Trustworthy Industrial IoT Gateways for Interoperability Platforms and Ecosystems

Francisco Fraile, Takuya Tagawa, Raul Poler, and Angel Ortiz

***Abstract*—The industrial Internet of Things (IIoT) is having a significant impact in the manufacturing industry, especially in the context of horizontal integration of operational systems in factories as part of information systems in supply chains. Manufacturing companies can use this technology to create data streams along the supply chain that monitor and control manufacturing and logistic processes, to in the end make these data streams interoperable with other software systems and to enable smart interactions among supply chain processes. However, the provision of these data streams may expose manufacturing operational systems to cyber-attacks. Therefore, cybersecurity is a critical aspect to design trustworthy gateways, which are system components that implement interoperability mechanisms between operational systems and information systems. Gateways must provide security mechanisms at different system layers to minimize threats. This paper presents the Device Drivers security architecture: trustworthy gateways between operational technology and information technology used in the virtual factory open operating system (vf-OS) platform, which is a multisided platform orientated to manufacturing and logistics companies to enable collaboration among supply chains in all sectors. The main contribution of this paper is the evaluation of fallback mechanisms to improve resilience. In situations when the system may be under attack, the proposed mechanisms provide means to quickly recover component availability, by applying alternative security measures to minimize the threat at the same time. Other significant contributions are: a description of the threat model for Device Drivers, a presentation of the security countermeasures implemented in the vf-OS system, the mapping of the vf-OS response objectives to the different characteristics of a trustworthy system: security, privacy, reliability, safety, and resilience and how the proposed countermeasures complement this response.**

***Index Terms*—Application virtualization, communication system security, industrial communications.**

I. Introduction

THE INDUSTRY 4.0 [1] concepts and technologies outline the future approach to supply chain operations. In this new scenario, smart products, smart equipment, software, and people interact to dynamically optimize operations, based on real time analytics and supply chain process automation. One of the backbones of this vision is the industrial Internet of Things (IIoT) [2]. This technology enables the creation of data streams throughout the production process (i.e., vertical integration) and among supply chains (i.e., horizontal integration) that can be used to monitor manufacturing assets and/or value streams in real time. In combination with other technologies like cloud storage, data analytics, big data, virtualization [3], or microservices architecture they provide the basis for gaining predictive insights in any supply chain process [1], from customer relationship management, through manufacturing flow management, to returns management [4]. Some of the applications of these technologies are to support the transition from product-centric to service-centric business models in manufacturing, waste reductions for lean manufacturing operations or support to collaborative manufacturing.

Fig. 1. vf-OS platform.

Interoperable digital manufacturing platforms [5] and ecosystems, such as the virtual factory open operating system (vf-OS) platform [6], are multisided platforms that address manufacturing and logistic companies. vf-OS enables the exploitation of Industry 4.0 technologies through a range of services to integrate better manufacturing and logistics processes within organizations and among supply networks. The vf-OS platform concept is illustrated in Fig. 1.

The value proposition for the different customer groups of the multisided platform is clear.

1) Manufacturing users can select and use vApps from the marketplace to integrate manufacturing and logistics processes, enabling collaboration in the value chain. If they do not find any suitable applications for their

Manuscript received November 30, 2017; revised March 2, 2018 and April 11, 2018; accepted April 18, 2018. Date of publication May 1, 2018; date of current version January 16, 2019. This work was supported by the European Commission under the Grant 723710. *(Corresponding author: Francisco Fraile.)*

F. Fraile, R. Poler, and A. Ortiz are with the Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de Valencia, 460232 Valencia, Spain (e-mail: ffraile@cigip.upv.es; rpoler@cigip.upv.es; aortiz@cigip.upv.es).

T. Tagawa is with the Department of Civil Engineering and Industrial Management Engineering, Nagoya Institute of Technology, Nagoya 4668555, Japan (e-mail: cjr17049@nitech.jp).

Digital Object Identifier 10.1109/JIOT.2018.2832041

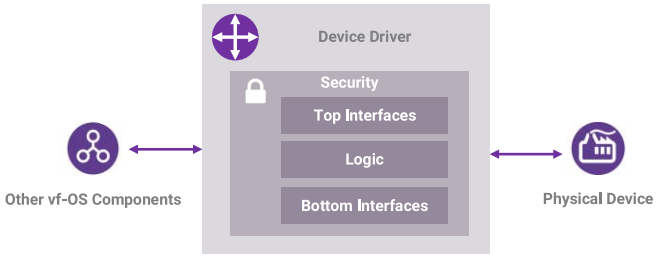


Fig. 2. Device driver high level architecture.

needs, they can demand new solutions to be developed to software developers.

2) Software developers gain access to a new and high-growth potential market of applications for Industry 4.0 and factories of the future.
3) Manufacturing solutions providers that deliver products and services to manufacturing users have new ways for collaborating and interacting with their customers and provide added value.
4) Service providers can provide new services (hosting, storage, cloud services, etc.) to realize the vf-OS ecosystem.

Within the vf-OS system architecture, certain components have been designed to interact with all kinds of manufacturing assets, both physical devices (e.g., PLCs or sensors) and business software applications [7] (e.g., ERPs or CRMs). These components, known as input–output (IO) components, make up the virtual factory I/O and implement interoperability mechanisms that are specifically addressed to manufacturing processes.

In particular, Device Drivers are components designed to interconnect physical devices to the vf-OS platform as shown in Fig. 2. They can be regarded as gateway components with secured core functionalities to integrate physical devices into the vf-OS platform. Device Drivers interact with physical devices through what is known as bottom interfaces. They implement inner logic functions (e.g., edge processing) and the vf-OS platform interacts with Device Drivers through top interfaces. Likewise, API connectors are components to interconnect on premise software to the vf-OS platform.

Cybersecurity mechanisms must be implemented both in the design and the execution of every component functionality. To that end, vf-OS provides the vf-OS holistic security and privacy concept, which is the framework for all security countermeasures in the vf-OS ecosystem. This research describes how the vf-OS response applies to the security of Device Drivers and proposes complementary countermeasures which can be used to further improve system trustworthiness. This paper is structured as follows. Section II describes the main security concepts taken into account in the design of vf-OS Device Drivers. Section III describes the complementary countermeasures proposed in this paper, which are based on the fallback principle. Finally, Section IV includes some final conclusions.

II. Security Concepts

The Device Drivers that implement interoperability mechanisms with industrial control system (ICS) components represent a very critical environment from a security perspective, since cyber-attacks may not only cause great economic losses to manufacturing companies, but may also pose a risk for operators or the environment. The confidentiality, integrity, and availability triad model is a classic model to guide information security policies within an organization. It can be noted that the priorities of operational systems and information systems regarding these system characteristics are not the same. Moreover, [8] provides a more elaborated definition of the system characteristics that enable system trustworthiness and the different priorities for information systems, operational systems, and IIoT systems integrating both. From this perspective, Device Drivers can be regarded as IIoT system components, as they combine the requirements and regulatory constraints of both information technology (IT) systems and operational technology (OT) systems regarding the system characteristics that enable trustworthiness.

1) *Security:* Security ensures that the system is protected from unintended or unauthorized access, change, or destruction.
2) *Privacy:* Privacy provides organizations control over the collection, processing, and storage of their information, by deciding how this information can be shared both within their own organization and with others.
3) *Reliability:* Reliability guarantees that the system's operation is uninterrupted and error-free for the specified time. Availability is related to reliability, but also takes into account planned operation stops.
4) *Safety:* System Safety ensures that the people, property and environment are not at any unacceptable risk during the system's operation.
5) *Resilience:* System resilience [8] provides means to dynamically avoid, absorb and rapidly recover from changing adverse conditions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents.

The combination of IT environments and OT environments means that Device Drivers have increased requirements for each system's trustworthiness characteristic compared to either OT or IT systems. This is presented in Fig. 3. Compared to traditional IT environments, Device Drivers have increased safety and resilience requirements, because they must also comply with the requirements of operational environments. Operation safety and availability have higher priority requirements in OT environments, due to the potential damage of an incident and the high costs of operation downtimes. This means that trustworthy Device Drivers need to implement safety and resiliency mechanisms to guarantee operations. However, traditional OT environments are rather isolated silos where physical separation and network isolation protect sensitive or vulnerable components and therefore security or privacy have lower priorities compared to IT systems. Since Device Drivers bridge the operational environment with the IT environment, this separation no longer exists and it is necessary to implement mechanisms to ensure system trustworthiness. Therefore, in Device Drivers, all these requirements converge and it is necessary to implement mechanisms to support all system trustworthiness characteristics.

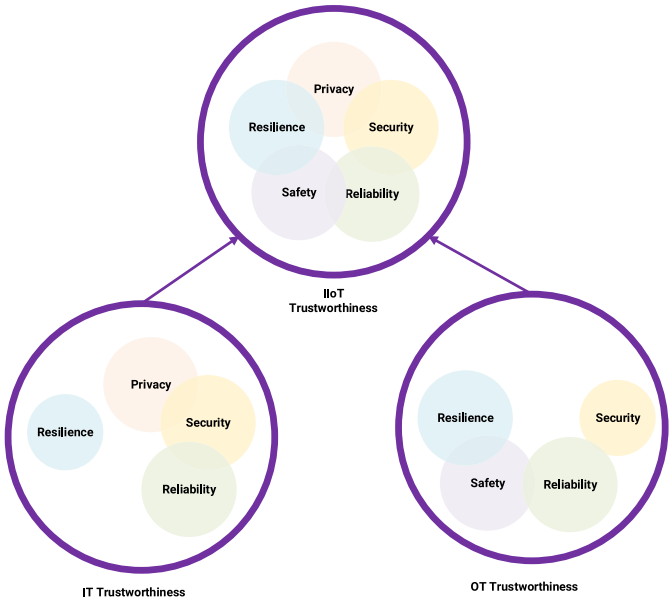


Fig. 3. Device driver trustworthiness.

IIoT system trustworthiness involves a range of activities and technologies that encompass the entire lifecycle of its components. These activities are guided by state-of-the-art cybersecurity frameworks, which are mostly based on international standards and known best-practices proven effective in operational environments. The industrial Internet consortium cybersecurity framework [8], the National Institute of Standards and Technology (NIST) framework for infrastructure cybersecurity [9], and the European Union Agency for Network and Information Security baseline security recommendations for IoT are examples of cybersecurity frameworks for IIoT systems. On the other hand, [11] and [12] provide exhaustive descriptions of the state-of-the-art in standardization and certification for Industry 4.0 and IIoT systems, including an analysis of the maturity and the relationships between these standards. Furthermore, [11] describes and compares different categories of standards relevant in the context of interoperable digital manufacturing platforms: standards for Industry 4.0 and ICS like ISA/IEC 62443 [13], standards for secure software development like the OWASP [14] and standards and schemes for IoT vendors, like the OWASP Internet of Things project [18].

The vf-OS holistic security and privacy concept response builds on these standards to respond to the main threats in the interoperable digital manufacturing platform ecosystem and implement system trustworthiness. To present the vf-OS security response for Device Drivers, Section II-A first presents the threat model and then Section II-B presents the security architecture of Device Drivers.

A. Threat Model

A threat model is an engineering technique to identify the threats, attacks, vulnerabilities, and countermeasures that affect the targeted system. The threat model used in this paper is the spoofing, tampering, repudiation, information disclosure, elevation of privilege (STRIDE) threat model developed by Microsoft to evaluate potential security risks [15]. Within the vf-OS system architecture, Device Driver attackers can be roughly categorized into two groups. The first group is people who have legitimate access rights to the vf-OS Platform (i.e., software developers, manufacturing providers, or manufacturing users). This user group is called insiders. The second group contains malicious third parties that do not participate in the vf-OS platform, otherwise known as outsiders. The following list applies the STRIDE threat model to Device Drivers from the manufacturing users' standpoint and analyzes the main potential threats, including example attack scenarios.

1) *Spoofing Identity:* Spoofing is the illegal access and use of authentication information, such as username and password. If an insider gets legitimate credentials to Device Driver services by some means, the insider may be able to operate the physical device in an illegitimate way, or steal sensitive operational data on premise. Likewise, an outsider can obtain the credentials (e.g., via phishing) and attempt to operate the Device Driver in the vf-OS platform via the Internet. Hajime is an example of this threat [16].
2) *Data Tampering:* Data tampering consists of malicious data modifications, both unauthorized changes to data at rest (persistent data) and alterations to data in transit (data as it flows between two computers over an open network, such as the Internet). The major data tampering threat for data at rest comes from insiders. Software developers and manufacturing providers could embed malicious code in vApps or Device Drivers in the development stage or over a version upgrade. Manufacturing users could make modifications that can compromise the operation of the device. Other software marketplace platforms are vulnerable to this threat [17]. Manufacturing providers can also tamper the firmware of the control devices, for instance during maintenance operations. Outsiders may alter data in transit in the connection to the vf-OS platform (through the top interfaces), but also in the connection to the physical device.
3) *Repudiation:* Repudiation happens when the system is not able to adequately track users' actions. As a result, authorized users can be denied the right to perform authorized actions without the means to prove so otherwise or malicious users can log their actions on behalf of others. Nonrepudiation refers to the system's ability to counter repudiation threats. In the context of vf-OS, the main repudiation threat is related to legitimate changes to configuration and logging data. For instance, a manufacturing user finds out that the configuration for a specific sensor is deleted and blames the software developer who had just performed an upgrade. However, the software developer states that the upgrade had no effect on the sensor's configuration data. In this situation, the system needs evidence that confirms what actually happened, so that there is no conflict between both stakeholders. For instance, log injection attacks represent a repudiation threat [18].

4) *Information Disclosure:* Exposing information to individuals who are not supposed to have access to it. Multiple manufacturing users are connected to the vf-OS platform. They do not know the internal vf-OS platform configuration, but it certainly shares resources with other companies. Sensitive information may be leaked to providers, customers, or even competitors. The target data breach is an example of this threat [19].
5) *Denial of Service (DoS):* DoS represents the threat of attackers to exploit the limited capacity of a system component to respond to unauthorized access requests. According to the cloud security alliance, distributed DoS (DDoS) is one of the top nine threats to cloud computing environments [16]. An outsider may do DoS or DDoS attacks to a device driver to disable it. Insiders may perform DoS attacks as well, but this kind of attacks may be easier to prevent. Permanent denial of service is another DoS attack that targets unsecured IIoT devices [20].
6) *Elevation of Privilege:* Elevation of privilege refers to the ability of an unprivileged user to gain privileged access and compromise or destroy the entire system. Insiders represent the main elevation of privilege threat, which stems from users with legitimate access rights who make changes to the configuration that compromise the system.

Lastly, it is important to bear in mind that the threat model cannot foresee all possible threats and thus, all systems are vulnerable to zero day attacks, which are cyberattacks that exploit security holes or vulnerabilities that were not foreseen during the design of the system security architecture. For instance, the Stuxnet attack exploited four different zero day vulnerabilities on nuclear plants in Iran [21]. Therefore, and as in this example, new vulnerabilities may be used and existing security measures may prove ineffective.

B. Device Driver Security Architecture

As a response to the different threats in the ecosystem, the vf-OS holistic security and privacy concept architecture provides a layered security approach to reduce the vulnerability surface of system components. As explained above, possible use scenarios of Device Drivers have strict requirements for system trustworthiness. Therefore, organizations need a comprehensive security response with effective countermeasures to secure the system. Table I summarizes the vf-OS response objectives for each system layer. Based on NIST guidelines [9], security measures need to be implemented into four phases for the life cycle of the response to the attack: 1) preparation; 2) detection and analysis; 3) containment eradication and recovery; and 4) post-incident activity. The following sections describe the main techniques and processes to comply with the response requirements.

1) Security Procedures: Regarding procedures, vf-OS provides security guidelines and procedures in the preparation phase of the security measures implemented at all layers to deliver trustworthy Device Drivers.

TABLE I
vf-OS HOLISTIC AND PRIVACY CONCEPT LAYERS

Layer	Response Objectives	System Trustworthiness Characteristic
Procedures	Secure development Integrity of roots of trust Secure installation Operations integrity	All
Application	Software integrity Confidentiality	Security, Privacy
Communication	Data in transit integrity Communications integrity Communication confidentiality	Security, Privacy
Network	Network integrity Architectural availability	Security, Privacy, Reliability
Device Driver	Endpoint availability Endpoint confidentiality Data at rest integrity	Reliability, Resilience, Privacy
Device	Endpoint safety	Safety

Design and installation procedures of Device Drivers comply with cybersecurity standards from both IT and OT systems to ensure system trustworthiness. In particular, the vf-OS security and privacy concept adopts the ISA/IEC-62443 standard [13] for the development of secure industrial system components. ISA/IEC-62243 provides technical security requirements for OT system components at different levels. Currently, there are security certifications in alignment with this standard and with its adoption, vf-OS ensures that it can meet the requirements of the most challenging scenarios from a security point of view. In addition to this, vf-OS also adopts the OWASP secure Web service development practices for the development of the RESTful application programming interfaces (APIs) used by vf-OS components. vf-OS also incorporates key points of the NIST cybersecurity framework [9]. Accordingly, Device Drivers must implement the vf-OS specific system cryptographic techniques and configurations into their software.

In order to protect the integrity of the roots of trust, i.e., the components that are inherently trusted, only certificates from accepted certificate authorities can be used to issue the required digital signatures for Device Drivers to be published in the marketplace.

Regarding the secure installation of Device Drivers, the network needs to comply with the ISA/IEC 62443 secure network definitions. Manufacturing users must follow the industrial device installation procedures defined in the ISA/IEC 62443 standard to install vf-OS software on their premises. Additional security procedures and cryptographic techniques like encryption guarantee the security of the data at rest in the Device Driver. The vf-OS system needs to be compatible with the ISA/IEC 62443 secure network definitions, so that Device Drivers can be installed in factories that implement this standard.

In order to protect the integrity of operations, in the detection and analysis phase, the vf-OS security command center enables continuous security monitoring, allowing to manage the configurations of the intrusion detection systems (IDSs) and security information and event management systems

deployed in the network to detect intrusions or abnormal component behavior. The vf-OS security command center can use this information to adjust the system to prevent the perceived security threat. In this sense, Device Drivers implement security event data logging and monitoring to collect and store security related data.

2) Application Security: For Application layer security, apart from the above described secure software development procedures, Device Drivers are signed with PKI device driver certificates issued by vf-OS accepted certificate authorities, to guarantee software integrity and to prevent any altered Device Drivers from running in any vf-OS Platform instance.

The system also implements a role-based access control—attribute-based access control (RBAC-ABAC) system to restrict access to assets and data, by allowing users to define precise access rules to Device Drivers to ensure confidentiality. This way, vf-OS adopts a centralized administration of the RBAC-ABAC system in the vf-OS security command center. The main advantages of this approach are to reduce the possibility of error in Device Driver implementations, to improve the response time in case of a security breach, to enforce uniformity across multiple stakeholders and to reduce the risk of nonrepudiation.

3) Communication Security: All data communications are secured using standard security recommendations for transport layer security (TLS) and data encryption using TLS [28] based network encryption for the vf-OS PKI infrastructure. Communications integrity and confidentiality is achieved by mutual authentication between system components. The vf-OS security center provides centralized control over authentication and communication between components. Components login against the vf-OS security command center, which generates a security token based on the user credentials. The vf-OS security command center acts as a reverse security proxy and intercepts all communications between components. The permissions are checked and the connection is forwarded to the destination component only if two conditions are fulfilled. The signed application configuration data must specify that the application can use the component resources and the user must be allowed to perform the operation. This centralization ensures that all components implement the same security controls over communications.

4) Network Security: Regarding the network layer, vf-OS adopts the secured network architecture concepts of ISA/IEC-62443 to protect ICS networks. According to this strategy, the network is divided into segments or levels using Firewalls. Each system component is connected to a specific segment according to its functionality. There are levels for IT systems, called enterprise levels, and levels for operational systems, called industrial levels. Applications and systems that interconnect separate levels are managed through demilitarized zone (DMZ) segments, applying what is known as a defence in-depth strategy, which has proven to be an effective technique to ensure network integrity against cyber-attacks.

vf-OS applies the ISA/IEC-62443 principles to the integration of operational systems distributed in supply chains to guarantee architectural availability, with techniques to minimize the threat of DDoS attacks, data tampering and to increase the system reliability. In this sense, there are two main deployment scenarios for Device Drivers depending on whether or not they connect to the physical device through an Internet connection: small-scale horizontal integration scenarios where Device Drivers do not connect to physical devices through the Internet and large-scale horizontal integration scenarios where they do.

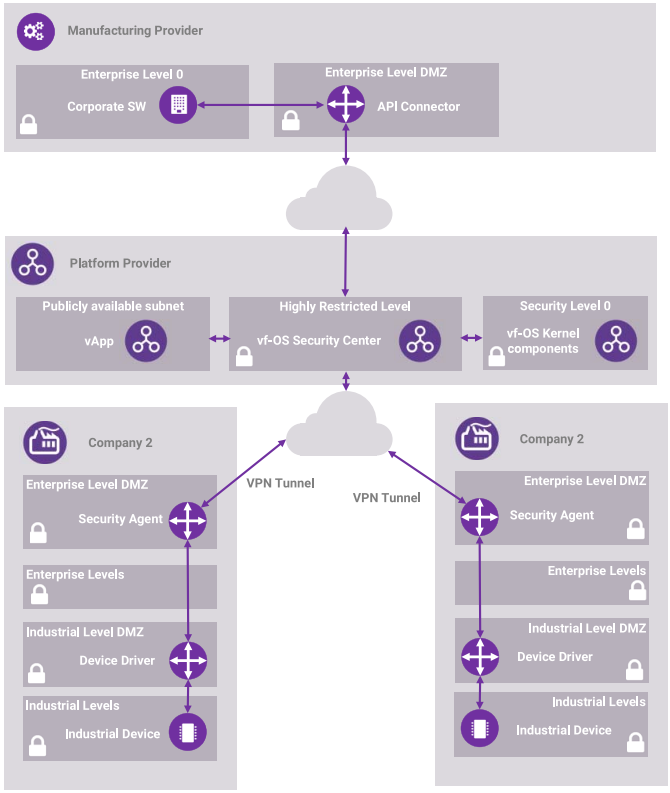


Fig. 4. Small scale horizontal integration.

To understand better the defense in-depth strategy, Fig. 4 illustrates a hypothetical example where an SME manufacturing provider—which delivers industrial devices—would like to use the vf-OS platform to provide added value services, in a new strategy to sell products-as-a-service or solution-as-a-service rather than standalone products. The vApps integrates its corporate software with the industrial devices at its customers' premises via Device Drivers. The vf-OS platform is installed on a private cloud connected to the manufacturing provider corporate network to leverage the vf-OS services and applications as an extension of their corporate software. The vApps that provide added-value services run into a publicly available subnet (DMZ), so that they are accessible to customers from the Internet.

Their customers can download from the vf-OS marketplace the required Device Driver and install it on premise. Some customers may already have a secured ICS network and therefore, they wish to install the Device Driver in the industrial level 2 (manufacturing operations) DMZ. The Device Driver interconnects with the vf-OS platform through intermediary components called security agents which are installed in the

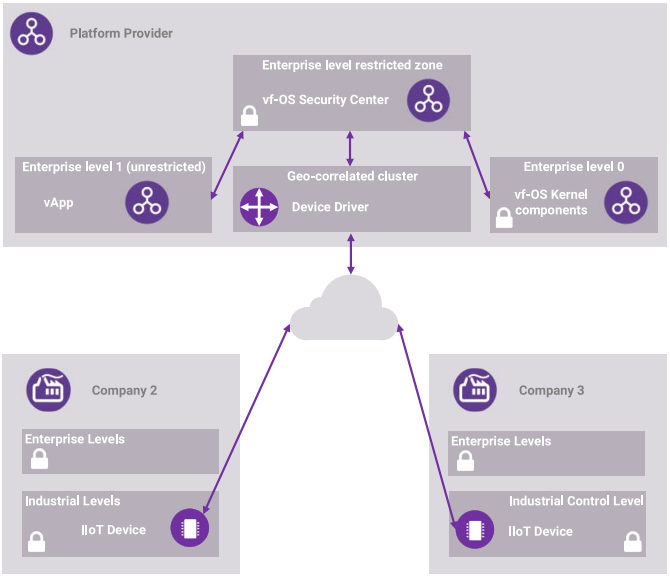


Fig. 5. Large scale horizontal integration.

DMZs between levels and basically act as security proxies. This way, the vf-OS platform can send to and receive data from the industrial device via the Device Driver following the ISA/IEC secure network architecture design principles. However, some other customers may not have a secured network architecture in place. In these cases, the Device Driver can connect to the vf-OS platform using a virtual private network (VPN) connection to seamlessly integrate the industrial device into the private network of the manufacturing provider and still provide a secured network environment. The Device Driver embeds the VPN client and configuration files to establish this connection, so that it is transparent for the end user.

In both cases, each Device Driver acts as a concentrator that gathers the data from the different industrial devices deployed in each factory. The customer is in control of the configuration of the access permissions and decides what data is shared with the manufacturing provider. Device Drivers must also implement redundancy to improve availability. This also applies to the rest of network components (like firewalls and switches).

In large scale horizontal integration scenarios, Device Drivers connect to physical devices using secured IIoT communication protocols. This is suitable when the number of devices that need to be integrated is very large and/or geographically scattered. In this scenario, Device Drivers are organized in geo-correlated clusters, implementing redundancy, and load balancing to improve availability.

5) Device Driver Security: Device Drivers can implement additional security mechanisms to improve system trustworthiness, in particular reliability and resilience.

Device Drivers can keep a short term historic of the device data in a local database. In case the Device Driver is disconnected from the vf-OS platform, it is possible to recover otherwise lost data from this short term historic, thus improving fault tolerance and reliability. In this sense, Device driver local storage is encrypted to ensure the integrity of data in rest.

Additionally, as mentioned above it is important to implement redundancy mechanisms to Device Drivers to avoid points of system failure in both deployment scenarios. Resilience can be achieved by implementing smart fallback mechanisms where the redundant Device Drivers introduce changes to the system in order to adapt to a possible threat. This smart fallback mechanism is presented in the next section.

III. Smart Fallback Mechanism

The vf-OS holistic privacy and security concept effectively provides security mechanisms at the different layers to implement system trustworthiness. However, it is not possible to foresee all possible threats when designing a system and the threat of zero day vulnerabilities always prevails. In this sense, the purpose of the security architecture described so far is to detect and protect the system against attacks. It is also important to take countermeasures on the premise that security will be broken. In both the Device Driver and Device layers in the Table I, resilience and safety must be taken into account in the design of the security architecture. More specifically, resilience means that in the event of an attack, the Device Driver must recover operations as soon as possible, while at the same time try to minimize the risk of persisting or propagating attacks. Since the Device Driver is the last gateway leading to the OT system, it is important to provide these resilient mechanisms herein.

The proposal presented in this paper is based on the implementation of fallback mechanisms for Device Drivers that are activated in case of emergency, when an IDS detects a possible attack at this layers or the Device Driver stops working normally. It is always important to keep a fallback system for emergency. The fallback is an alternative system used in case of an unplanned outage. Creating a fallback means creating a copy, therefore the normal system and fallback system have exactly the same configuration (boot files, program files, data storage, etc.). This allows an administrator to recover quickly even in case of emergency. In terms of security, however, this allows an attacker to attack the same system in the same way by using the same vulnerability or passing the same authentication. For this reason, this proposal introduces diversity in the fallback system of the Device Driver. Thus, in case of an unplanned outage or a detected intrusion, instead of restoring the exact Device Driver runtime configuration, the proposed smart fallback mechanism introduces diversity on different runtime features with two objectives. First, the changes are aimed to make it harder for the attackers to persist in the attack, assuming that the zero-day vulnerability they are exploiting does not hold for all possible runtime configurations. The second objective is to increase the level of protection of the operational system, at least momentarily until the system is back to a normal state (i.e., the attack is over). This security improvement of the fallback systems comes at the expense of decreasing to some extent the functionality of the Device Driver.

A. Runtime Configuration Diversity in Fallback Mechanism

This section describes the different options for introducing diversity in the runtime configuration of Device Drivers.

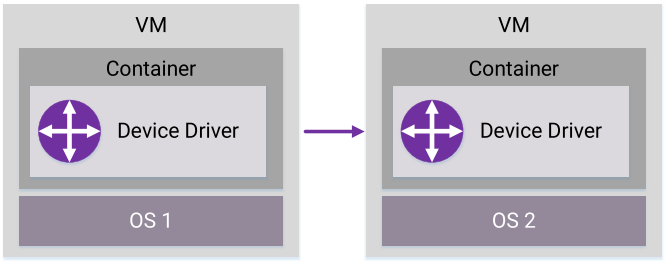


Fig. 6. OS diversity.

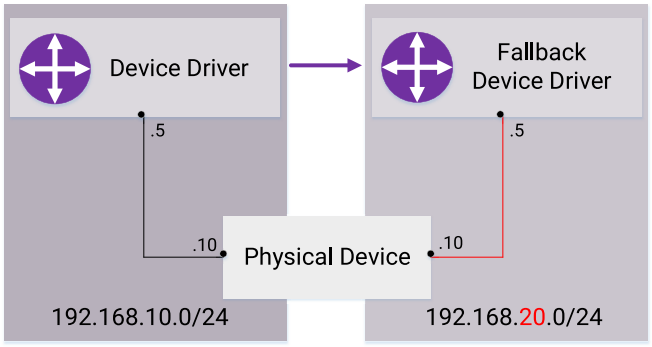


Fig. 7. Network diversity.

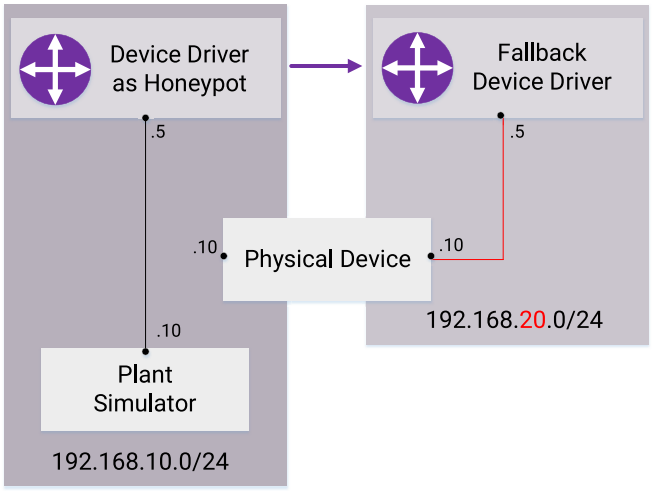


Fig. 8. Network diversity with honeypot.

The effectiveness of these fallback mechanisms have been validated through simulations. The simulation environment implements a virtual network composed of a field network level and a Device Driver network level. The Device Driver and Device Driver smart fallback prototypes used in the simulations are implemented with Node-Red [23] flows and it uses Modbus TCP [24] to communicate with field devices. The attacks are simulated using the Kali Linux [25] penetration testing tool and the Nmap security scanner [26]. To simulate the IDS system and smart fallback, a network scan in the Device Driver or the field device network segments triggers a change in the configuration of the Node-Red runtime of the Device Driver, to switch the default Device Driver Node-Red flow to the Device Driver fallback flow. The next sections describe the different smart fallback mechanisms proposed.

1) Operating System Diversity: Let us assume that an attacker managed to break all vf-OS security mechanisms and ultimately gained control of the Device Driver. The attack on the Device Driver is based on a previously unknown vulnerability of the underlying operating system (OS). Many attacks exploit OS vulnerabilities. For instance, support for Windows XP ended in April 8, 2014 [27], however, many enterprises are still using this OS in the control system without updating. This OS has the famous vulnerability called ms08_067_netapi, which allows remote code execution [28]. In this situation, making changes in the OS configuration of the Fallback Device Driver may overcome the unknown vulnerability, or at least present a new hurdle for attackers to persist in the attack. This might earn enough time to restore system security and stop the attack, since attackers cannot anticipate the OS configuration of the fallback system. Device Drivers are containerized components and, in order to implement OS diversity, they implement OS-level virtualization methods, such as clear containers [29] that isolate the OS kernel of each component. This concept is illustrated in Fig. 6.

2) Network Diversity: Network configuration of the Device Driver is another possible element that can be changed to the different configuration easily when switching to the Fallback Device Driver. The example is illustrated in Fig. 7. This is the most basic example. The normal Device Driver connects to the physical device with the network 192.168.10.0/24 and the fallback connects to it with the network 192.168.20.0/24. The Fallback Device Drive always connects to the physical device to have runtime data from physical device so that operators switch at any time in emergency. It does not connect to the top interfaces to isolate it from the threat of invasion via the top interfaces. The network 192.168.20.0/24 is normally separated from any other systems. By changing the IP address, attackers have to scan the network. This diversity itself may not be so effective against an attack, however, in the sense of having many diversity, this also makes some time to restore system security and it presents a new hurdle for attackers to persist in the attack. This is just changing IP address and therefore, it is also possible to introduce this kind of network diversity in vf-OS components quite easily.

It is also possible to have a system structure as illustrated in Fig. 8. In addition to simply switching networks like the example in Fig. 7, this system structure activates a Device Driver as a honeypot [30] which does not actually connect to the physical device. A honeypot is a system imitating a vulnerable system, which attracts attackers and bypasses possible attacks against genuine devices that are in use. It enables collecting logs of attack activities and analyzing the attack method based on the logs. This technique has been commonly practiced in IT [30].

This way, the honeypot Device Driver connects to a plant simulator in the same subnet as the normal Device Driver configuration, whereas the actual connection to the device is switched to a Fallback Device Driver like in the previous example. The plant simulator had previously used data collected from actual physical devices to emulate the actual plant. Since this simulator data may be exposed, it needs to be processed to protect sensitive information. The network

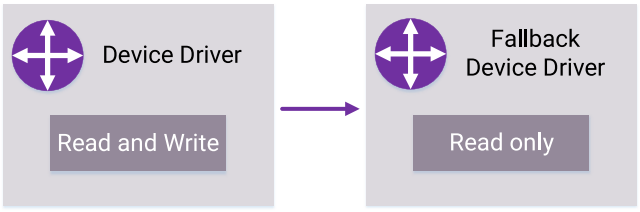


Fig. 9. Configuration diversity.

192.168.10.0/24 is completely isolated. The plant simulator has the same IP address as the physical device as if it were the genuine plant. Attackers might think the plant simulator is the actual plant and this system makes attackers stuck in the network 192.168.10.0/24. This system achieves both objectives 1 and 2.

3) Gateway Configuration Diversity: Regarding resilience, in order to recover the control systems from the cyber-attacks as soon as possible, it is necessary to prevent the physical device from being affected by the cyber-attacks. A cyber-attacker can only attack devices that are connected to the network and as discussed, network isolation is an effective security measure to protect operational systems. Therefore, isolating the target from the network is the best way to prevent the cyber-attacks to reach the operational level.

The best way to protect physical device from the threat of cyber-attacks is just pull out the cable connecting to the network. However, in the vf-OS Platform, if the physical devices are completely separated from the network, engineers cannot do remote monitoring and data cannot be sent to the vf-OS Platform. Therefore, in order to ensure the minimum productivity of the factory, authors propose the architecture illustrated in Fig. 9. The normal Device Driver allows communication from the Top Interfaces to the Bottom Interfaces as well as the opposite way. It is possible to send data from the Bottom Interfaces to the top interfaces (Read), and send an instruction from the top interfaces to the bottom interfaces (Write). This enables normal operation of vf-OS. On the contrary, the Fallback Device Driver only allows Read. When the attack is done or if something like cyber-attacks are suspected, the system switches the Device Driver to the fallback Device Driver. This makes it possible to prevent the physical devices from being affected by cyber-attacks more and secure the safety of the physical devices by not allowing attackers to write or change configurations of the controller. However, the fallback still can do Read and store the information in a short-term historic, therefore the manufacturing users can gather data and secure minimal productivity. While the Fallback Device Driver is operating, field engineers operate the factory manually.

IV. Conclusion

Industrial IoT gateway are critical components of interoperability platforms that integrate operational systems and information systems. This paper has presented the main threats and the security response in the design and implementation of Device Drivers, which are the vf-OS platform components that interact with manufacturing physical devices.

This paper has presented the different security mechanisms to realize system trustworthiness. In addition to the security mechanisms provided by the vf-OS holistic security and privacy concept, authors propose a mechanism for reinforcing resilience in the Device Driver layer to realize a system that can reduce the damage in the event of attack and can restore Device Driver operation immediately. The mechanism is based on the fallback principle: the fallback is an alternative system used when the operation of the component is compromised. The proposed smart fallback mechanism applies different diversity mechanisms to change the configuration of the fallback system in order to respond to various and unexpected situations.

Device Drivers act as a gateway connecting the vf-OS Platform and the physical device. In the event of a cyberattack targeting field devices via the network, the Device Driver is the last gateway leading to the physical devices. Therefore, when the prepared security breaks down, it is critical to build a resilient mechanism to contain the threat.

References

[1] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the industrie 4.0 working group," Nat. Acad. Sci. Eng., Forschungsunion, Frankfurt, Germany, Rep., 2013. [Online]. Available: http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf

[2] O. Vermesan, P. Friess, and P. Friess, *Internet of Things: Global Technological and Societal Trends*. Aalborg, Denmark: River, 2011, pp. 9–52.

[3] M. Marques, C. Agostinho, R. Poler, G. Zacharewicz, and R. Jardim-Gonçalves, "An architecture to support responsive production in manufacturing companies," in *Proc. IEEE 8th Int. Conf. Intell. Syst. (IS)*, Sep. 2016, pp. 40–46.

[4] C. Agostinho *et al.*, "Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology," *Comput. Ind.*, vol. 79, pp. 64–76, Jun. 2016.

[5] *Factories 4.0 and Beyond. Working Document, Recommendations for the Work Programme*, Eur. Factories Future Res. Assoc., Brussels, Belgium, 2016.

[6] *vf-OS Homepage*. Accessed: May 9, 2018. [Online]. Available: http://www.vf-os.eu

[7] A. Boza, L. Cuenca, R. Poler, and Z. Michaelides, "The interoperability force in the ERP field," *Enterprise Inf. Syst.*, vol. 9, no. 3, pp. 257–278, 2015.

[8] *Industrial Internet of Things Volume G4: Security Framework*, Ind. Internet Consortium, Needham, MA, USA, 2016.

[9] *Framework for Improving Critical Infrastructure Cybersecurity, Draft Version 1.1*, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2017.

[10] *Baseline Security Recommendations for IoT in the Context of Critical Information Infrastructures*, Eur. Union Agency Netw. Inf. Security, Heraklion, Greece, 2017.

[11] "State of the art syllabus overview of existing cybersecurity standard and certification schemes v2," Eur. Cyber Security Org., Brussels, Belgium, Rep., 2017. [Online]. Available: www.ecs-org.eu/documents/uploads/updated-sota.pdf

[12] "Status of international cybersecurity standardization for IoT (draft)," Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. NISTIR 8200, 2018.

[13] *Industrial Communication Networks—Network and System Security—Part 1-1: Terminology, Concepts and Models*, IEC Standard TS 62443-1-1, 2009.

[14] *Open Web Application Security Project Homepage*. Accessed: May 9, 2018. [Online]. Available: https://www.owasp.org/index.php/Main_Page

[15] *Microsoft Threat Modeling Tool (2016) User Guide*, Microsoft Company, Redmond, WA, USA, 2015.

[16] W. Grange. (Jan. 2017). *Hajime Worm Battles Mirai for Control of the Internet of Things*. [Online]. Available: https://www.symantec.com/connect/blogs/hajime-worm-battles-mirai-control-internet-things
[17] HP Research. (2013). *HP Research Reveals Nine Out of 10 Mobile Applications Are Vulnerable to Attack*. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=1528865#.WplohujOWUm
[18] Open Web Application Security Project. (2016). *Log Injection*. [Online]. Available: https://www.owasp.org/index.php/Log_Injection
[19] R. Vamosi. (May 2014). *IoT Hack Connected to Target Breach*. [Online]. Available: https://www.mocana.com/blog/2014/02/05/iot-hack-connected-target-breach
[20] *Industrial Control Systems Cyber Emergency Response Team Alert: BrickerBot Permanent Denial-of-Service Attack*. Accessed: May 9, 2018. [Online]. Available: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-102-01A
[21] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier," Symantec Corporation, Mountain View, CA, USA, White Paper, Security Response, vol. 5, no. 6, p. 29, 2011.
[22] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. San Diego, CA, USA: Syngress, 2014.
[23] Node-RED. *Flow Based Programming for the Internet of Things*. Accessed: May 9, 2018. [Online]. Available: https://nodered.org/
[24] Node-Red-Contrib-Modbus-TCP. *Node-RED Nodes to Communicate With Modbus TCP Servers*. Accessed: May 9, 2018. [Online]. Available: https://flows.nodered.org/node/node-red-contrib-modbustcp
[25] Kali Linux. *Penetration Testing and Ethical Hacking Linux Distribution*. Accessed: May 9, 2018. [Online]. Available: https://www.kali.org/
[26] *Nmap Security Scanner*. Accessed: May 9, 2018. [Online]. Available: https://nmap.org/
[27] J. Oh, *Fight Against 1-Day Exploits: Diffing Binaries vs Anti-Diffing Binaries*. Black Hat, 2009.
[28] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, vol. 1. Reading, U.K.: Addison-Wesley, 2001.
[29] Intel Whitepaper. *Intel Clear Containers: Building A Virtualization Continuum*. Accessed: Oct. 31, 2017. [Online]. Available: https://clearlinux.org/sites/default/files/cc-summary-v1.pdf
[30] N. Provos and T. Holz, *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. 1st ed. Upper Saddle River, NJ, USA: Addison-Wesley, 2007.

Francisco Fraile received the B.S. and M.S. degrees in telecommunication engineering from the Universidad Politècnica de València (UPV), Valencia, Spain, the M.S. degree in microwave engineering from the Högskolan I Gävle University, Gävle, Sweden, and the Ph.D. degree in telecommunication engineering from UPV, in 2013.

He is currently a Research Engineer with the Research Centre on Production Management and Engineering, UPV. His current research interests include machine-to-machine communications, industrial IoT systems, manufacturing systems, and cyber-security and networking.

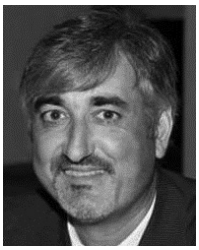
Takuya Tagawa received the B.S. degree in industrial engineering from the Nagoya Institute of Technology (NIT), Nagoya, Japan, where he is currently pursuing the master's degree.

He is an Invited Researcher with the Research Centre on Production Management and Engineering, UPV. His current research interests include cybersecurity, industrial control systems, cloud computing, and IoT systems.

Raul Poler received the Ph.D. degree in industrial engineering from the Universitat Politècnica de València (UPV), València, Spain, in 1998.

He is a Professor in operations management and operations research with the UPV, where he is the Director of the Research Centre on Production Management and Engineering. He is the Founding Partner of the spin-off UPV EXOS Solutions S.L. He is the Director of the Master in Industrial Engineering and Logistics, Alcoy Campus, UPV. He has led several Spanish Government and European Research and Development Projects. He is the Director of the INTEROP-Lab. He has authored or co-authored over 300 research papers in a number of leading journals and in several international conferences. His current research interests include enterprise modeling, collaborative networks, supply chain management, knowledge management, production planning and control, decision support systems, and evolutionary algorithms.

Dr. Poler is a member of the Executive Board of the Association for the Development of Organization Engineering. He is the Chair of the Education Activity of the IFIP WG 5.8 Enterprise Interoperability.



Angel Ortiz received the Ph.D. degree in industrial engineering from the Universitat Politècnica de València (UPV), València, Spain, in 1998.

He is a Professor in operations management, logistics, and supply chain management with UPV, where he is the Deputy Director of the Research Centre on Production Management and Engineering. He is the founding partner of the spin-off UPV EXOS Solutions S.L. He is the Director of the Master in Advance Engineering in Production, Logistics and Supply Chain, UPV. He has led several Spanish Government and European Projects. He has authored or co-authored over 250 research papers in a number of leading journals and in several international conferences. His current research interests include supply chain strategy, supply chain reference models, enterprise modeling, collaborative networks, supply chain management, business process management, and production planning and control.

Dr. Ortiz is a member of the Association for the Development of Organization Engineering.