Enhancing Security and Dependability of Industrial Networks with Opinion Dynamics

Juan E. Rubio[1], Mark Manulis[2], Cristina Alcaraz[1], and Javier Lopez[1]

[1]Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071, Malaga, Spain
{rubio,alcaraz,jlm}@lcc.uma.es
[2]Surrey Centre for Cyber Security (SCCS), University of Surrey, Guildford, United Kingdom,
m.manulis@surrey.ac.uk

Abstract

Opinion Dynamics poses a novel technique to accurately locate the patterns of an advanced attack against an industrial infrastructure, compared to traditional intrusion detection systems. This distributed solution provides profitable information to identify the most affected areas within the network, which can be leveraged to design and deploy tailored response mechanisms that ensure the continuity of the service. In this work, we base on this multi-agent collaborative approach to propose a response technique that permits the secure delivery of messages across the network. For such goal, our contribution is twofold: firstly, we redefine the existing algorithm to assess not only the compromise of nodes, but also the security and quality of service of communication links; secondly, we develop a routing protocol that prioritizes the secure paths throughout the topology considering the information obtained from the detection system. **Keywords:** advanced, persistent, threat, opinion dynamics, quality, service, routing, protocol.

1 Introduction

Today, most critical infrastructures of all industrial sectors (such as transport or the Smart Grid) are controlled by SCADA systems (Supervisory Control and Data Acquisition), which access in real time to the devices that govern the production chain. As far as cybersecurity is concerned, these devices have traditionally lacked protection, since industrial networks had to function in isolation from other environments. However, there is currently a growing interconnection of control systems with other networks (such as the Internet) for the outsourcing of services or the storage of data, which is caused by the decrease in cost and standardization of hardware and software. As a result, there has also been a growth of reported security threats, as industrial systems are now also victims of the problems suffered by information technologies [1][2].

In this regard, the **Advanced Persistent Threats (APT)** represent the most critical hazard in recent years. These are sophisticated attacks perpetrated against a specific organization, where the attacker has considerable experience and resources to penetrate the victim's network, using a multitude of vulnerabilities and attack vectors [3]. They use stealthy techniques to go undetected for a prolonged period of time, from the initial intrusion to the subsequent propagation movements (a.k.a. lateral movements) within the APT life-cycle. Stuxnet was the first APT reported (in 2010), although many others APTs have appeared later, such as Duqu, DragonFly, BlackEnergy, and ExPetr [4].

Diverse security services must be applied to detect and deter the effects of these threats and minimize the impact on the infrastructure, combining cutting-edge technologies for accurately monitoring these threats. Traditional security solutions like firewalls or antivirus software must be coupled with advanced services (e.g., data loss prevention, advanced detection of malware, trusted computing) and security awareness procedures to protect the industrial systems from a holistic point of view, during their entire life-cycle. In this sense, Intrusion Detection Systems (IDS) represent a first line of defense against the wide range of cyber-threats leveraged by an APT. Numerous mechanisms have been proposed in the industry and academia that make use of machine learning techniques [5] or propose advanced services that analyze the internal traffic to detect specific attacks [6]. However, they only address security in precise points of the infrastructure or they do not consider the persistence of attacks over time. Consequently, there is still a need to find other defense solutions that enable the traceability of APTs throughout the control network, beyond the initial intrusion.

For this goal, authors in [7] propose a distributed consensus algorithm based on *Opinion Dynamics* [8], making use of graph theory. They demonstrate the feasibility of this novel approach to keep track of the anomalies suffered by devices over the entire network, potentially caused by an APT. This information can be used to put in place accurate mechanisms aiming to prevent the propagation of the APT or to minimize their impact. However, the initial approach from [7] does not take into account anomaly indicators concerning the Quality of Service (QoS) of the communication links. As a consequence, it has limitations in the monitoring of the network health and in the choice of countermeasures to ensure best-possible connectivity in the presence of APT. The previously proposed response technique for the maintenance of network paths does not sufficiently apply to traditional industrial scenarios. By improving the original approach we are able to design a more realistic response technique, showing the effectiveness of the Opinion Dynamics to ensure the continuity of communications in the presence of an APT. More concretely, we present a routing protocol that ensures the delivery of messages with a low probability of interception, while ensuring a decent level of QoS, resulting in a combined approach. Our contributions in this article can be summarized as follows:

- Improved Opinion Dynamics model based on anomaly indicators related to the QoS of the communication links and the security of hosts.
- Enhanced routing approach for reliable connectivity in presence of APT based on the improved Opinion Dynamics model.

The remainder of this paper is organized as follows: Section 2 introduces the preliminary concepts related to the Quality of Service indicators used in this work and the original APT-related anomaly detection approach based on Opinion Dynamics. The improved approach based on QoS indicators is proposed in Section 3 and the enhanced routing approach is presented in Section 4. Section 5 provides simulations and evaluation of our results. Section 6 draws conclusions.

2 Preliminary concepts and related work

2.1 Quality of Service indicators for Routing Protocols

Critical infrastructures governed by industrial networks require to work at all times, even in the presence of intruders; for this goal, we propose the use of a routing protocol as a response technique that uses the security information provided by a distributed detection system. However, in order to guarantee the delivery performance, this protocol must also make resource reservation and excise network control, in order to respond in a timely manner.

In traditional data networks, routing protocols simply use shortest path algorithms for the path computation, based on a single metric like hop-count or delay. In turn, QoS-aware routing protocols take into account further metrics to addressing the Quality of Service, in particular [9][10]:

- **Delay time.** It measures the time taken to transfer data across the network from one node to another. This value is often used to establish allowance limits for the communication links, in order to select the fastest route. In real-time operations, jitter or packet delay variations are used, measured with a sliding window of fractions of seconds. This is due to the dependence on the application (e.g., isolated environment of sensors, Internet connection to the SCADA system) or the network congestion, which could potentially slow down the communications.
- **Bandwidth.** It holds the maximum rate of data that can be transferred from a source to a destination per time unit. In order for the industrial devices to measure it, it is reasonable to determine the maximum bandwidth available at a given time. However, the computation of this value (along with delay) for routing purposes is a challenging problem since it can frequently change, as well as delay [11]. Also, in presence of an APT, there could not be any centralized control for allocating bandwidth among the nodes. For this reason, most existing QoS-aware routing protocols in the literature assume that the available bandwidth is known [12]. There

are some others that estimate this value with carrier-sense capability of the underlying protocols (e.g., IEEE 802.11) to measure the idle and busy time ratio, and then adding this information to the route control packets.

- **Packet loss.** Packet loss can be used to measure availability, which represents the probability that some recipient is reachable with the claimed quality at a given moment of time. The packet loss is usually calculated as the ratio of lost packets or dropped connections in connection-oriented systems (e.g., upon retrieval of information from sensors).

Based on the set of adequate metrics, QoS-aware routing protocols perform resource estimation at each node and proceed with the route selection [13][14][15]. Routes are usually chosen to maximize the available bandwidth while minimizing the delay and the loss probability. However, finding a path that simultaneously satisfies more than one constraint is a NP-Problem. For this reason, heuristic approaches resulting in more efficient algorithms are often used in the literature. For instance, [16] adopts three different criteria for the Optimized Link State Routing Protocol [17]. Another efficient scalable heuristic applied in [18] is based on Lagrangian relaxation. Another approach is based on the shortest-widest path algorithm [19], where a path with maximum bandwidth is found using a variant of the Dijkstra shortest-path algorithm and if there exists more than one such path then the one with the lowest delay is chosen.

Apart from these approaches, it is also possible to generate a single QoS metric from multiple parameters of the communication links. For the sake of simplicity and with the aim of aggregating different metrics (i.e., delay, bandwidth, packet loss ratio) our approach utilizes the following QoS function [20]:

$$\mathcal{S}(c) = \frac{B(c)}{D(c) \times L(c)} \tag{1}$$

where for a given communication link c, the metrics applied are the link's bandwidth B, delay D and packet loss L. Due to the reasons discussed before, the estimation of these metrics at each node is out of the scope of this article.

The output of $\mathcal{S}(c)$, when evaluated for a given communication link, is directly proportional to the quality of service that it experiences. This information can already be used for establishing a priority when selecting the routes along the network. However, besides the QoS measures applied to communication links, we will also introduce a security-based criterion for the selection of nodes that are traversed by our routing protocol. This additional information is provided by the Opinion Dynamics based detection system, explained in the following.

2.2 Using Opinion Dynamics for APT Detection

Compared to traditional defense solutions, Opinion Dynamics [8] has been demonstrated to be a suitable technique for APT detection, as originally described in [7]. In a later publication, its authors extend this work to enable the traceability of attacks along their whole life-cycle by analyzing the movements and anomalies suffered across the affected network [21].

From a general perspective, this distributed cooperative algorithm models the behavior of a group of individuals in a society: each one (which we will refer to as 'agent') holds his/her own opinion, which is, to a certain extent, influenced by the rest (and so does his/her opinion). After some time, the entire society is fragmented into formed consensus of distinct spectra of opinions, belonging to agents who are closer in their posture. Applied in the context of intrusion detection (with multiple of these agents deployed over the infrastructure), Opinion Dynamics can help to identify the portions of the network which are subject of an attack (and their criticality degree), by correlating the anomalies (subtle or evident) sensed by agents. At the same time, it is possible to trace events occurred in the network from the very first moment the intruder breaks in.

The formal description of the algorithm and how to apply consensus to this particular context is explained in the following. Suppose a network architecture given by the graph $G(V, E)$, where V represents the set of devices within the production chain (e.g., controllers, sensors or actuators) and E refers to the communication links that connect these nodes. Let A be the set of agents such that $A = a_1, a_2, ..., a_n$, being $|A| = |V|$. According to Opinion Dynamics, $x_i(t)$ represents the individual opinion of agent a_i in the iteration t, which can be valued from zero to one (where one means the highest anomaly). To represent the influence between agents, each agent i assigns a weight to each neighbor j, which is denoted by w_{ij}. We assume that $\sum_{k=1}^{n} w_{ik} = 1$, in such a way that all agents account for their own opinion.

Altogether, for a single execution of the algorithm at any given time of the control system cycle, the formation of the new opinion for agent i in the next iteration $t+1$ is described with this expression: $x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t)$. This formula models the opinion as a weighted average of the rest of agents' opinions. If we successively calculate this value for many iterations, different clusters of opinions are formed when t tends to infinity, which can also be visualized in a graph. Then, this information can be used to accurately identify different attacked areas within the network, which are potentially monitored by large sets of agents that exhibit the same anomaly pattern. The more affected areas will be those which comprise a greater number of agents with a high opinion value. At this point, in terms of adapting this multi-agent algorithm to our particular scenario, two questions appear:

1) The representation of the initial opinion $x_i(0)$ for every agent i, that in practice holds the degree of anomaly detected by them. Authors of the original publication [7] arbitrarily select a set of hierarchically connected nodes within the network that play the role of agents to perform the detection; then, they model their initial opinion by computing the deviation in the *betweenness centrality* score [22] with respect to its value in normal conditions. This indicator holds the level of connectivity that every node within the topology experiences. However, as it will be analyzed later in Section 3 and mentioned in [21], Opinion Dynamics is open to include new anomaly indicators that serve as an input to agents. In our case, we are interested in representing anomalies caused by the compromise of both devices and communication links.

2) The representation of the weight given by each agent to its respective neighbors, in order to consider their influence on the opinion about the severity of the incidence detected. The original approach is based on a simple criterion to choose the weight assigned among agents: the closer two opinions of two connected nodes are (their values), the higher the weight assigned between them will be. This means that, for every agent, the weight given to its neighbors is uniformly divided into those agents whose opinion is very similar to its own, considering an ε threshold for the difference between both values. Intuitively, this simulates the fact that agents located nearby with the same degree of anomaly sensed are prone to detect the same threat in their surroundings. Again, although this may be a valid criterion to model the weight, it could be enhanced to realistically reflect other environmental conditions involved (e.g., Quality of Service), as discussed in Section 3.

After assessing the security of the network with Opinion Dynamics, we can use this information to execute defense procedures. Authors in [7] leverage a simple message routing algorithm to ensure the reachability of nodes in presence of an attack. However, more techniques can be combined and deployed dynamically, either proactive (e.g., placing honeypots over the affected zones to gain knowledge from the adversary or using redundancy of links between agents) or reactive (e.g., recovery of nodes or links to reduce the impact on the infrastructure). In this sense, a potential study could be conducted to find an effective defense strategy (e.g., through specific validation approaches like game theory). In this paper, we show the weaknesses of the original proposal and illustrate the utility of the detection with an alternative solution that addresses those issues. This will be described later in Section 4.

3 Modified Opinion Dynamics approach: analysis of communication links

As argued in Section 2.2, the original approach based on Opinion Dynamics for the APT detection [7] requires further improvement. First, the aforementioned approach only focuses on the detection of topological changes over a graph-defined network, where a subset of nodes of V (called the *Dominating Set*) are in charge of exchanging their opinions, which are represented with the ratio of change in their *betweenness centrality* indicator. Accordingly, the attacker model just contemplates the compromise of nodes to perform a removal of links. Even though this is valid to show the applicability of the algorithm using graph theory, we must go beyond and come up with different ways to model such opinion value in a real industrial ecosystem. The reason is that APTs comprise a wide range of attack vectors besides the mere denial of service, which pose a source of different anomalies (mostly subtle), that are potentially measured and correlated by the agent associated with the affected node. Therefore, the aim is to realistically analyze the security state of each node and its neighborhood, in order to create a quantitative value that would serve as an input to the Opinion Dynamics. In general, two (possibly simultaneous) approaches can be suggested for this task:

- Use an IDS to retrieve events and alerts based on which the security state of the node in question can be analyzed. This may also include events triggered by vulnerability scanners or antivirus software.
- Analyse the incoming and outgoing network traffic and perform comparison with the normal behavior, for example, by applying machine learning techniques and assessing anomalies with regard to the traffic volume, delays, network connections and protocols used.

With this, we assume that the agent would have enough input data to compute a single opinion value for the security state of its monitored device. At this point, the effectiveness from the use of specific ways to derive such value could be compared, which would strongly depend on the actual network setup (e.g., topology, technologies, communication protocols) and it is not in the scope of this paper; instead, we point out that the novelty and effectiveness of the Opinion Dynamics approach resides in the ability to correlate anomalies throughout the network and thereby get insight into the location and severity of attacks; the way to uptake the individual anomaly detection is customizable and reliant on the security scenario that we want to achieve, thereby working as a framework.

One related issue is the implementation of this mechanism in an industrial infrastructure. As discussed in [21], these agents can be either logical or physical. On the one hand, we can assume that the status of individual devices can be retrieved from a centralized entity, which consists of a computationally powerful node in charge of correlating the anomalies from all agents. Ideally, this node would then apply protection measures (e.g., honeypots, data recovery, backup servers) based on the security state of the network. In practice, this can be easily implemented by using switches in port-mirroring mode, so that all traffic from the nodes is relayed to a central correlator system, for instance. On the other hand, we could also consider that these agents can be physically deployed over the network, in form of monitoring devices or integrated with the software of the industrial assets. However, this option is not as feasible, since manufacturers and operators of critical infrastructures are reluctant to introduce modifications in their hardware and software (mainly due to computational limitations and use of privative software).

Regardless of the method used for the anomaly detection, we especially look into the security of the opinion exchange in this paper. In this regard, the original approach does not provide details about how the agents transfer their opinions between them or to a central correlator. If the same communication channels are used to deliver the Opinion Dynamics values, we must prevent against an attacker being able to compromise these links and potentially forge malicious opinions. At the same time, besides assessing the security of each node, the algorithm should also take the QoS of the communication links into consideration to safely send this information, as well as to route other messages (e.g., commands or data) between the devices. In the following, we propose a modification of the weight calculation mechanism to consider the QoS of the communication links and the confidence assigned to neighbors for the opinion transmission. This poses a solution to the second issue raised in Section 2.2.

To begin with, let's consider the original model: each agent i determines the weight given to every neighbor j in its neighborhood N_i through this expression:

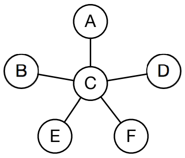
$$w_{ij} = 1/N'_i \tag{2}$$

where N'_i is the subset of neighbors of N_i, whose difference in opinion with agent i is below ε. Otherwise, w_{ij} becomes zero. Even though this is just a criterion to reflect the degree of similitude between agents, it lacks much accuracy since it leaves behind several other aspects involved; in this case, we want to introduce an additional factor to regulate this weight through considering the QoS of the channel in the neighborhood.

Let $\mathcal{S}\colon E \to \mathcal{R}$ be a function that assigns QoS scores to communication links in the network defined by $G(V, E)$, as presented in Section 2.1. The higher the score of S for a given link is, the more QoS it provides. For a given i, we aim to fairly distribute w_{ij} by giving a higher value to those agents j whose $\mathcal{S}(e_{ij})$ is greater, where $e_{ij} \in E$ represents the bidirectional communication link between i and j. This methodology complies with the following three conditions:

- **C1.** The sum of weights given by agent i to the neighbors in N'_i must be 1, also considering threshold ε. $\sum_{j=1}^{N'_i} w_{ij} = 1$.
- **C2.** The own agent i must have a sensitive fixed weight assigned to itself. For instance, we can assume $w_{ii} = 0.5$; the reason is that it is not fair that it associates a higher level of confidence to any other agent, whose link of communication can be minimally compromised.
- **C3.** The rest of weight (1/2 in this case) assigned by agent i is distributed among neighbors in N'_i proportionally to the quality of their communication links. If we define $q = \sum_{j}^{N'i} \mathcal{S}(e_{ij})$, then the resulting weight value is defined by $w_{ij} = (1 - w_{ii}) * \mathcal{S}(e_{ij})/q$.

Example. Table in Figure 1 shows the calculation of w_{ij} for the node C in the example graph (where $i = C$) following the proposed methodology, compared to the original one. The weight value that is computed using the new methodology is denoted by w'_{ij}. In both cases, a value of $\varepsilon = 0.35$ has been considered. As we can see, the new distribution of weight results more equitable, where node C assigns a higher weight to nodes A and D, since their links show a better quality and security (which is represented by the $\mathcal{S}(e_{Cj})$ column).



j	$x_j(0)$	$\mathcal{S}(e_{Cj})$	w_{Cj}	w'_{Cj}
A	0	0.9	0.25	0.26
B	0.6	0.8	0	0
C	0.22	0.55	0.25	0.5
D	0.5	0.75	0,25	0.21
E	0.12	0.1	0.25	0.03
F	0.9	0.2	0	0

Figure 1: Example of weight calculation by agent C

4 QoS-Aware Routing based on Opinion Dynamics

In response to an APT, the combined opinions determined by the monitoring agents on the industrial network with regard to the security of its nodes and the QoS aspects of their communication links can subsequently be used to enhance network routing. Here we present a novel approach aiming to enhance routing algorithms used in industrial networks such that the probability of packets being intercepted by potentially compromised network nodes is minimized while the Quality of Service of paths through which these packets are routed is maximized. This way, we can ensure the confidentiality and reliability of the network until the threat is completely eradicated from the infrastructure.

Note that our approach can also replace the initial response mechanism proposed in [7] which aims to enhance delivery of messages in presence of APT by relying on a redundant non-compromised part of the network topology and using secret sharing to split packets into chunks that are randomly dispatched over multiple paths. Their approach has a number of shortcomings as discussed in the following. First, their attack model is based on a complete removal of communication links by compromised nodes and does not consider a more realistic scenario where such links may experience varying QoS levels as a result of an attack. As observed in the recent years, many APT usually rely on zero-day vulnerabilities and make use of stealthy techniques to go unnoticed for a prolonged period of time, until they finally exfiltrate information or destroy the physical equipment. Therefore, it is necessary to consider a more subtle behavior of the attacker who may not wish to fully disrupt the communication and be detected. Second, the assumption on the existence of a redundant non-compromised topology in industrial control networks is not realistic. The architecture of such networks very frequently responds to a fixed configuration where all resources are rigidly connected with each other and so installation of a separate network topology might require significant investment and modifications of existing hardware devices. Third, their approach relies on the shortest-path estimation for which sending network nodes are assumed to know the entire network topology and has therefore limitations when used in combination with existing routing protocols that may not require nodes to have this knowledge.

Our approach is more general and realistic in that it aims at enhancing available routing algorithms to take into account the anomalies determined by the monitoring agents for the QoS levels of communication links and the security of network nodes when making routing decisions rather than selecting an optimal route based on the shortest path only. In order to set the background for our approach, we consider a typical architecture of an industrial network following the ISA-95 standard [23]. In practice, due to the modernization of industrial technologies in recent years, these networks have evolved towards a more distributed model. Control devices (i.e., PLCs or RTUs) govern the production cycle by retrieving data from field devices (i.e., sensors and actuators), according to the information exchanged with SCADA systems. These are evolving towards cloud-based solutions, that interconnect other services within the organization. This way, we see how the network is divided into two main sections: the industrial assets (which we will refer to as 'operational technologies', OT) and the IT (Information Technologies). This is the base that authors assume to extend the topology used in [7], used now to show the feasibility of our routing protocol.

Let $G(E, V)$ be a graph that describes the overall network topology. This graph is composed of the two subgraphs $G(V_{IT}, E_{IT})$ and $G(V_{OT}, E_{OT})$, which are interconnected by a set of intermediate firewalls V_{FW} so that $V = V_{IT} \cup V_{OT} \cup V_{FW}$. More specifically, both are joined by the firewalls V_{FW}, that have connections with the nodes of V_{IT} and V_{OT} that belong to the Power Dominating Set (PDS) of those subnetworks. This concept refers to a set of hierarchically selected nodes that have the maximum dominance over the entire network [24]. With respect to the network topology, we note that each of these subnetworks has a different configuration. On the one hand, $G(V_{OT}, E_{OT})$ follows a power-law distribution of the type $y \propto x^{-\alpha}$ [25], which models the hierarchical topology of an electric power grid and its high-level substations, which are subsequently connected to nodes with less connectivity (e.g., sensors and actuators). On the other hand, $G(V_{IT}, E_{IT})$ presents a small-world distribution, that models the traditional topology of TCP/IP networks on the Internet [26].

Over this distribution of nodes, there are two types of communication flows: information about the production chain delivered from the lower layers to the managerial IT network and, in reverse way, control commands issued from that section (e.g., the SCADA system) to the industrial process. For both types of the communication flows we base our approach on the Bellman-Ford algorithm [27] that is at the core of the Distance Vector Routing (DVR) [28] protocol, which determines the path to remote nodes using hop count as a metric. Each node holds a table that contains the distance to each node and the next hop in the route. This information is exchanged periodically with the neighbors, to ultimately compute the path using the Bellman-Ford algorithm. This contrasts to the Dijkstra's path-finding algorithm [29] used in [7], that finds the shortest path by requiring all nodes to have overall knowledge of the network topology and is at the core of the Link-State Routing (LSR) protocol [30]. In this protocol routers periodically flood the entire network to ensure that each node holds a synchronized copy of the routing table. By choosing DVR over LSR we can compute paths locally without involvement centralized routers as communicating with such nodes in presence of APT would impose additional risks.

The Bellman-Ford algorithm uses a weighted directed graph $G(V, E)$. The shortest distance from a node to the rest is determined by overestimating the true distance, following the principle of relaxation. In our case, since we want to prioritize QoS and security for the chosen path over the distance, we represent the weight assigned to each link $e_{ij} \in E$ as

$$W(e_{ij}) = \frac{X_t(j)}{S(e_{ij})} \tag{3}$$

where $X_t(j)$ is the final anomaly value of node j after executing the Opinion Dynamics as specified in Section 3. We select j instead of i since we want to prevent the messages against propagating to a node that is potentially compromised. On the other hand, $S(e_{ij})$ refers to the QoS score of the communication channel e_{ij}, as specified in Section 2.1. The higher the anomaly sensed by the agent in node j is, the greater the weight assigned to that link will be. Correspondingly, the S score is inversely proportional to that value. By this means, we take into consideration the security of devices and the Quality of Service of their links when deploying our response technique in form of routing protocol.

Such DVR-based routing approach can be executed at any time of the production chain, paired with a previous execution of the Opinion Dynamics algorithm for adapting the network to the current security level, thereby achieving resilience. Therefore, we assume the process to update the routes can be executed as frequently as the security scenario imposes, which would not imply additional computing costs for the devices if we consider that the detection algorithm is executed in a central correlator system separated from the industrial network, as suggested in [7]. In the following, we prove the effectiveness of our technique by simulating successive attacks against a network. Note that this approach can be validated in the future with game theory to consider dynamic attack behaviors and additional defense solutions.

5 Simulation and evaluation

In this section, our primary aim is to prove that the proposed QoS-aware routing approach based on Opinion Dynamics can effectively minimize the interception of messages, avoiding paths that contain compromised nodes while ensuring an acceptable level of quality. First, we define the attack model used in our simulation that determines how the anomalies are generated over the network and measured by the agents. Then, we execute the technique (i.e., the delivery of messages and the QoS analysis) with different parametrization of the topology and attacks performed. Finally, we evaluate the simulation findings.

5.1 Attack model: simulation of attacks and anomalies

In order to define a more realistic attack model for our response technique, we assume an attacker can break into the infrastructure by leveraging zero-day vulnerabilities and then use stealthy techniques to propagate over the network, until information is filtered or disruption to the infrastructure is caused.

Therefore, contrary to the approach based on the alteration of links, we consider an attack model based on a succession of lateral movements over the network nodes, aiming to infect as many devices as possible so that the security when delivering messages is jeopardized. Let *attackSet* be this sorted set of attack stages that an APT can perform against the industrial network, which is defined by $G(V, E)$ and is composed by the IT and OT sections, as explained in Section 4. This set comprises a finite number of elements of the following kind:

- **attackITnode**: the adversary initializes the APT or propagates the attack to a device in the IT subnetwork.
- **attackFWnode**: the attacker compromises a firewall (when the previously compromised node has connection with it), in order to propagate to the other section of the control network.
- **attackOTnode**: the intruder compromises a node in the industrial section of the network.

Every time the attacker takes over a new device, two main variables change:

1. From the **security** perspective, the agent associated with the compromised node notices an increase in the anomaly level, that ranges from zero to one, as described before. If we define x as the initial opinion vector for all agents, then x_i^t is updated in the simulation after attack number t. For simplicity, we assign a value that is randomly generated according to a uniform distribution over $(0, 1)$, simulating the existence of both subtle and evident anomalies.
2. From the perspective of **Quality of Service**, the agent also senses a potential alteration in the QoS experienced in the incoming or outgoing connections, as a consequence of the attack. The value of $S(e_{ik})$ for all $e_{ik} \in E$ in the simulations is originally chosen from a uniform distribution over $(0, 1)$, to represent the presence of channels with different QoS levels. In the event of an attack, the value of $S(e_{ik})$ and $S(e_{ki})$ scores decreases (being zero the minimum), where i is the attacked node and k refers to all neighbors of i such that there exists $e_{ik} \in E$ (since each connection is bidirectional). This decrease is represented by δ. Since the attacker can leverage stealthy techniques to go unnoticed without affecting the communications, this value is also chosen uniformly at random from $(0, 1)$.

Algorithm 1 describes the proposed APT life cycle. For all the attack stages in the provided *attackSet* parameter, the security of agents and the QoS score of

Algorithm 1 APT life cycle

```
output: $X$ representing the final opinion value for all agents, $S$ representing the QoS scores of links
local: Graph $G(V, E)$ representing the network, where $V = V_{IT} \cup V_{OT} \cup V_{FW}$
input: $attackSet \leftarrow attackStage_{APT_x}$, representing the APT chain of attacks

$x \leftarrow zeros(|V|)$ (initial opinion vector)
{$attack \leftarrow$ first attack from $attackSet$}
while $attackSet \neq \oslash$ do
    $x(attackNode) \leftarrow U(0,1), \delta \leftarrow U(0,1)$
    for neighbour in neighbours(attackNode) do
        $S(attackNode, neighbour) \leftarrow S(attackNode, neighbour) - \delta$
        $S(neighbour, attackNode) \leftarrow S(neighbour, attackNode) - \delta$
    end for

    $X \leftarrow$ ComputeOpinionDynamics$(x, S)$
    $attackSet \leftarrow attackSet \setminus attack$
end while
```

the links is reevaluated, as described before: firstly, the attacked node (specified with *attackNode*) is assigned with a random value of anomaly (i.e., the opinion of its agent) in the uniform (0,1) distribution. Then, each of its ingoing and outgoing links are updated with a diminished QoS score, according to the value of δ. Afterwards, Opinion Dynamics is executed to aggregate all opinions and calculate their final values, which eases the identification of zones under the effect of the APT following the algorithm explained in Section 3. Finally, this information can be input to the routing protocol.

5.2 Reliable message delivery

Once the attack model has been defined, we can execute the defender's code based on the routing protocol in presence of an APT to firstly show that messages are successfully delivered in a way that the probability of traversing a compromised node (i.e., with an opinion value greater than zero) is lower than using the previously proposed approach in [7]. To simulate this, a set of 100 different messages are randomly generated, whose sender and recipient belong to the graph $G(V, E)$, making sure that more than one path exists between both nodes. Half of these messages are control commands (i.e., sent from the IT section to one device in the lowest levels of the infrastructure), while the other half are data packets, generated in the production chain and dispatched to the IT subnetwork. Therefore, messages are delivered in both ways based on the industrial topology defined in Section 4.

In order to compare the level of security experienced by the response technique and consequently compare it with other solutions, we define the *compromise level* indicator for each of the messages sent. This holds the sum of anomaly values (i.e., opinions calculated with the Opinion Dynamics algorithm, represented with X in Algorithm 1), which are measured by the set of nodes that compose the path described by the message, in the route from the recipient to the destination. The greater this value is, the highest probability for the message to be intercepted will be. For a given number N of messages transmitted, we can determine the *average compromise level* as

$$\frac{\sum_{i=1}^{N}\sum_{j=1}^{|R|} X_j}{N} \tag{4}$$

where X_j is the opinion of agent j, $1 \leq j \leq |R|$, and R is the set of nodes that each message i traverses. This overall value is calculated for our custom routing protocol and will be compared with two other approaches: on the one hand, **(a) the previously proposed mechanism in [7]**, that is based on the Dijkstra's shortest-path algorithm, without considering the opinions of the agents; on the other hand, **(b) the Dijkstra's path-finding algorithm parametrized with the opinion of agents** as weights for the search of the optimal path (i.e., the route with a minimal compromise level). In other words, for the computation of the path from sender to recipient in $G(V, E)$, (a) uses a weighting function W for each edge $e_{ij} \in$ such that $W(e_{ij}) = 1$ if e_{ij} simply exists (so that the destination is reached in the minimum number of hops). As for (b), $W(e_{ij}) = X_j$, hence prioritizing not to hop to a compromised node. Our aim is to show how **(c) our approach based on Bellman-Ford algorithm**, that uses the weighting function defined in Equation 3, achieves better security (i.e., the value of compromise level) than (a), with closer results to (b).

In this experiment (carried out in Matlab), we have generated a random industrial network of 50, 100 and 200 nodes following the topology described in Section 4 (where the two halves of nodes are respectively used for the IT and OT subnetworks and an extra firewall node is used to merge them). Over these topologies, we have simulated the effect of an APT (according to algorithm 1) composed by 50, 100 and 200 attack actions, respectively. We have represented the overall behavior of Stuxnet (since it is one of the most documented attacks) at a basic level: the APT begins by compromising one node from the IT network (originally using malicious USB flash drives) and then spreads through the entire subnetwork until it finally breaks into the OT section, where the threat propagates until it infects the target device (the uranium enriching centrifuges).

By making sure the number of attacks reaches the number of nodes, we represent the most critical scenario when the APT takes over the entire network, thereby showing the effectiveness of the algorithm at all times (although this validation process could be further optimized if attacker and defender were part of a dynamically confronted competition with specific action rules, by means of game theory). After each attack phase, the Opinion Dynamics algorithm is executed and the set of 100 random messages are delivered, putting into play the three aforementioned routing algorithms. Finally, the average of compromise is calculated. The plot in Figure 2 shows the evolution of this value over the entire set of attacks for the three assessed solutions.

As we can see, the Dijkstra's algorithm that uses the opinion of agents as weights to compute the path serves as the baseline of the minimum compromise level that can be achieved. However, our approach based on Bellman-Ford algorithm presents a similar result with a slight increment of anomaly experienced, that still remains far from the high value experienced by the Dijkstra's scheme proposed in [7], as we wanted to demonstrate. We will now prove that our approach also provides better Quality of Service requirements.

5.3 Quality of Service experienced

After analyzing how our solution effectively experiences a lower level of compromise when routing the messages, it is also necessary to prove that the paths generated by the protocol also achieve an adequate Quality of Service, which is the main contribution of this paper. This would ensure a fast and reliable communication, especially necessary when the computed paths impose several hops to reach the recipient as a consequence of avoiding the effect of the attack.

Following the previous methodology, we aim to deliver a set of 100 messages over the graph $G(V, E)$ in such a way that the number of hops is minimized and the Quality of Service experienced is maximized. This time, we define the *QoS level* indicator for each message sent as the sum of individual QoS scores for all the successive edges that belong to the path (as explained in Section 2.1) divided by the number of hops that this message performs. The greater this value is, the better quality of service with a lower number of nodes traversed will be. Given N messages transmitted, we can determine the *average QoS level* as

$$\frac{\sum_{i=1}^{N} \frac{\sum_{j=1}^{|R|} S(e^j)}{hops_i}}{N} \tag{5}$$

where S is the QoS score function from Equation 1, e^j refers to edges from the route R which is taken by message i, and $hops_i$ is the number of intermediate hops. This average QoS value is calculated for our routing approach in presence of APT using the same topology and attack scenarios as in the previous test case, and is compared with the two other approaches: **(a) the previously proposed mechanism in [7]**, that is based on the Dijkstra's shortest-path algorithm without accounting for any QoS implications; and **(b) the Dijkstra's path-finding algorithm parametrized with the QoS score of the edges** as weights for the search of the optimal path (i.e., the route with a maximum quality). Thus, (a) uses an W weighting function for each edge e_{ij} such that $W(e_{ij}) = 1$ if e_{ij} simply exists, while in (b) it uses $W(e_{ij}) = 1/S(e_{ij})$, hence prioritizing the path with maximum Quality of Service. In this case, our aim is now to show how **(c) our approach based on Bellman-Ford algorithm**, that uses the weighting function defined in Equation 3, achieves a better QoS level than (a), with closer results to (b).

The plot in Figure 3 represents the evolution in the average QoS levels. As the previous test case, the QoS-aware Distance Vector Routing presents a QoS level per hop ratio similar to the Dijkstra's algorithm weighted with the QoS scores. As we can see, the three routing approaches have their QoS levels diminished as the APT evolves (due to the attacks and consequent decrease of the S scores, as explained in Algorithm 1), but our approach shows a higher QoS level, close to the one experienced by the optimal Dijkstra's solution. Therefore, we have demonstrated our reliable routing approach behaves in a nearly optimal way, more efficiently than the original response technique [7]. Figures 2 and 3 prove that QoS- or security anomaly-based routing alone are not sufficient, since both criteria must be complied to ensure a delivery of messages balanced with a decent level of security and QoS. In addition, table-driven routing algorithms like DVR with the Bellman-Ford algorithm also ensure an ad-hoc selection of routes without any central entities involved in the communications, which can help achieving a higher level of security while alleviating the large amount of traffic that route updates like the original protocol can imply.

6 Conclusions and future work

Nowadays, APTs impose a major problem for the security of Industrial Control Networks, for which the Opinion Dynamics approach has been shown to represent a promising solution. In this work, we have revisited the original approach to analyze its applicability and enhance the algorithm with the aim of enabling reliable communications. We have defined an aggregated Quality of Service score that permits to prioritize the opinions of agents transmitted through trustworthy links and made the detection system account for anomalies in the communication channels besides anomalies sensed on the nodes. We have further developed an alternative routing approach that uses the anomaly information measured by the agents in relation to the security of nodes and the QoS indicators of the communication links to choose paths with an almost optimal degree of security and QoS. Finally, the superiority of our approach over prior work has been demonstrated with an extended attack model and simulations using modern industrial network topologies. Our ongoing work involves defining an extended set of response solutions coupled with a more realistic attack model, with the aim of finding an adaptable defensive strategy using game theory. This is being addressed in TI&TO, a two-player game where the defender leverages multiple protection measures based on the Opinion Dynamics detection approach.

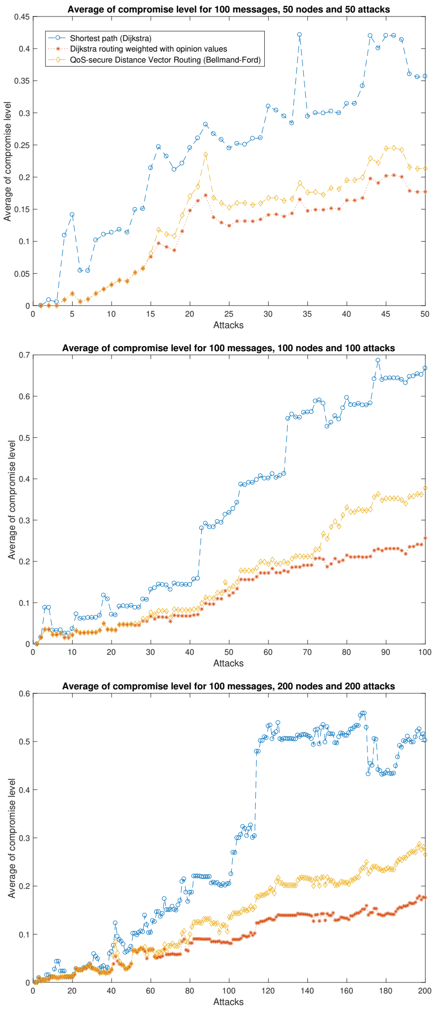


Figure 2: Average compromise level

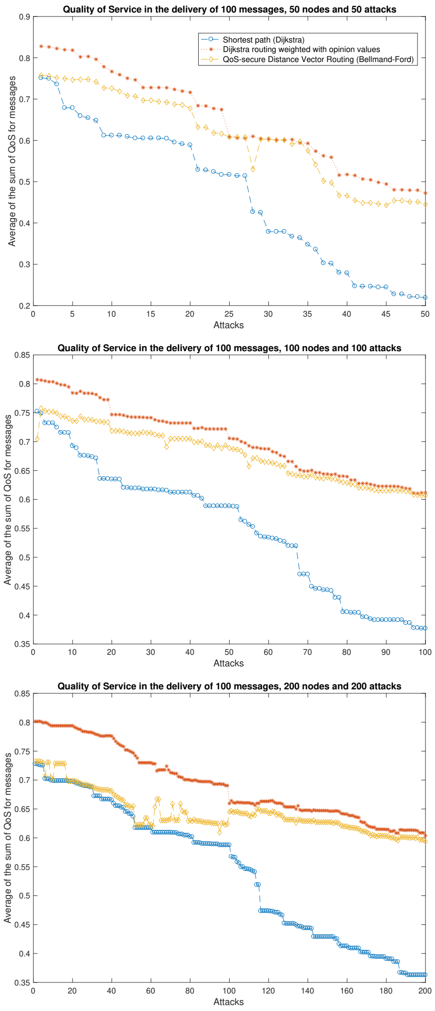


Figure 3: Average QoS level

Acknowledgments

This work has been partially supported by the research project SealedGRID (H2020-MSCA-RISE-2017), financed from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 777996, as well as by the project DISS-IIoT, financed by the University of Malaga (UMA) through the "I Plan Propio de Investigación y Transferencia". Likewise, the work of the first author has been partially financed by the Spanish Ministry of Education under the FPU program (FPU15/03213).

References

[1] ICS-CERT. Overview of Cyber Vulnerabilities. `http://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities`, 2018. [Online; Accessed July 2018].

[2] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[3] 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.

[4] 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.

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

[6] Mirco Marchetti, Fabio Pierazzi, Michele Colajanni, and Alessandro Guido. Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks*, 109:127–141, 2016.

[7] 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.

[8] Rainer Hegselmann, Ulrich Krause, et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3), 2002.

[9] Hakim Badis and Khaldoun Al Agha. Qolsr, qos routing for ad hoc wireless networks using olsr. *European Transactions on Telecommunications*, 16(5):427–442, 2005.

[10] Eric Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A framework for qos-based routing in the internet. Technical report, 1998.

[11] Chunhung Richard Lin and Jain-Shing Liu. Qos routing in ad hoc wireless networks. *IEEE Journal on selected areas in communications*, 17(8):1426–1438, 1999.

[12] Lei Chen and Wendi B Heinzelman. A survey of routing protocols that support qos in mobile ad hoc networks. *IEEE Network*, 21(6), 2007.

[13] Francis Joseph Ogwu, Mohammad Talib, Ganiyu A Aderounmu, and Adedayo Adetoye. A framework for quality of service in mobile ad hoc networks. *Int. Arab J. Inf. Technol.*, 4(1):33–40, 2007.

[14] Shigang Chen and Klara Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *IEEE network*, 12(6):64–79, 1998.

[15] Afreen Begum Sana, Farheen Iqbal, and Arshad Ahmad Khan Mohammad. Quality of service routing for multipath manets. In *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*, pages 426–431. IEEE, 2015.

[16] Ying Ge, Thomas Kunz, and Louise Lamont. Quality of service routing in ad-hoc networks using olsr. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2003.

[17] Thomas Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). Technical report, 2003.

[18] Hakim Badis and Khaldoun Al Agha. A distributed algorithm for multiple-metric link state qos routing problem. In *Mobile And Wireless Communications Networks: (With CD-ROM)*, pages 141–144. World Scientific, 2003.

[19] Hakim Badis and Khaldoun Al Agha. Quality of service for the ad hoc optimized link state routing protocol (qolsr). 2005.

[20] Zheng Wang and Jon Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on selected areas in communications*, 14(7):1228–1234, 1996.

[21] 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.

[22] Sen Nie, Xuwen Wang, Haifeng Zhang, Qilang Li, and Binghong Wang. Robustness of controllability for networks based on edge-attack. *PloS one*, 9(2):e89066, 2014.

[23] International Society of Automation. ISA-95 standard. `https://www.isa.org/isa95/`, last retrieved in December 2017, 2017.

[24] Teresa W Haynes, Sandra M Hedetniemi, Stephen T Hedetniemi, and Michael A Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002.

[25] Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

[26] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[27] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[28] Mahesh K Marina and Samir R Das. On-demand multipath distance vector routing in ad hoc networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 14–23. IEEE, 2001.

[29] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[30] Bernard Fortz and Mikkel Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE journal on selected areas in communications*, 20(4):756–767, 2002.