) := |\{ f \in F(n) : \quad F(n) \cap F(m) = \emptyset \quad \forall\, m \in N \setminus \{n\}\}|,$$

if $w(n, n) > 0$.

The community detection can then be performed on this graph using the Louvain method [15] or Stochastic Block Models (SBMs) [16], both of which have been used in the cyber security domain before [17].

Step 5 in algorithm 1 is motivated by the elsewise experimentally observed but unfavorable assignment of nodes exposing unique facts to communities that fail to share these facts. It also mirrors the self-loops linked to super-nodes that are produced in phase 2 of the Louvain algorithm.

After the host communities are computed, the asset-wise concluded ITBP modules may be assigned to them. On the one hand, this leads to a generalization of said conclusions in the sense that the procedure may add candidates of missing labels for certain assets. On the other hand, they serve as an inherent cluster explanation rendering our methodology an XAI approach. This explanation can be traced back to the underlying hints, providing the administrator the

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

Table 1. Derived Segmentation of CIC-IDS2017-Monday using Louvain Method

cluster id	cluster members	ITBP modules	
1	DNS/DC Server (Win Server 2016)	APP.2.1 General Directory Service APP.2.2 Active Directory APP.3.6 DNS Servers APP.5.3 General E-Mail Clients and Servers APP.6 General Software SYS.1.2 Windows Server SYS.2.1 General Client SYS.2.2 Windows Clients SYS.4.5 Removable Media	CON.3 Backup Concept NET.2.2 WLAN Usage NET.3.3 VPN OPS.1.1.2 Proper IT Administration OPS.1.1.3 Patch and Change Management OPS.1.1.4 Protection Against Malware OPS.1.1.5 Logging OPS.2.2 Cloud Usage ORP.4 Identity and Access Management
2	MAC Ubuntu 16.4 64B Ubuntu 14.4 32B Ubuntu 14.4 64B Ubuntu server 12 Public Win Vista 64B	APP.1.2 Web Browsers APP.5.3 General E-Mail Clients and Servers APP.6 General Software SYS.2.1 General Client SYS.2.3 Linux and Unix Clients SYS.4.5 Removable Media	CON.3 Backup Concept NET.2.2 WLAN Usage NET.3.3 VPN OPS.2.2 Cloud Usage OPS.1.1.4 Protection Against Malware
3	Web server 16 Public	APP.3.3 File Servers	CON.3 Backup Concept OPS.1.2.5 Remote Maintenance ORP.4 Identity and Access Management
4	Ubuntu 16.4 32B Win 10 64B Win 10 pro 32B Win 7 Pro 64B Win 8.1 64B	APP.1.2 Web Browsers APP.5.3 General E-Mail Clients and Servers APP.6 General Software SYS.2.1 General Client SYS.2.2 Windows Clients SYS.4.5 Removable Media	CON.3 Backup Concept NET.2.2 WLAN Usage NET.3.3 VPN OPS.2.2 Cloud Usage OPS.1.1.4 Protection Against Malware

possibility for feedback to fine-tune the model.

6. Experiments

Next, we want to evaluate the applicability of our method in realistic scenarios.

6.1. Support of an ISMS

In this section, we want to examine the opportunities of our approach within an ISMS context. We consider the Intrusion Detection Evaluation Dataset CIC-IDS2017 [18] provided by the University of New Brunswick. It was generated in a testbed and is provided in the form of PCAP files. It particularly suits our requirements, as the hosts of the victim network are well-described. We restrict ourselves to data from Monday, July 3, 2017, as it is free of malicious activity.

A drawback of this choice is the rather limited number of assets and the short observation time, which for instance leads to absent hints on the OS for a number of devices. Furthermore, our probe did not encounter any hint that the Web server acts as such in this time period.

As claimed in section 5, the approach has XAI properties and thus the potential to respect user feedback. We take advantage of this by dropping a hint for mobile applications, which would otherwise lead to undesired ITBP labels.

We run our method using Louvain clustering on the social network graph constructed by Algorithm 1. Louvain clustering locally optimizes modularity, in that most of the edges for nodes of a certain community connect to other nodes of the same community. The computed clusters and implied BSI modules are listed in Table 1. Clearly, web server and domain controller are well separated from the remaining machines.

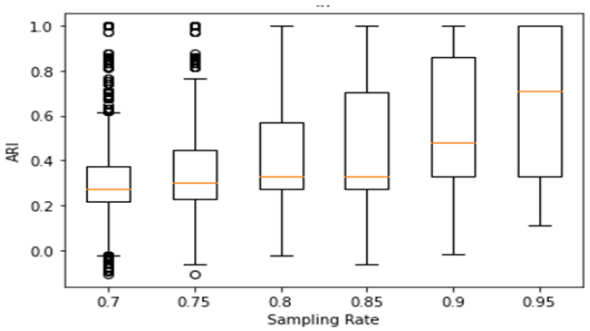

Figure 2. Cluster reproduction when partially hiding Observations

The apparent mixing of operating systems in cluster 2 can be explained by the absence of hints for the expected OSs. With other words, the Linux detection has been generalized to otherwise similar machines. Cluster 4 basically forms the cluster of Windows clients. The misassignment of the 'Ubuntu 16.4 32B'-machine to this cluster is due to the fact that the DPI reports an access to Windows Marketplace.

6.2. Stability Considerations

The output of phase one of the Louvain method depends on order in which nodes are considered. We

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

therefore rerun the method with different seeds for the random number generator. In 76% of the runs, we recovered exactly the decomposition shown in Table 1. Several decompositions consisting of 3 clusters have been found in 24% of the runs. In rare occasions, the graph was split into two parts only.

The Leiden algorithm [19] is intended to overcome certain defects of the Louvain method. For our data, it always correctly separates both 'DNS/DC Server Win Server 2016' and 'Web server 16 Public' from the remaining graph but is unable to decompose the latter any further.

We further want to get an insight into the dependency of the method's accuracy on the number of observations. With the notation of Algorithm 1, we take the set of observations for CIC-IDS2017 used in section 6.1

$$\{(n, h) : n \in N, h \in H(n)\}$$

as ground truth. We randomly restrict to a fraction

$$p \in \{0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$$

of these observations, construct the social network graph from the remaining hints, and run Louvain clustering. Assets lacking any hints after dropping observations were put in respective single-element communities. We then compute the Adjusted Rand Index [20] as a similarity measure between the obtained clustering and the one given in Table 1. We repeat this procedure 10,000 times for each p with a fixed seed for the random generator. The results of this experiment are depicted in Figure 2. Note that in extreme cases, even for $p = 0.7$ the exact clustering may be recovered. While both median and mean of the cluster similarity rise when p is increased, we also observe an increase in variance.

6.3. Detection Rate

The stability considerations of the previous subsection raise questions regarding hint distribution and asset visibility. The number of hints per asset computed for the CIC-IDS2017 dataset before erasure of undesired hints is bounded from below by two. To get more insight into the potential of our method, we took measurements in two real computer networks. Unfortunately, the data cannot be published due to privacy concerns. The first network was observed over the period of about 18 months. We tracked the number of newly discovered network hosts as well as the number of network hosts that could be assigned at least one hint. The observed temporal developments are depicted in Figure 3. We recognize a steadily growing number of discovered network hosts.

The share of assets, for which the currently implemented expert system could find a hint, is comparatively small.

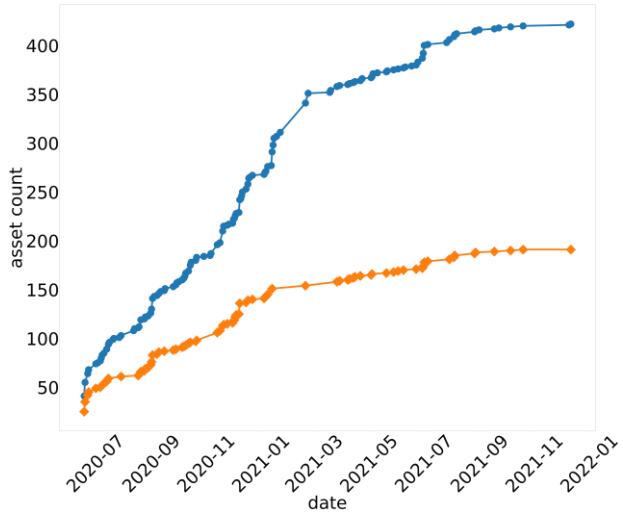

Figure 3. Temporal development shows the number of detected devices (blue) versus devices that could be assigned at least one hint (orange)

The final quota of describable assets to detected assets reads 0.45. As we dealt with a rather non-prototypical network here, we assume this number to be in the lower range for the percentage of assets provided with hints. Indeed, in the second real network, this quota was significantly higher, namely 0.95.

7. Discussion and Future Work

The presented procedure extends classical DPI as we observe traffic (this is, edge) properties to describe nodes instead of using them to describe communication relations. Let us briefly review our progress and sketch ongoing and future research directions.

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

7.1. Methodology

In the current expansion stage our approach has to be regarded as a demonstrator. Two contributions to support administrators in understanding their network have been established but are subject to further research.

The first contribution derived properties of the communicating network assets from network traffic observations. To ensure practicability, the utilized heuristics have to be enhanced, continuously maintained, and tested for effectiveness and consistency on multiple distinct networks. To lower the number of subject matter experts required to facilitate this task, we want to replace the heuristic mappings from the hints to the facts by weak labelling approaches [21] in future versions of our algorithm. First experiments in that direction have been encouraging: Instead of sources for a hint, we interpreted each of the mapping rules described in section 4.3 as a "labelling function" for the respective fact. Using thresholding by the elbow method, a number of hints could automatically be confirmed to be facts for the CIC-IDS2017 dataset. To successfully apply weak supervision, we need to correctly treat both conflicting (such as operating systems) and non-exclusive facts (like the type of installed browsers). Also, the encoding of a priori confidence into the labelling function might improve results. Given that not all assets might expose hints to derive their properties, an opportunity to still apply weak supervision approaches is to extend derived facts to unlabeled machines by multi-modal models.

As we have seen, through administrator feedback one may take advantage of interpretability of the heuristic labels to improve results by neglecting hints. This is still true, yet more involved when using weak supervision by an interactive refinement of the labelling functions.

In the second contribution provided in this paper, we only scratched the surface of community detection methodology. While the Louvain method showed promising results, more sophisticated methods for the analysis of social networks are available that go beyond modularity optimization. The Louvain algorithm also extracts non-overlapping communities only. This is a questionable property for real-world applications.

7.2. Continuous Compliance

A permanent surveillance and enforcement of security policies is desirable and subject of active research [22]. This work contributes to the vision of "continuous compliance" by proposing means to keep the network documentation constantly up-to-date. Moreover, an "intelligent switch" is capable of labelling devices with user-defined tags and, using the latter, of enforcing security policies for the communication between these devices. A possible feedback channel from our analyzer to this policy engine opens the opportunity to update the computed device classification dynamically. This mechanism paves the way for a dynamic, highly adaptive, and fine-grained micro-segmentation without agents to be installed on the particular hosts. A research question of high practical relevance is the interplay between business processes and compliance rules. Currently, our approach is ignorant of the former.

7.3. Secondary Concerns

Nevertheless, the examples of section 1 demonstrate that an initial, yet not fully accurate network role recognition is possible. As network security must be considered not a state, but a process, the first implementation of an ISMS does not have to be perfect yet and is subject to continuous progress and improvement. More important, the results emphasize the necessity of feedback and thus once more the crucial role in the cybersecurity process taken by humans, as already elaborated in [23]. To explain the result of the asset labelling to the user and refine the derivation of facts and ITBP modules to be applied using his feedback, an advanced presentation layer has to be created using modern UX principles.

From a legal perspective, conformity to the GDPR must be ensured. As in the presented approach no connection to user identity is established, no temporal information is stored and the classification-IDs may be filed in a pseudonymized way, we do not expect major obstacles here.

8. Conclusion

The modules from the BSI IT Baseline Protection are a valid choice for labelling network devices. We proposed a domain-knowledge-based procedure that computes such labels by passive observation of the traffic. Moreover, the clustering of the induced social network graphs assists the structural analysis of the IT environment in the course of security audits. Therefore, this approach supports administrators to gain permanent control over their network.

9. References

[1] Guide to Basic Protection based on IT-Grundschutz: 3 Steps to Information Security (Tech. Rep.). (2017). https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Grundschutz/International/Basic_Security.pdf (Access Date: 28 August 2023).

[2] Schildt, H., Alberts, K., Förster, S., Hoffmann, B., and und Jessika Welticke, F. N. (Eds.). (2022). IT-Grundschutz-Kompendium. Köln: Reguvis. https://www.bsi.bund.de/SharedDocs/ Downloads/DE/BSI/Grundschutz/Kompendium/IT_Grundschutz_Kompendium_Edition2022.pdf (Access Date: 07 June 2023).

 | DOI: 10.20533/ijcc.2046.3359.2023.0028

[3] Ring, M., Dallmann, A., Landes, D., and Hotho, A. (2017). IP2Vec: Learning similarities between IP addresses. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (p. 657–666). DOI: 10.1109/ICDMW.2017.93.

[4] Smeriga, J., and Jirsik, T. (2019, 8). Behavior-Aware Network Segmentation using IP flows. In ARES '19 (p. 1–9). Canterbury, UK. DOI: 10.1145/3339252.3339265.

[5] Wang, P., Zhou, Y., Zhu, C., and Yue, R. (2018, 03). Role classification with netflow data in intranet. In 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI) (p. 279–282). DOI: 10.1109/ICACI.2 018.8377620.

[6] Huffer, K. M. T., and Reed, J. W. (2017). Situational Awareness of Network System Roles (SANSR). In Cyber and Information Security Research (pp. 8:1–8:4). New York, NY, USA: ACM. DOI: 10.1145/ 3064814.3064828.

[7] Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., . . . Li, L. (2012). RolX: Structural role extraction and mining in large graphs. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1231–1239). DOI: 10.1145/2339530.2339723.

[8] Jakalan, A., Gong, J., Weiwei, Z., and Su, Q. (2015, 03). Clustering and Profiling IP Hosts Based on Traffic Behavior. Journal of Networks, 10(2), 99–107. DOI: 10.4304/jnw.10.2.99-107.

[9] Yen, T.-F., Oprea, A., Onarlioglu, K., Leetham, T., Robertson, W., Juels, A., and Kirda, E. (2013). Beehive: Large-scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks. In Proceedings of the 29th Annual Computer Security Applications Conference (pp. 199–208). New York, NY, USA: ACM. DOI: 101145/2523649.2523670.

[10] Niedermaier, M., Hanka, T., Plaga, S., von Bodisco, A., and Merli, D. (2018, 08). Efficient Passive ICS device discovery and identification by MAC address correlation. In 5th International Symposium for ICS and SCADA Cyber Security Research 2018 (ICS-CSR 2018). DOI: 10.14236/ewic/ICS2018.3.

[11] Lastovicka, M., Jirsik, T., Celeda, P., Spacek, S., and Filakovsky, D. (2018). Passive OS fingerprinting methods in the jungle of wireless networks. In NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium (p. 1–9). DOI: 10.1109/ NOMS.2018.8406262.

[12] Moore, A., Zuev, D., and Crogan, M. (2005, 08). Discriminators for use in flow-based classification (Tech. Rep. No. RR-05-13). Queen Mary, University of London.

[13] Rubin-Delanchy, P., Adams, N. M., and Heard, N. A. (2016). Disassortativity of computer networks. In 2016 IEEE Conference on Intelligence and Security Informatics (ISI) (pp. 243–247). DOI: 10.1109/ISI.2016.7745482.

[14] Jakalan, A., Gong, J., Su, Q., and Hu, X. (2015). Community Detection in large-scale IP networks by Observing Traffic at Network Boundary. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1)), 59–64.

[15] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008, 10). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008. DOI: 10.1088/1742-5468/2008/10/P10008.

[16] Peixoto, T. P. (2014, 03). Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. Phys. Rev. X, 4, 011047. DOI: 10.1103/PhysRevX.4.011047.

[17] Beukema, W. J. B., Attema, T., and Schotanus, H. A. (2017). Internal Network Monitoring and Anomaly Detection through Host Clustering. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017) (p. 694–703). SciTePress. DOI: 10.5220/0006288606940703.

[18] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018, 01). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In ICISSP (pp. 108–116). DOI: 10.5220/ 0006639801080116.

[19] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," Scientific Reports, vol. 9, no. 1, p. 5233, Mar 2019. DOI: 10.1038/s41598-019-41695-z.

[20] L. J. Hubert and P. Arabie, "Comparing partitions," Journal of Classification, vol. 2, pp. 193–218, 1985. DOI: 10.1007/BF01908075.

[21] Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017, 11). Snorkel: Rapid training data creation with weak supervision. Proc. VLDB Endow., 11(3), 269282. DOI: 10.14778/ 3157794.3157797.

[22] Lorenz, C., Clemens, V., Schrötter, M., and Schnor, B. (2022). Continuous Verification of Network Security Compliance. IEEE Transactions on Network and Service Management, 19(2), 1729–1745. DOI: 10.1109/TNSM.2021.3130290.

[23] Krille, fA., Perez, J., and Schwinger, S. (2021, 02). Kein Deus ex Machina: Warum Mensch und Maschine gemeinsam Cybersecurity machen müssen. In Deutschland. Digital. Sicher. 30 Jahre BSI (p. 367–381). SecuMedia Verlag.

[24] Kögel, J., Dietz, K., Mühlhauser, M., Seufert, M., Gray, N., Hoßfeld, T., . . . Arzig, C. (2023). Website of the Wintermute Project. https://www.projekt-wintermute.de (Access Date: 20 August 2023).

Acknowledgements

This work was partly funded as part of the Wintermute project [24] by the German Federal Ministry of Education and Research (BMBF) under contract No. 16KIS1126K. The consortium consisted of genua GmbH, IsarNet Software Solutions GmbH, acs plus

GmbH and the universities of Bamberg, Bremen and Würzburg. Responsibility for the information and views expressed in this publication lie entirely with the authors. We want to express our gratitude to our research partner XITASO GmbH.

 | DOI: 10.20533/ijcc.2046.3359.2023.0028