J. E. Rubio, C. Alcaraz, R. Rios, R. Roman, and J. Lopez, “Distributed Detection of APTs: Consensus vs. Clustering”, *25th European Symposium on Research in Computer Security (ESORICS 2020)* vol. 12308, pp. 174-192, 2020.
http://doi.org/https://doi.org/10.1007/978-3-030-58951-6_9
NICS Lab. Publications: https://www.nics.uma.es/publications

Distributed Detection of APTs: Consensus vs. Clustering

Juan E. Rubio, Cristina Alcaraz, Ruben Rios
Rodrigo Roman, Javier Lopez
Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain
{rubio,alcaraz,ruben,roman,jlm}@lcc.uma.es

Abstract

Advanced persistent threats (APTs) demand for sophisticated traceability solutions capable of providing deep insight into the movements of the attacker through the victim's network at all times. However, traditional intrusion detection systems (IDSs) cannot attain this level of sophistication and more advanced solutions are necessary to cope with these threats. A promising approach in this regard is Opinion Dynamics, which has proven to work effectively both theoretically and in realistic scenarios. On this basis, we revisit this consensus-based approach in an attempt to generalize a detection framework for the traceability of APTs under a realistic attacker model. Once the framework is defined, we use it to develop a distributed detection technique based on clustering, which contrasts with the consensus technique applied by Opinion Dynamics and interestingly returns comparable results.

Keywords: clustering, consensus, opinion dynamics, distributed detection, traceability, advanced persistent threat

1 Introduction

In recent years, there has been a growing interest for advanced event management systems in the industrial cyber-security community for two main reasons: (i) the integration of cutting-edge technologies (e.g., Big Data, Internet of Things) into traditionally isolated environments, which adds complexity to data collection and processing [1]; and (ii) the emergence of the new attack vectors as a result of the Industry 4.0 evolution, which have not been properly studied in context and may form part of an Advanced Persistent Threat (APT) [2].

APTs consist of sophisticated attacks perpetrated by resourceful adversaries which cost millions every year to diverse industrial sectors [3]. The main concern with these threats is that they are especially difficult to detect and trace. In this context, traditional Intrusion Detection Systems (IDS) only pose a first line of defense in an attempt to identify anomalous behaviours in very precise points of the infrastructure [4], and they are tailored to specific types of communication standards or types of data, which is not sufficient to track the wide range of attack vectors that might be used by an APT.

It is then necessary to fill this gap between classic security mechanisms and APTs. The premise is to find proper mechanisms capable of monitoring all the devices (whether physical or logical) that are interconnected within the organization, retrieve data about the production chain at all levels (e.g., alarms, network logs, raw traffic) and correlate events to trace the attack stages throughout its entire life-cycle. These measures would provide the ability to holistically detect and anticipate attacks as well as failures in a timely and autonomous way, so as to deter the attack propagation and minimize its impact.

To cope with this cyber-security scenario, novel candidate solutions such as the Opinion Dynamics approach emerge [5]. These alternatives propose to apply advanced correlation algorithms that analyze an industrial network from a holistic point of view, leveraging data mining and machine learning mechanisms in a distributed fashion. In this paper, we formalize a framework that enables the design and practical integration of such distributed mechanisms for the traceability of APTs, while also comparing the features of the aforementioned solutions according to the cyber-security needs of the industry nowadays, both qualitatively and experimentally. Altogether, we can summarize our contributions as:

- Characterization of the context in terms of security requirements and available solutions;
- Definition of a framework for developing solutions that enable the distributed correlation of APT events, based on these security needs and a new attacker model;
- Identification of effective techniques and algorithms for the traceability of APTs that satisfy the proposed framework;
- Qualitative and quantitative comparison of approaches in an Industry 4.0 scenario.

The remainder of the paper is organized as follows. Section 2 presents the state of the art of intrusion detection and anomaly correlation mechanisms, as well as the preliminary concepts involved in the studio. Then, Section 3 presents the security and detection requirements, whereas Section 4 defines the framework for developing solutions that fulfill them. Based on such framework, the studied solutions are addressed in Section 5, and experimentally analysed in Section 6. Finally, extracted conclusions are discussed in Section 7.

2 Background and preliminaries

At present, there is a plethora of intrusion detection approaches tailored for traditional industrial scenarios (cf. [6]) and Industry 4.0 networks (cf. [7]).

This includes specification-based IDSs, which compare the current state of the network with a model that describes its legitimate behaviour [8]; physics-based modeling systems, which simulates the effect of commands over the physical dynamics of the operations [9]; and other more traditional strategies such as signature and anomaly detection systems. Most of these detection approaches focus on the analysis of certain aspects of industrial control systems, such as the communication patterns, the behaviour of sensors and actuators, and others.

Still, industrial technologies are becoming more heterogeneous and attacks are extremely localized, which makes crucial to monitor all elements and evidences. Therefore, it is important for industrial ecosystems to set up more than one detection solution to ensure the maximum detection coverage [10]. Moreover, all solutions should coexist with advanced detection platforms that take the infrastructure from a holistic perspective, correlate all events and track all threats throughout their entire life-cycle [11]. This holistic perspective is even more necessary in light of the existence of APTs: sophisticated attacks comprised of several complex phases – from network infiltration and propagation to exfiltration and/or service disruption [3][12].

These advanced detection platforms have been explored in traditional Information Technology (IT) environments through forensic investigation solutions, using proactive (that analyse evidences as incidents occur) or reactive techniques (where evidences are processed once the events occur). Examples of these include flow-based analysis of traffic in real time [13] or the correlation of multiple IDS outputs to highlight and predict the movements of APTs, using information flow tracking [14] or machine learning [15]. Still, most of them are limited to a restricted set of attacks and are not applied to a real setup.

In turn, the progress in the Industry 4.0 has not been significant with respect to actual APT traceability solutions. In this sense, the Opinion Dynamics approach [5] paves the way for a new generation of solutions based on the deployment of distributed detection agents across the network. The anomalies reported by these agents are correlated to extract conclusions about the sequence of actions performed by the adversary, and also to identify the more affected areas of the infrastructure. Such assessment can be conducted in a centralized entity or using a distributed architecture of peers [16]. At the same time, it is open to integrate external IDS to examine anomalies in the vicinity of nodes, as well as the abstraction of diverse parameters such as the criticality of resources or the persistence of attacks.

Despite the many capabilities of this solution (explained in Section 5.1), it is necessary to define a more general detection model to lay the base for the precise application of more APT traceability solutions in the Industry 4.0 paradigm. The reason is that the Opinion Dynamics capabilities can be implemented modularly, they can be integrated into other correlation algorithms and each one has a different effect on many security, detection, deployment and efficiency constraints. These points will be addressed in the next section, where we define the security and detection requirements involved, to latter present the traceability framework.

3 Security and detection requirements

Based on the state of the art presented in Section 2, this section enumerates the requirements for the development of advanced solutions and systems that provide a holistic perspective on industrial ecosystems. According to [7], we should consider the following detection requirements:

(D1) **Coverage.** APTs make use of an extensive set of attack vectors that jeopardize organizations at all levels. Therefore, the system must be able to assimilate traffic and data from heterogeneous devices and sections of the network, while also incorporating the input of external detection systems.

(D2) **Holism.** In order to identify anomalous behaviors, the system must be able to process all the interactions between users, processes and outputs generated, as well as logs. This allows to generate anomaly and traceability reports at multiple levels (e.g., per application, device or portion of the network, as well as global health indicators).

(D3) **Intelligence.** Beyond merely detecting anomalous events within the network in a timely manner, the system must infer knowledge by correlating current events with past stages and anticipate future movements of the attacker. Similarly, it should provide mechanisms to integrate information from external sources – that is, cyber threat intelligence [17].

(D4) **Symbiosis.** The system should have the capability to offer its detection feedback to other Industry 4.0 services, by means of well-defined interfaces. This includes access control mechanisms (to adapt the authorization policies depending on the security state of the resources) or virtualization services (that permit to simulate response techniques under different scenarios without interfering the real setup), among others.

On the other hand, we can also establish the following security requirements with regards to the deployment of the detection solution over the network:

(S1) **Distributed data recollection.** It is necessary to find distributed mechanisms – such as local agents collaborating in a peer-to-peer fashion – that allow the collection and analysis of information as close as possible to field devices. The ultimate aim is to make the detection system completely autonomous and resistant to targeted attacks.

(S2) **Immutability.** The devised solution must be resistant to modifications of the detection data at all levels, including the reliability and veracity of data exchanged between agents (e.g., through trust levels that weigh the received security information), and the storage of such data (e.g., through unalterable storage mediums and data replication mechanisms such as immutable databases or distributed ledgers).

(S3) **Data confidentiality.** Apart from the protection against data modification, it is mandatory that the system provides authorization and cryptographic mechanisms to control the access to the information generated by the detection platform and all the interactions monitored.

(S4) **Survivability.** Not only the system must properly function even with the presence of accidental or deliberate faults in the industrial infrastructure, but also the system itself cannot be used as a point of attack. To achieve this, the detection mechanisms must be deployed in a separated network that can only retrieve information from the industrial infrastructure.

(S5) **Real-time performance.** The system must not introduce operational delays on the industrial infrastructure, and its algorithms should not impose a high complexity to ensure the generation of real-time detection information. Network segmentation procedures and separate computation nodes (e.g., Fog/Edge Computing nodes) can be used for this purpose.

4 APT traceability framework for the Industry 4.0

After defining the detection and security requirements that a conceptual APT traceability solution must fulfill, we now describe the guidelines for the design and construction of its deployment architecture, the algorithms to be used, and the attacker model under consideration.

4.1 Network architecture and information acquisition

The industrial network topology is modelled with an acyclic graph $G(V, E)$, where V represents the devices and E is the set of communication links between them. This way, V can be assigned with parameters to represent, for instance, their criticality, vulnerability level or the degree of infection; whereas the elements in E can be associated with Quality of Service (QoS) parameters (e.g., bandwidth, delays), or compromise states that help to prioritize certain paths when running resilient routing algorithms.

For the interest of theoretical analysis, these networks are frequently generated using random distributions that model the architecture of real industrial systems. Also, the topology can be subdivided into multiple network segments with different distributions [5]. This is useful, for example, to study the effects of the attack and detection mechanisms over the corporate section (containing IT elements) and the operational section (OT, containing pure industrial assets), which can be connected by firewalls, so that $V = V_{IT} \cup V_{OT} \cup V_{FW}$.

Regardless of the topology configuration, the detection approach must acquire information from the whole set of nodes V to fulfill requirement D1 (Coverage, c.f. Section 3), by using agents that are in charge of monitoring such devices, complying with S1. Each of these agents can be either mapped to a individual device (following a 1:1 relationship), which would be ideal for S1, or aggregate the data from a set of physical devices belonging to the infrastructure. In either case, we can assume they are able to retrieve as much data as possible from their assigned devices, which encompasses **network-related parameters** (e.g., links with other nodes, number packets exchanged, delays), **host-based data** (e.g., storage, computational usage) or **communication information** (e.g., low-level commands issued by supervision protocols). These data items are aimed to feed the correlation algorithm with inputs in the form of an anomaly value for every device audited, which is formalized by vector x. This way, x_i represents the anomaly value sensed by the corresponding agent on device i, for all $i \in 1, 2, ..., |V|$. Such value can be calculated by each agent autonomously (e.g., applying some machine learning to determine deviations in every data item analyzed) or leveraging an external IDS that is configured to retrieve the raw data as input, thereby conforming to requirement D3.

From a deployment perspective, this leads to the question of where to locate the computation of anomalies and their subsequent correlation. As for the former, agents are implemented logically, since it is not always feasible to physically integrate monitoring devices into the industrial assets due to computational limitations. Consequently, these processes may have to run in separate computational nodes. However, we still want to achieve a close connection to field devices while avoiding a centralized implementation such as the one presented in [5]. The solution is then to introduce an intermediate approach based on the concept of *distributed data brokers*. These components collect the data from a set of individual devices via port-mirroring or network tapping, using data diodes to decouple agents from actual systems and ensure that data transmission is restricted to one direction, thereby shielding the industrial assets from outside access and complying with requirements S3 and S4.

These data brokers can also convey the detection reports (i.e., the anomalies sensed by its logical agents over the area where it is deployed) to other brokers in order to execute the correlation in a collaborative way. In consequence, they must be strategically deployed in a separate network such that there is at least one path between every two brokers.

Due to this distributed nature, the correlation algorithm can make use of two data models: *Replicated database*, which assumes that every agent has complete information of the whole network (e.g., through distributed ledgers), and *Distributed data endpoints*, where the information is fully compartmentalized and the cross-correlation is conducted at a local level. Both approaches have their advantages and disadvantages. The replicated database provides all agents with a vision of the network, although it imposes some overhead. As for the distributed data endpoints, they reduce the number of messages exchanged, yet the algorithm must deal with partial information coming from neighbour brokers.

Altogether, the ultimate election of the algorithm, data model and architectural design of the agents responds to performance and overhead restrictions. This composes the detection mechanism at a physical layer, while at an abstract level, it must also return a set of security insights that are based on the attacker model, described in the following.

4.2 Attacker model

To analyze the parametrization of traceability algorithms, we need to characterize the chain of actions performed by an APT over the network. We will use the formalization described in [5] as a starting point. That paper reviews the most relevant APTs reported in the last decade and extracts a standard representation of an APT in the form of a finite succession of attacks stages: *initial intrusion*, *node compromise*, *lateral movement*, and *data exfiltration or destruction*. These stages cause different anomalies that are potentially inspected by the detection agents, according to an ordered set of probabilities [5].

Using these stages, APTs can be represented as a finite sequence of precise events. However, the original authors did not consider the possibility of parallel APT traces being executed simultaneously across the network, hence generating cross-related events. We refer to them as *concurrent routes* followed by the APT in the attack chain or different APTs taking place (that may eventually collaborate). We extend the aforementioned attacker model with this novel feature and formalize it in Algorithm 1.

In the algorithm, the effect of multiple APTs are implemented as a succession of updates on vector x, both in the concerned node (i.e., the anomaly sensed by its agent) and its neighbours when certain attack stages (e.g., a *compromise* stage) are involved. The routine continues until all attack stages have been executed for the entire set of APTs considered. On the other hand, the *theta* values refer to the ordered set of probabilities presented in [5]: the lower value the index of theta, the higher anomaly is reported by the agent after that phase. That paper also illustrates the attenuation function, which decreases the values of vector x over time based on the persistence of attacks, the criticality of the resources affected and their influence over posterior APT phases.

During the execution of these iterations, the correlation algorithm can be executed at any time to gain knowledge of the actual APT movements, by using the anomalies as input. The complete explanation of inputs and outputs for the traceability solution is further explained in next subsection.

4.3 Inputs and outputs of the traceability solution

After introducing how the information from physical devices is collected by agents in practice and how the anomalies can be calculated in theoretical terms for our simulations, we summarize the set of inputs for traceability solutions as:

(I1) **Quantitative input.** expressed with vector x to assign every industrial asset with an anomaly value prior to conducting the correlation. As previously mentioned, it can be calculated by each associated agent or using external detection mechanisms integrated with the data broker by taking an extensive set of data inputs to comply with D1 and D2. In our simulations, this value is given by the attack phases executed on the network in a probabilistic way, without the detection mechanism having any knowledge about the actual stages.

Algorithm 1 Attacker model - anomaly calculation

```
input: $attackSet_k$, representing the chain of actions of APT $k, 1 \leq k \leq numOfApts$
local: Graph $G(V, E)$ representing the network
output: $x_i$ representing the anomaly value sensed by each agent $i$ at the end of the APT
network, where $x_i \in (0, 1)$

$x \leftarrow zeros(|V|)$ (initial opinion vector)
while $attackSet_k \neq \oslash \forall k \in \{1, .., numOfApts\}$ do
    for $k \leftarrow 1$ to $numOfApts$ do
        if $|attackSet_k| > 0$ then
            $\{attack \leftarrow next\ attack\ from\ attackSet_k\}$
            if $attack == initialIntrusion_(IT, OT, FW)$ then
                $attackedNode_k \leftarrow random\ v \in V_{(IT,OT,FW)}$
                $x(attackedNode_k) \leftarrow x_{attackedNode_k} + \theta_3$
            else if $attack == compromise$ then
                $x_{attackedNode_k} \leftarrow x_{attackedNode_k} + \theta_2$
                for $neighbour$ in $neighbours(attackedNode_k)$ do
                    $x_{attackedNode_k} \leftarrow x_{attackedNode_k} + \theta_5$
                end for
            else if $type(attack) == LateralMovement$ then
                $previousAttackedNode \leftarrow attackedNode_k$
                $attackedNode_k \leftarrow \textsc{SelectNextNode}(G, attackedNode_k)$
                $x_{previousAttackedNode_k} \leftarrow x_{previousAttackedNode_k} + \theta_5$
                $x_{attackedNode_k} \leftarrow x_{attackedNode_k} + \theta_{3,4}$
            else if $attack == exfiltration$ then
                $x_{attackedNode_k} \leftarrow x_{attackedNode_k} + \theta_4$
            else if $attack == destruction$ then
                $x_{attackedNode_k} \leftarrow x_{attackedNode_k} + \theta_1$
            end if

            $x \leftarrow \textsc{AttentauteOldAnomalies}(x)$
            $attackSet_k \leftarrow attackSet_k \setminus attack$
        end if
    end for
end while
```

(I2) **Qualitative input.** the previous values need to be enriched with information to correlate events in nearby devices and infer the presence of related attack stages, according to Section 4.2. At the same time, we also need to prioritize attacks that report a higher anomaly values. We assume that the resulting knowledge can be reflected in form of a weight w_{ij}, which is assigned by every agent i to each of its neighbours and represents the level of trust given to their anomaly indications when performing the correlation (fulfilling S2). This parameter can be subject to a threshold ε, which defines when two events should be correlated depending on the similarity of their anomalies. Further criteria could be introduced to associate anomalies from different agents.

With respect to the outputs of the traceability solutions, they should include, but are not limited to the following items:

(O1) Local result to determine whether the agent is generating an anomaly due to whether the actual infection of the associated node, as a result of a security threat in a neighbour device or a false positive.

(O2) Information at global level, to determine the degree of affection in the network and the nodes that have been previously taken over, filtered by

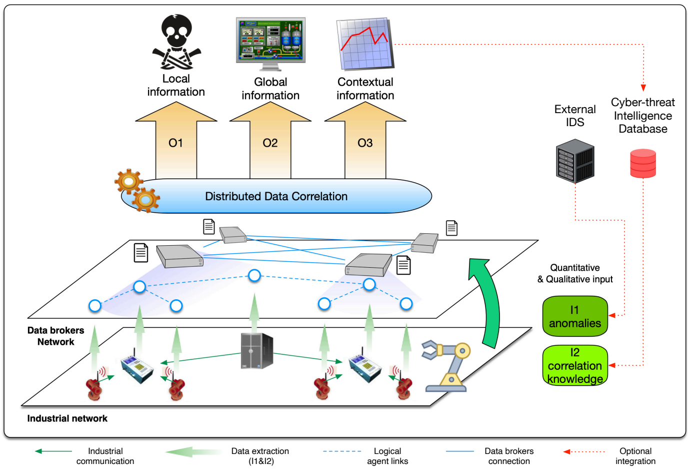


Figure 1: Distributed detection framework

zones.

(O3) Contextual information that permits to correlate past events and visualize the evolution of the threat, while anticipating the resources that are prone to be compromised (D3 & D4).

This comprehensive analysis of the requirements and techniques defines a framework for the development of distributed detection solutions for APTs in industrial scenarios, as depicted in Figure 1. The following section presents some of the candidate solutions that implement them and hence achieve the APT traceability goals proposed so far.

5 Distributed traceability solutions

After explaining the proposed framework, we look for feasible solutions that can effectively accommodate all of its statements. More specifically, we revisit the original Opinion Dynamics approach and compare it with two alternative mechanisms. The former is based on the concept of consensus and the two latter are based on clustering.

5.1 Opinion Dynamics

Despite its novelty, the Opinion Dynamics approach has faced a number of improvements since its inception in [18]. It was originally defined as a mechanism to address the anomalies caused by a theoretical APT, whose attacker model was better formalized in [19]. The event correlation and traceability capabilities were updated in [16], to latter show its implementation on an industrial testbed in [5]. Additionally, it has been studied its application to the Smart Grid [20] or the Industrial Internet of Things [21]. Compared to these, the aim of the framework is to ease the design of alternative solutions with results equal to or better than those of Opinion Dynamics.

The correlation approach of Opinion Dynamics is based on an iterative algorithm that takes the anomalies of individual agents as input, and generates their resulting 'opinions' based on this formula:

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t)$$

Where $x_i(t)$ stands for the opinion of the agent ($i \in \{1, ..., |V|\}$) at iteration t, such that $x_i(0)$ contains the initial anomaly sensed in vector x prior to execute the correlation (I1), as stated in Section 4.3. As for w_{ij}, it represents the weight given by each agent i to the opinion of each neighbour j in $G(V, E)$, as to model the influence between them (I2). At this point, the original paper defined a $\varepsilon = 0.3$ threshold to hold the maximum difference in opinions between every pair of agents i and j, to associate a weight $w_{ij} > 0$. This way, the weight given by an agent i is equally divided and assigned to each other neighbour agent k that complies with ε (including itself), having $\sum_{k=1}^{n} w_{ik} = 1$. In [16], the authors provide a methodology to include further security factors and other metrics (e.g., QoS in communication links) when calculating these weight values.

Altogether, the correlation is performed by every agent as a weighted sum of the closest opinions, and such calculation can be performed by solely using the information from neighbouring agents, thereby adapting to the distributed architecture based on data brokers (either replicating data or not). When executing this algorithm with a high number of iterations, the outputs of all agents are distributed into different groups that expose the same anomaly value, which correspond to related attacks. As a result, the network is polarized based on their opinions, hence satisfying O1. From these values, it is also possible to study the degree of affection in different network zones and extract global security indicators (O2). Likewise, the evolution of a sequential APT can also be visualized if we account for the agents opinions over time (O3), as described in [5].

5.2 Distributed anomaly clustering solutions

Opinion Dynamics belongs to a set of dynamic decision models in complex networks whose aim is to obtain a fragmentation of patterns within a group of interacting agents by means of *consensus*. This fragmentation process is locally regulated by the opinions and weights of the nodes, that altogether abstracts the APT dynamics and its effects on the underlying network. This ultimately enables to take snapshots of the current state of the network and highlight the most affected nodes, thereby tracing APT movements from anomaly events.

This rationale can also be applied to define different mechanisms with similar results. We propose to adapt clustering algorithms as an alternative for the correlation of events that fulfill the defined framework. These have been traditionally used as an unsupervised method for data analysis, where a set of instances are grouped according to some criteria of similarity. In our case, we have devices that are affected by correlated attacks (see Section 4.2) and show similar anomalies, which results in the devices being grouped together.

Classical clustering methods [22], such as K-means, partition a dataset by initially selecting k cluster centroids and assigning each element to its closest centroid. Centroids are repeatedly updated until the algorithm converges to a stable solution. In our case, the anomalies detected by the agents (denoted by the vector x) play the role of the data instances to be grouped into clusters. However, the parametrization of this kind of algorithms impose two main challenges to properly comply with the inputs and outputs of an APT traceability solution:

- **The election of k.** It is one classical drawback of the K-means, since that value has to be specified from the beginning and it is not usually known in advance, as in this case. Numerous works in the literature have proposed methods for selecting the number of clusters [23], including the use of statistical measures with assumptions about the underlying data distribution [24] or its determination by visualization [25]. It is also common to study the results of a set of values instead of a single k, which should be significantly smaller than the number of instances. The aim is to apply different evaluation criterion to find the optimal k, such as the Calinski and Harabasz score (also known as the Variance Ratio Criterion) [26], that minimizes the within-cluster dispersion and maximizes the between-cluster dispersion.

- **Representation of topological and security constraints.** By applying K-means, we assume the dataset consists of a set of multi-dimensional points. However, here we have an one-dimensional vector of anomalies in the range [0,1]. Also, the clusterization of these values is subject to the topology and the security correlation criteria which might determine that, for example, two data points should not be grouped in the same cluster despite having a similar anomaly value. Therefore, it becomes necessary to provide this knowledge to the algorithm and reflect these environmental conditions as inputs (I1 and I2) to the correlation. In this sense, some works have proposed a constrained K-means clustering [27], and specific schemes have been developed to divide a graph into clusters using Spanning Trees or highly connected components [28].

As for the first challenge, we can assume that the value of k is defined by the different classes of nodes within the network depending on their affection degree, which corresponds to the number of consensus between agents that Opinion Dynamics automatically finds. Here we can adopt two methodologies: (1) a *static* approach where we consider a fixed set of labels (e.g., ‘low’, ‘medium’, ‘high’ and ‘critical’ condition) to classify each agent; or (2) a *dynamic* approach where k is automatically determined based on the number and typology of attacks. In this case, we can study the Variance Ration Criterion in a range of

k values (e.g., k={1-5}) to extract the optimal value with the presence of an APT.

This procedure needs further improvements to make the solution fully distributed, so that each agent is in charge of locally deciding its own level of security based on the surrounding state, instead of adopting a global approach for all nodes. This bring us to the second challenge. A first naive solution would be to introduce additional dimensions to the data instances representing the coordinates of every node, together with the anomalies in vector x. We call this approach *location-based clustering*. However, this approach still needs to figure out an optimal value of k, and does not take into account the presence of actual links interconnecting nodes in $G(V, E)$.

To circumvent this issue while also adopting an automatic determination of the number of clusters, we propose an *accumulative anomaly clustering* scheme, which is formalized in Algorithm 2. This algorithm begins by selecting the most affected node within the network and subsequently applies the influence of their surrounding nodes. This is represented by adding an entire value to the anomalies of such agents (initially from 0 to 1), which is proportional to the anomaly of the influencing node (see *max* in the algorithm). This addition is performed as long as the difference between both anomalies (i.e., the influencing and influenced node) does not surpasses a defined threshold ε, similar to the Opinion Dynamics approach in order to comply with I2. Then, the algorithm continues by selecting the next one in the list of nodes inversely ordered by the anomaly value, until all nodes have been influenced or have influenced others. At that point, k is automatically assigned with the number of influencing nodes, and K-means is ready to be executed with the modified data instances. The resulting values of each agent corresponds to the decimal part of their associated centroid. This is comparable to the 'opinions' in the Opinion Dynamics approach.

The intuition behind this model of influence between anomalies (which can be enriched to include extra security factors to specify I2) assumes that successive attacks raise a similar anomaly value in closest agents, as Opinion Dynamics suggests. At the same time, it addresses the issue of selecting k and including topological information to the clusterization. It is validated from a theoretical point of view in Appendix A. In the following, the accuracy of these correlation approaches are compared under different attack and network configurations.

6 Experiments and discussions

After presenting some alternative solutions to Opinion Dynamics that fulfill the distributed detection framework presented in Section 4, this section aims to put these approaches to the test. More specifically, we consider the attacker model explained in Section 4.2, which is applied against a network formalized by $G(V, E)$, following the structure introduced in Section 4.1. These theoretical APTs generate a set of anomalies that serve as input to compare the traceability capabilities of each correlation approach:

- **Location-based clustering:** as presented earlier, it consists of the K-

Algorithm 2 Accumulative anomaly clustering

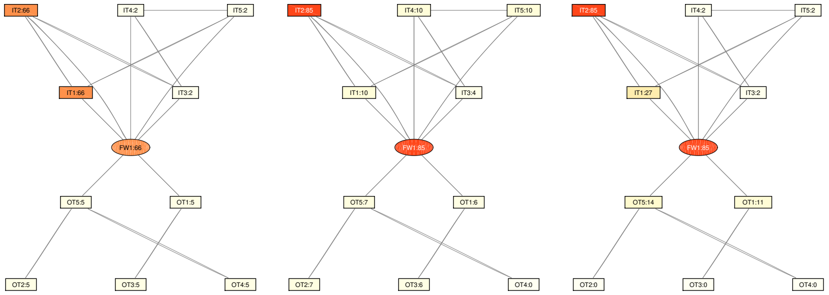
```
input: $x_i$ representing the initial anomaly value sensed by each agent i within the network, where $x_i \in (0, 1)$
output: $z_i$ representing the agents O1 output of each agent i after clustering
local: Graph $G(V, E)$ representing the network, where $V = V_{IT} \cup V_{OT} \cup V_{FW}$

$max \leftarrow |V|, k \leftarrow 0$
$y \leftarrow x, x' \leftarrow x$ sorted in descending order
for all $i \in x'$ do
    $anyNeighbourFound \leftarrow False$
    for all $j \in neighbours(i, G)$ do
        if $y_j \leq 1$ AND $|y_i - y_j| \leq \epsilon$ then
            $y_j \leftarrow y_j + max * 10$
            $anyNeighbourFound \leftarrow True$
        end if
    end for
    $y_i = y_i + max * 10$
    if $anyNeighbourFound$ then
        $k \leftarrow k + 1$
    end if
    $max \leftarrow max - 1$
end for
$clusters, centroid \leftarrow kmeans(y, k)$
for all $v_i \in V$ do
    $c \leftarrow clusters(v_i)$
    $Z_i \leftarrow IntegerPart(centroid(c))$
end for
```

means algorithm taking the anomalies and coordinates of each node as data instances. These are grouped in a number of clusters, k, which is selected in the range from 1 to 5 according to the Variance Ratio Criterion.

- **Accumulative clustering:** as previously presented, it allows to distributedly locate the infection while automatically determining the optimal k.
- **Opinion Dynamics:** is the approach that serves as inspiration for our framework and serves for comparison with the novel detection methods introduced above.

These traceability solutions are simulated under different network and attack configurations, as explained next. We start by running a brief attack test-case that illustrates the features of each approach in a simple network scenario. Based on Algorithm 1, Figure 2 shows the detection outputs (O1 and O3) of the three approaches when correlating the anomalies of an APT perpetrated against a simple infrastructure. This network is modelled according to the concepts introduced in Section 4.1, to include an IT and OT section of nodes connected by a firewall. Concretely, the figure shows an snapshot of the detection state after the adversary has performed a lateral movement from IT node 2 to compromise the firewall. The numeric value assigned to each node represents O1, which will attenuate over time to highlight the most recent anomaly, according to O3.

As noted in the figure, location-based clustering fails to accurately determine where the threat is located and selects a wide affection area instead, which is composed by *IT1*, *IT2* and *FW1* nodes (i.e., grouped in the same cluster due



(a) Location-based clustering (b) Accumulative clustering (c) Opinion Dynamics

Figure 2: Network topology used in the test case

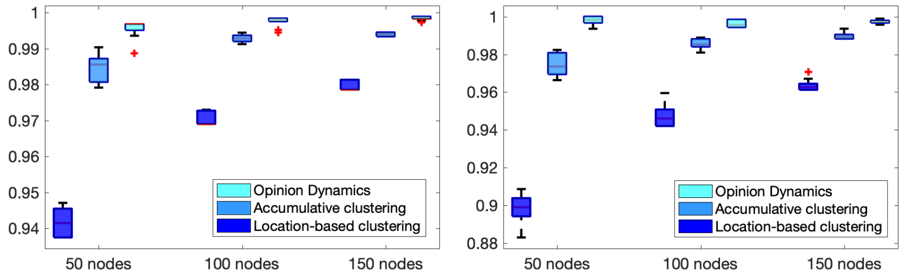
to the average anomaly in such zone). On the other hand, the accumulative clustering and Opinion Dynamics show a similar result, and successfully identify both *IT2* and *FW1* as the affected nodes in this scenario. As for the rest of nodes, they agree on a subtle affection value due to the noise present in the network and the anomalies sensed in the vicinity of the attacked nodes. As previously stated, this is modelled in a probabilistic way [5].

We now execute these solutions with a more complex network and APT model in order to study their accuracy. In the context of cluster analysis, the 'purity' is an evaluation criteria of the cluster quality that is applicable in this particular scenario. It holds the percentage of the total number of data points that are classified correctly after executing the clustering algorithm, in the range [0,1]. It is calculated according to the following equation:

$$Purity = \frac{1}{N} \sum_{i=1}^{k} max|c_i \cap t_j| \tag{1}$$

where N is the number of nodes, k is the number of clusters, c_i is a cluster in C and t_j is the classification that has the max count for cluster c_i. In our case, by 'correct classification' we mean that a cluster c_i has identified a group of nodes that have actually been compromised, which is determined in the simulations (but not known by the traceability solutions). This value can be calculated after a single execution of these three approaches to study how the results of the initial test-case escalate to larger networks and more challenging APTs.

Specifically, we run 10 different APTs on randomly generated network topologies of 50, 100 and 150 nodes, respectively. For simplicity, we start by executing an individual instance of the Stuxnet APT [5] according to the attacker model established in Section 4.2. This attack can be formally defined by the following succession of stages:



(a) Single APT

(b) Two APTs

Figure 3: Purity average for the three test cases

$$attackSet_{Stuxnet} = \{initialIntrusion_{IT}, LateralMovement_{FW},$$
$$LateralMovement_{OT}, destruction\}$$

At this point, it is worth mentioning that the lateral movement in the OT section is performed three times to model the real behavior of this APT and its successive anomalies, as explained in [5]. The purity value is then calculated after every attack stage of each of the ten APTs, to ultimately compute its average with respect to the number of nodes that have been successfully detected and grouped in the cluster with highest value of affection.

Figure 3(a) represents these average values in the form of boxplots, where each box represents the quartiles of each detection approach given the different network configurations. As it can be noted, the Opinion Dynamics stands out as the most accurate solution, closely followed by the accumulative clustering approach. The purity of the location-based clustering falls behind, and the three of them increase their value as the network grows in size due to the higher number of nodes that are successfully deemed as healthy and hence not mixed with those that are indeed affected by the APT.

Similar results are obtained when we execute two APT attacks in parallel over the same network configurations, as shown in Figure 3(b). In this case, the former APT is coupled with another attack, which can be assumed to be part of Stuxnet or a completely different attack trace within the network, composed by the following stages:

$$attackSet_{AnotherAPT} = \{initialIntrusion_{OT}, LateralMovement_{FW},$$
$$LateralMovement_{IT}, destruction\}$$

The second APT is located in a different area of the network so that it begins by sneaking into the OT section to subsequently propagate towards the IT portion of the infrastructure. This causes the spread of anomalies throughout

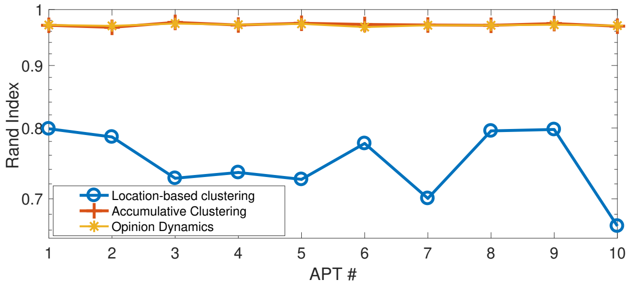


Figure 4: Evolution of the Rand Index for 10 APTs and 150 nodes

the network hence putting location-based clustering to the test. Despite a subtle decline in the purity of the solutions (especially in the location approach due to the anomaly dispersion), they still output an appreciable accuracy.

On the other hand, the superiority of Opinion Dynamics and accumulative clustering over the first approach is also evident with the study of additional accuracy indicators, such as the *Rand Index*. It penalizes both false positive (FP) and false negative (FN) labeling of affected nodes during clustering, with respect to true positive (TP) and true negative (TN) decisions, according to the following formula:

$$Rand\,Index = \frac{TP+TN}{TP+FP+FN+TN} \tag{2}$$

Figure 4 shows the Rand Index value after each of the ten APTs in the previous experiment (each one composed of two parallel attack traces), for the largest network size (150 nodes). The plot clearly shows a steady accuracy of the two latter approaches (close to 1), contrasting with a lower value in the location-based approach, which faces a lack of precision when it comes to correctly locate the affection areas, for the same reasons discussed before. Despite the promising results of our clustering approach in terms of accuracy, we are currently in the process of assessing its performance (compliance with S5 requirement).

7 Conclusions

The irruption of APTs is demanding for the development of innovative solutions capable of detecting, analyzing and protecting current and upcoming critical infrastructures. After an exhaustive analysis of the security and detection requirements for these solutions, we come up with a framework for developing APT traceability systems in Industry 4.0 scenarios, inspired by a promising approach called Opinion Dynamics. This framework considers various network architectures, types of attack and data acquisition models to later define the inputs and outputs that traceability solutions should include to support the aforementioned requirements. This lays the base for the development and comparison of novel solutions in this context. As a means to validate the proposed framework, we define two novel protection mechanisms based on clustering, which feature comparable results to the Opinion Dynamics (based on consensus). According to our experiments, the proposed clustering mechanism also presents optimal traceability of events in a distributed setting. The roadmap of this research now leads to further validation and possible extensions of the proposed framework, as well as to the application of these techniques to diverse real-world industrial scenarios.

Acknowledgments

This work has been partially supported by the EU H2020-SU-ICT-03-2018 Project No. 830929 CyberSec4Europe (cybersec4europe.eu), the EU H2020-MSCA-RISE-2017 Project No. 777996 (SealedGRID), and by a 2019 Leonardo Grant for Researchers and Cultural Creators of the BBVA Foundation. The first author has been partially financed by the Spanish Ministry of Education under the FPU program (FPU15/03213) and R. Rios by the 'Captacion del Talento Investigador' fellowship from the University of Malaga.

References

[1] Ateeq Khan and Klaus Turowski. A survey of current challenges in manufacturing industry and preparation for industry 4.0. In *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry"(IITI'16)*, pages 15–26. Springer, 2016.

[2] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. *The Journal of Supercomputing*, pages 1–32, 2016.

[3] Antoine Lemay, Joan Calvet, François Menet, and José M Fernandez. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, 72:26–59, 2018.

[4] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, 46(4):55, 2014.

[5] Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang. Tracking apts in industrial ecosystems: A proof of concept. *Journal of Computer Security*, 27:521–546, 09/2019 2019.

[6] Pu Zeng and Peng Zhou. Intrusion detection in scada system: A survey. In *Intelligent Computing and Internet of Things*, pages 342–351. Springer, 2018.

[7] Juan E. Rubio, Rodrigo Roman, and Javier Lopez. Analysis of cybersecurity threats in industry 4.0: the case of intrusion detection. In *The 12th International Conference on Critical Information Infrastructures Security*, volume Lecture Notes in Computer Science, vol 10707, pages 119–130. Springer, Springer, 08/2018 2018.

[8] Ramasubramanian Sekar, Ajay Gupta, James Frullo, Tushar Shanbhag, Abhishek Tiwari, Henglin Yang, and Sheng Zhou. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 265–274. ACM, 2002.

[9] Hui Lin, Adam Slagell, Zbigniew Kalbarczyk, Peter W Sauer, and Ravishankar K Iyer. Semantic security analysis of scada networks to detect malicious control commands in power grids. In *Proceedings of the first ACM workshop on Smart energy grid security*, pages 29–34. ACM, 2013.

[10] Juan E. Rubio, Cristina Alcaraz, Rodrigo Roman, and Javier Lopez. Current cyber-defense trends in industrial control systems. *Computers & Security Journal*, 07/2019 2019.

[11] Nour Moustafa, Erwin Adi, Benjamin Turnbull, and Jiankun Hu. A new threat intelligence scheme for safeguarding industry 4.0 systems. *IEEE Access*, 6:32910–32924, 2018.

[12] Sujit Rokka Chhetri, Nafiul Rashid, Sina Faezi, and Mohammad Abdullah Al Faruque. Security trends and advances in manufacturing systems in the era of industry 4.0. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1039–1046. IEEE, 2017.

[13] Andrew Vance. Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing. In *2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, pages 173–176. IEEE, 2014.

[14] Guillaume Brogi and Valérie Viet Triem Tong. Terminaptor: Highlighting advanced persistent threats through information flow tracking. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2016.

[15] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J Aparicio-Navarro. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359, 2018.

[16] Juan E. Rubio, Mark Manulis, Cristina Alcaraz, and Javier Lopez. Enhancing security and dependability of industrial networks with opinion dynamics. In *European Symposium on Research in Computer Security (ESORICS2019)*, volume 11736, pages 263–280, 09/2019 2019.

[17] Seokcheol Lee and Taeshik Shon. Open source intelligence base cyber threat inspection framework for critical infrastructures. In *2016 Future Technologies Conference (FTC)*, pages 1030–1033. IEEE, 2016.

[18] Juan E. Rubio, Cristina Alcaraz, and Javier Lopez. Preventing advanced persistent threats in complex control networks. In *European Symposium on Research in Computer Security*, volume 10493, pages 402–418, 2017.

[19] Juan E. Rubio, Rodrigo Roman, Cristina Alcaraz, and Yan Zhang. Tracking advanced persistent threats in critical infrastructures through opinion dynamics. In *European Symposium on Research in Computer Security*, volume 11098, pages 555–574, Barcelona, Spain, 08/2018 2018. Springer, Springer.

[20] Javier Lopez, Juan E. Rubio, and Cristina Alcaraz. A resilient architecture for the smart grid. *IEEE Transactions on Industrial Informatics*, 14:3745–3753, 08/2019 2018.

[21] Juan E. Rubio, Rodrigo Roman, and Javier Lopez. Integration of a threat traceability solution in the industrial internet of things. *IEEE Transactions on Industrial Informatics*, 02/2020 In Press.

[22] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[23] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

[24] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[25] Jeff Bilmes, Amin Vahdat, Windsor Hsu, and Eun-Jin Im. Empirical observations of probabilistic heuristics for the clustering problem. *Technical Report TR-97-018, International Computer Science Institute*, 1997.

[26] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[27] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

[28] Satu Elisa Schaeffer. Graph clustering. *Computer science review*, 1(1):27–64, 2007.

A Correctness proof of the clustering detection approach

This section presents the correctness proof of the consensus-based detection, both the location and accumulative approach. This problem is solved when these conditions are met:

1. The attacker is able to find an IT/OT device to compromise within the infrastructure.
2. The traceability solution is able to identify an affected node, thanks to the clustering mechanism and fulfilling O1.
3. The detection can continuously track the evolution of the APT and properly finish in a finite time (termination condition), complying with O2 and O3.

The first requirement is satisfied under the assumption that the attacker breaks into the network and then moves throughout the topology following a finite path, according to the model explained in Section 4.2. Thus, an APT is defined as at least one sequence of attack stages against the network defined by $G(V, E)$. If we study each of these traces independently, and based on the distribution of G, the attacker can either *compromise* the current node v_i in the chain (as well as performing a *data exfiltration or destruction*) or propagate to another $v_j \in V$, whose graph is connected by the means of firewalls, according to the interconnection methodology illustrated in [5] and summarized in Section 4.1.

As for the second requirement, it is met with the correlation of anomalies generated by agents in each attack phase. As presented with the attacker model, the value of these anomalies are determined in a probabilistic manner, depending on two possible causes: (1) the severity of the attack suffered and the criticality of the concerned resource; or (2) an indirect effect caused by another attack in the vicinity of the monitored node. Either way, the O1 correlation helps to actually determine whether the attack has been effectively perpetrated against that node, or it belongs to another APT stage in its surroundings. This information is deduced from the combination of I2 (the contextual information) together with these anomalies (i.e., I1), by using K-means to group these nodes and associate them with actual attacks.

We can easily demonstrate the third requirement (i.e., the termination of the approach) through induction. To do so, we specify the initial and final conditions as well as the base case:

Precondition: we assume the attacker models an APT against the network defined by graph $G(V, E)$ where $V \neq \oslash$, following the behaviour explained in Algorithm 1. On the other hand, the detection solution based on clustering can firstly sense the individual anomalies in every distributed agent, hence computing I1 and I2.

Postcondition: the attacker reaches at least one node in $G(V, E)$ and continues to execute all stages until $attackSet = \oslash$ in Algorithm 1. Over these steps, it is possible to visualize the threat evolution across the infrastructure, following the procedure described in Algorithm 2 in the case of accumulative clustering, and running K-means with both I1 and spatial information, in the case of location-based clustering.

Case 1: the adversary intrudes the network and takes control of the first node $v_i \in V$, and both clustering approaches cope with the scenario of grouping healthy nodes apart from the attacked node. This is calculated by the K-means algorithm within a finite time, by iteratively assigning data items to clusters and recomputing the centroids.

Case 2: the adversary propagates from a device node v_i to another v_j, so that there exist $(v_i, v_j) \in E$. In this case, the correlation with K-means aims to group both affected nodes within the same cluster, which can be visualized graphically. As explained before, this is influenced by the attack notoriety and the closeness in the anomalies sensed by their respective agents (i.e., the threshold ϵ in Algorithm 2), as well as extra information given by I2.

Induction: if we assume the presence of $k \geq 1$ APTs in the network, each one will consider Case 1 at the beginning and will separately consider Case 2 until $attackSet = \oslash$ for all k, ensuring the traceability of the threat and complying with the postcondition. Eventually, these APTs could affect the same subset of related nodes in G, which is addressed by the K-means to correlate the distribution of anomalies (again, attempting to distinguish between attacked nodes and devices that may sense side effects), in a finite time.

This way, we demonstrate the validity of the approach, since it finishes and it is able to trace the threats accordingly.