Fighting Insider Threats, with Zero-Trust in Microservice-based, Smart Grid OT Systems

Marina Stanojevic, Darko Capko, Imre Lendak, Sebastijan Stoja, Bojan Jelacic

Faculty of technical sciences, University of Novi Sad
Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia
E-mail: marina.stanojevic@uns.ac.rs; dcapko@uns.ac.rs; lendak@uns.ac.rs; sebastijan.stoja@uns.ac.rs; bojan.jelacic@uns.ac.rs

Abstract: The Operational technology (OT) systems, utilized in critical infrastructure systems, can largely benefit from microservice-based control center architectures, by lowering upfront investment and maintenance costs. Many system operators are cautious and do not choose such modern system architectures, citing cybersecurity as a major concern. We intend to tackle that challenge and, in this paper, we investigate the threats to such mission critical systems and propose mitigation strategies aimed at lowering the likelihood of cyber-attacks. We developed a threat model focused on both external and insider threats and we group them. We utilize Microsoft's STRIDE methodology to analyze the threats on a per-service level, in a specific use case, in the smart grid sector. We propose mitigations for each threat, by putting the zero-trust principle, at the core of our proposal. We calculate the resulting risks, for each threat, based on impact and likelihood, and show that it is significantly reduced when all proposed measures are applied.

Keywords: Microservices; Security; Threat modeling; Smart Grid; Cloud computing; Zero Trust principle; Insider threats

1 Introduction

Every critical system is target to cyber-criminals and the energy industry is no exception. According to statistics, the energy industry became a prime target in the last decade [1]. We can identify several types of attackers, based on their motive for an attack. The most common motives behind corporations or even states are sabotage, where the goal is to present specific utility as unreliable which can result in losing customers and money. Another type of attacker is an individual motivated by a desire for revenge because of some personal reason or desire to prove themself as capable, to hack systems. This individual can be an insider that already has access and knowledge of the system's vulnerabilities making it easier to perform an attack.

Smart grids are considered as a critical infrastructure, whose continuous operating is mandatory. With introducing new functionalities, new vulnerabilities are introduced, as well [2]. Smart grid is consisted of two systems, information technology (IT) which is responsible for business and operational technology (OT) taking care of real-time operations. As the demand for electricity is growing, it is also becoming clear that not all current ways of energy supply are sustainable. Humanity needs to start leveraging renewable energy sources, such as solar panels or windmills. Smart grid OT system is mainly developed in monolithic and service-oriented (SOA) architecture which makes it robust and non-scalable. Traditional, monolithic solution for smart grid OT was deployed on utility-owned computing resources and security over a decade relied on physical isolation and security by obscurity.

Microservices architecture is known for over a decade [3] but is not yet leveraged in all industries. It is based on building systems as sets of multiple independent services, each running as separate processes, which communicate with each other through lightweight mechanisms. The main advantage, compared to traditional, monolithic architectures is that the services are deployed separately. Another main advantage is out of the box solution for vertical scalability. As one node in a microservice architecture is considered to be responsible for one function, if the number of requests for that function is increased, more instances of that specific service could be started to respond to the increased load. Another advantage of breaking up the monolith into multiple services is improved fault tolerance. If a single function of a monolith fails, the complete system might become unusable. For microservices that is not the case. As only the affected functions are unavailable and if there are no dependencies between them, the system can continue to provide functions that are still up and running while failure is investigated and fixed.

Deploying system such as smart grid OT in cloud environment is tempting because of all the benefits cloud brings, but there is also a big concern regarding security that should not be neglected. Although it is important to protect system on perimeters, which is mostly present in traditional monolithic smart grid OT solution as well, breaking the monolith to multiple microservices introduces new threats. Implementing a concept of zero-trust principle between microservices can reduce the risk of attacks to be perform as it is required for all users or nodes in the system to be authenticated and continuously validated while using the system features. It is also a good practice when it comes to protecting the system from insiders.

In this paper, an accent will be on building a threat model for smart grid OT systems whose architecture is built on a microservices platform [4]. After the threat model is built, we propose a novel architecture with appropriate security controls following zero-trust principle. The proposed architecture was tested and verified with Microsoft's Spoofing, Tampering, Repudiation, Information

Disclosure, Denial of Service, Elevation of Privilege (STRIDE) methodology. Risks for threat exploitation were identified and measured.

The paper consists of six sections. In the following section, related works are analyzed. Section 3 decomposes reference architecture into components and explains them together with a diagram representing application's data flow. Section 4 contains the threat model and proposes mitigations for each threat. Each component is analyzed using the STRIDE methodology and risks for threat exploitation are calculated and that is presented in Section 5. Section 6 contains our conclusions.

2 Related Work

In this section, we give an overview of the related literature on the researched fields: security in microservice-based architecture, insider threats and zero-trust model.

2.1 Security in Microservice-based Architecture

An overview on the current state of security in microservice-based systems is given in paper [5]. The authors analyzed 70 articles and gray literature on the topic and presented summed security ideas, principles, analyses, mechanisms and designs used to protect microservice-based systems. By decomposing components, the need for network connectivity is introduced and with it attack surface expanded. As microservices are designed to trust their peers, compromising one, all the others become exploited.

In [6] authors answer the question "What are the risks and how they can be addressed in an early phase or minimized after an attack?", giving the list of recommendations. As presented in literature survey [7], most studies focus on the stopping or mitigating attacks and not much on recovering from them. Intrusion detection systems can be used in container environments, as well [8]. Communication is the biggest concern when it comes to securing microservice-based architecture and authorization and authentication turned out to be the most used security mechanisms [9]. Authors present literature review and they found that mechanisms such as OAuth 2.0, OpenID Connect are used to overcome this problem. The authors of reference [10] developed a framework for establishing trust and making communication secure. Testing showed that due to communication overhead, system performance is slightly impacted.

2.2 Insider Threats

Even though 85% of damage comes from insiders [11] who can be either employee or trusted third party with appropriate privileges (e.g., cloud vendor), the most studies cover protection from attacks coming from external threats. The first step to improve defense from insiders is to establish strong security policies and constantly monitor employee activities [12]. One way to approach a better understanding of insider threat is to identify the problem space, technical and behavioral events and indicators and to analyze potential attackers and their motivation [13]. Reference [14] reports similar research with a focus on Internet-of-Things (IoT) environments. Authors of reference [15] analyzed 120 real case studies and defined attack patterns that could help in detecting insider-threats.

Insider threats are present in microservice-based systems as well especially because in deploying microservices, involvement of special governance tools is needed. In reference [16] authors identify integrity threats and define set of security requirements for microservice-based systems. They propose a framework for insider-resistant integrity protection.

2.3 Zero-trust Model

One example of a security-as-a-service solution is presented in reference [17] in which the authors introduced a flexible monitoring and policy enforcement infrastructure for network traffic. Cloud applications can leverage this solution to detect and block external and internal threats. Although shifting responsibility to others is tempting, zero-trust principle is gaining popularity among the majority of companies [18] which is quite the opposite security concept. As the word says itself, this principle is based on treating all network traffic as hostile, where it is not important if it came from inside or outside of perimeter.

Implementation of Kubernetes and Istio service mash gives out of the box solution for zero-trust model in containerized environment. Reference [19] proposes an additional set of tools whose usage can protect data that are transferred between microservices. An interesting study was done on the impact of the zero-trust model implementation on system performance [20]. Results showed that Istio reduced latency variability in responding to sequential HTTP requests and that the CPU and memory usage can be increased. Another paper on protection of security data is [21] where authors implemented new policies in the zero-trust model. An access control proxy is introduced whose task is to analyze access request, user type, device type, application type and data type. Overall strategy for establishing zero-trust model in cloud computing environment is given in [22]. As cloud environment cannot be trusted due to its dynamic and shareable landscape main challenge is to protect resources from data breaches. The authors propose implementation of trust engine, that will dynamically calculate trust

which is used later in transaction requests. Although the benefits of implementing zero-trust have been well researched, exploring the disadvantages and costs of zero-trust is neglected [23].

The division of smart grid OT system into multiple microservices gives more flexible, scalable architecture and better performance results [4]. As the proposed architecture has not been analyzed from the security point of view, that will be done in this paper. As the zero-trust model seems to be very good for mitigating insider threats, we will prove that by implementing zero-trust, risks for exploiting threats, even from insiders, is significantly reduced.

3 Reference Architecture

Smart grid OT system is considered critical, and it is expected that it provides real-time service 24 hours a day. Some of its functions is to give overview of the distribution network's health, connectivity, status of equipment on the field, etc. Potential network outage could endanger people's lives so fast detection and quick utility personnel response are crucial.

In this research, smart grid OT system is built on microservice-based architecture, deployed in cloud environment. Using this architecture gives better system performance [4] and response because if the demand is higher, the number of microservice instances can be increased. Deploying the system in the cloud environment lowers the needed upfront costs and overall cost-consumption. The main disadvantage in this approach is that new security vulnerabilities are introduced and that the system is now exposed to the public internet and to the cloud vendor.

Reference architecture used in this paper is consisted of seven services, each deployed separately in the same microservice cluster and two databases. Client application and SCADA are not considered as part of the microservice cluster as they are applications that utilities use as an access point to the services. Client application is usually deployed in control room of utility that is monitored and controlled, while SCADA is used to monitor and control devices in the field. In the following subsections, each component and system's data flow are described.

3.1 System Components

Architecture of the system is shown in Figure 1.

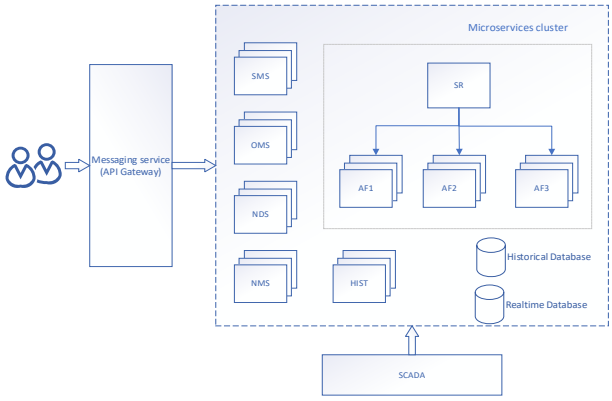

Figure 1
Reference architecture

System is consisted of the components which are divided in four groups:

1. Perimeter components – The entry point of the microservice cluster is Messaging service (MS) and its responsibility is to forward client requests to the appropriate service. Every change coming from process environment (SCADA) is processed in the Network dynamics service (NDS) and stored in RD. Communication is two-way, status of remote points can be changed through this service.
2. Core services – Network management service (NMS) is responsible for maintaining the distribution network's model and for orchestrating model updates. Outage management service (OMS) keeps track of outages in the power grid, defines recovery plans, performs automatic actions to power restoration. Service responsible to keep records of repairs and to create work orders to send crews to the field is Switching management service (SMS). Historian service (HIST) is component used as proxy to Historical database. It allows adding new entries and reading existing.
3. Services for analytic functions – Service for function requests (SR) upon receiving request from MS, divides it to multiple requests based on number of roots and forward them to appropriate analytic function. Analytic function (AF) engages algorithm for specific energetic calculation. Since calculations for multiple roots can be done in parallel, more instances of this service are used.
4. Data storage – Realtime database (RD) contains information about the current state of the distribution network. Every change in the system is preserved in Historical database (HD) database.

3.2 Inputs, Outputs and Data Flow

Data flow diagram in Figure 2 represents how data is sent through an application and its components. It is important input for any threat analysis.

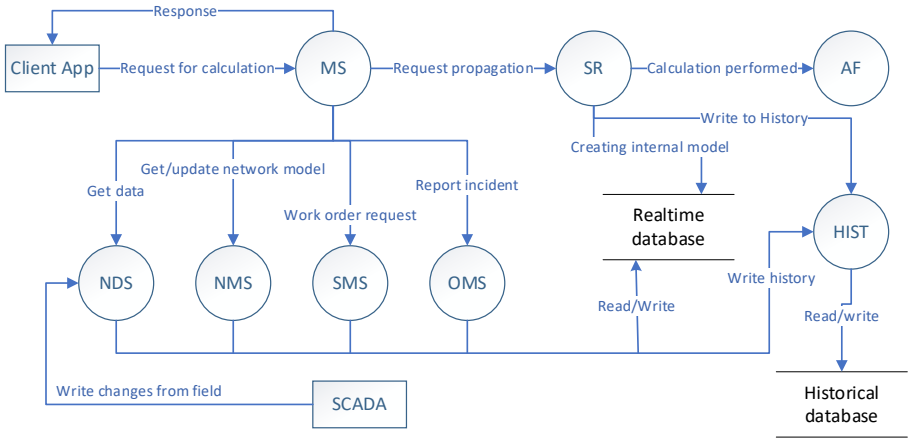

Figure 2
Data flow diagram

Each service has a list of methods it exposes to other services and to applications outside the microservice cluster (e.g., SCADA and client application). Services have read and write access to Realtime database and to HIST. SR is receiving requests from MS, obtaining information from Realtime database that are necessary for calculations and propagating request to appropriate AF. This service is also communicating with HIST and writing history to Historical database through it. Regarding client application, it is communicating with system through MS whose role is to be a mediator and to forward request to appropriate service. SCADA is communicating with NDS service directly and is sending updated status of field equipment.

4 Threat Modeling and Mitigation

First step in threat modeling is to decompose application to its components, which is easier in microservice architecture because application is already decomposed to multiple services. Every connection point between components should be observed from the attacker's point of view and analyzed how it could be exploited to gain leverage. The threat model diagram of application's data flow is presented in Figure 3. It focuses on detecting treats coming from insiders. For threat analysis, Microsoft Threat Modeling tool [24] was used. As it can be seen on data

flow diagram in Figure 2, the services NMS, NDS, MDM, WOM and OMS have the same flows, so they are modeled as one Service in Figure 3, to simplify diagram.

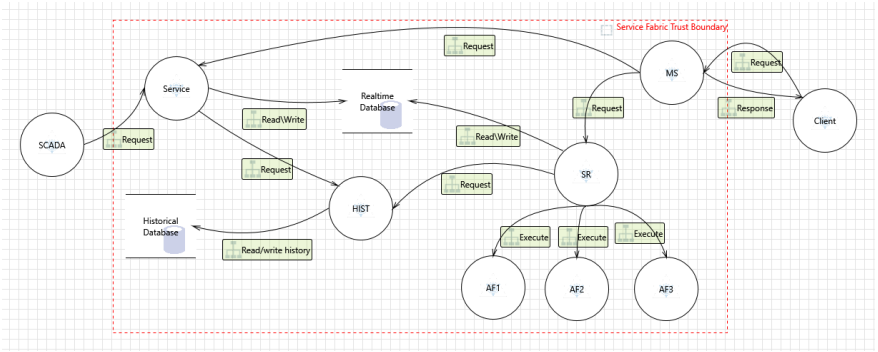

Figure 3
Threat modeling diagram

After making the diagram, presented in Figure 3, a report was generated which gave overview on threats between each component and proposed mitigation. After analyzing report, we came to conclusion that threats for all services are the same. Proposed mitigation list is given in the following subsections and it is necessary to apply them to all services in the system. Figure 4 contains graphical representation of proposed mitigation list.

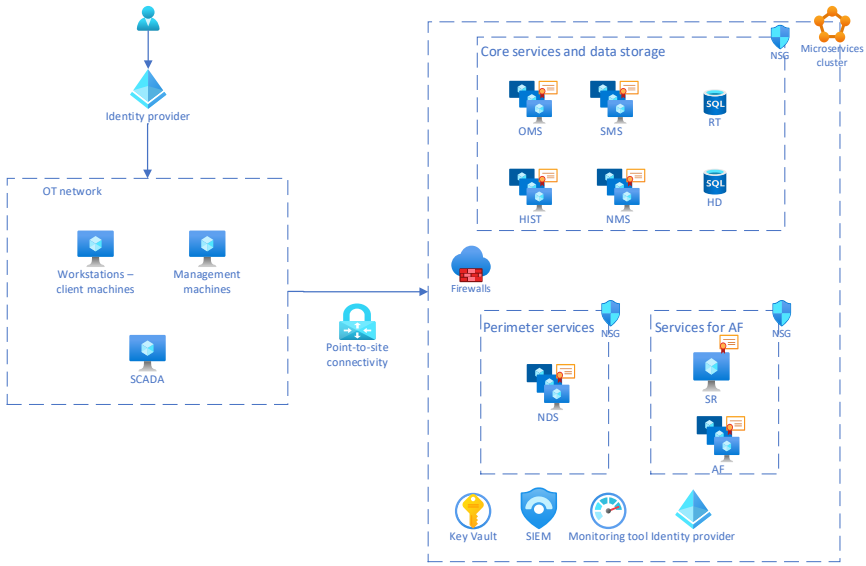

Figure 4
Proposed mitigation

4.1 Establishing Strict Business Rules

Appropriate periodical security education and repetitive psychological (background) checks are essential for all employees. To grant admin rights to the user, an additional paperwork should be requested, such as that he has additional responsibility and will be liable in court if his account is used for malicious activities. Workstations must be in physically isolated room that is constantly monitored and access to this room should be given only to employees with privileges. External devices, such as cameras, USB or hard drives are not allowed. This way, possibility of stealing information is significantly decreased. System deployment and update is considered critical and will be supervised and approved by personnel with highest privileges. Adding new nodes to Service Fabric cluster outside the deployment procedure is prohibited.

4.2 Authentication

The first step is to restrict anonymous access to the Service Fabric cluster. That can be done with implementing authentication process, where users must prove that they are who they claim to be. To secure client-to-node access, identity provider is used to which users must authenticate before gaining access privilege to application. Besides requesting strong passwords from clients through policy, multi-Factor authentication (MFA) must be applied. This way the adversary can't login even if he steals user's password.

To secure system from insider threats, implementing zero-trust principle is required so nodes authentication is also needed. Recommendation is to use certificates (e.g., Cluster X.509). This will prevent nodes that are not certified to join Service Fabric cluster and to gain unauthorized access to other nodes. It must be ensured that Service Fabric client-to-node certificate is different from node-to-node certificate. Service Fabric certificates should be obtained from an approved Certificate Authority (CA) and self-signed or test certificates cannot be allowed in production. The same principle applies when it comes to scaling up, each service instance must have a valid certificate.

4.3 Authorization

There are different system use cases, client access to cluster, access to cluster for deployment purpose, admin access and access from SCADA to update status of equipment. To implement zero-trust principle, it is not enough to analyze and protect only system perimeters so authorization methods will be applied to each microservice individually as they all interact with each other. Role-based access control (RBAC) must be implemented by following defense in depth principles. For each service, there is a separate group of privileges assigned and defined for

each user what privileges he has. Recommendation is that only administrator account has full access to management capabilities. After establishing RBAC for client's access, authorization controls will be enabled, to prevent user from achieving privilege escalation. This will enable verification of caller's permissions and establishing whether he has or does not have enough privilege to execute method on the server.

Access Control Lists (ACL) must be implemented on files (e.g., XML files) and prevent their unauthorized manipulation. This is especially important in between nodes communication when service is trying to manipulate with data of other service and change its state. When updating critical configuration, additional check for admins must be applied, for example to repeat MFA.

4.4 Protecting Secrets

Binaries which contain sensitive information (e.g., code for topology calculation) must be obfuscated. Sensitive data stored by services (e.g., internal model created by service for requests) will be encrypted and disk-level encryption will be used on nodes in the cluster. Message security protection level will be set to encrypt and sign. This way even if an adversary intercepts a message, it will be encrypted and unusable. A key vault will be used as a solution which has its own authentication and authorization for certificate and key storage. Important condition is that the data should not be seen or extracted by anyone without rights (e.g., vendor) and that all secrets are guarded using industry algorithms and appropriate key lengths.

4.5 Network Security

To additionally harden network security, firewall that will allow network traffic only from specific IP addresses and ports will be deployed. Regarding communication between services, throttling will be enabled. Network segregation is important to separate services that have communication with external systems (NDS and SR) from the rest of the system. Each network group have defined inbound and outbound network rules, which is another layer of communication restriction between services. Preventing over-consumption of resources by limiting concurrent calls, instances or sessions is also a good practice.

Any machine from the cluster shouldn't be exposed publicly or have public endpoints. Here we propose that for accessing machines that are in the Service Fabric cluster, the company set up point-to-site VPN and dedicate one host machine in corporate network to be used for remote connections for administration purposes. Access to that host machine, must be restricted only to personnel, with high authority, whose job is to run the business. Introducing point-to-site VPN, additional level of authentication is required. To access machine in

cluster, user must authenticate to corporate network and to identity provider whether through installing certificate on machine or providing credentials. Point-to-site VPN will be used also for client access, each client will have dedicated workstation on which client application is installed. Same applies to communication between SCADA which is on machine from OT network.

4.6 Database Security

Database access will be configured with roles and least-privileged accounts will be used to connect to the database. Regarding service login, instead of direct access to the tables, which must be forbidden, there is a list of selected stored procedures which are allowed to be executed by each service. Members of the database admin server role will be very limited and never contain accounts used by services. Firewall is configured for Database Engine Access. Login auditing will be enabled on DB Server. Digital signature will be added to critical database securables. Strong encryption algorithms must be used to encrypt data in the database. In microservice architecture, good practice is that every service has its own database so that risk of interfering with each other's data is not possible. If that is not possible, Row Level Security (RLS) must be applied. RLS enables implementation of restrictions on data row access. For example, ensuring that services can access only those data rows that are pertinent to their scope.

4.7 Logging and Monitoring

Proper logging of all security events and user actions builds traceability in a system and denies any possible repudiation issues. Logging successful and failed authentication attempts must be enabled. Besides that, application has its own logs and audits every request. These log files are considered sensitive information and are protected from unauthorized access by restricting view/write privileges only to administrators who will do any necessary inspections. Another good practice is to disable deletion of these files, they are archived periodically instead.

Monitoring metrics can be useful in detecting system's anomalous behavior. If we take denial of service attack as an example, monitoring system could detect it at the beginning of an attack, because number of requests is increased, and the system would start increasing the number of service instances. A sudden increase in the number of instances could trigger a rule which would create incident. An external monitoring tool is recommended, because it would reduce the need to connect and monitor directly. When it comes to what should be monitored, besides number of requests and number of service instances, successful/unsuccessful logins to the identity provider, key vault accesses, request solve duration, CPU/memory usage for each machine, traffic between system components, accesses to database could be helpful. A Security Information and

Event Management (SIEM) solution is recommended tool which in addition brings threat detection by analyzing collected security events. According to the zero-trust principle, access to these tools is restricted only to admins.

4.8 Protecting from Outside

Special attention is needed for field devices that communicate with the system through the SCADA. Although the SCADA is outside of the microservice cluster and its security is out of scope for this paper, some measures need to be taken. The NDS service is tracking frequency of value changes and if some field device is sending more changes than usual, incident is reported with high priority. SIEM can be used as a detection tool in this scenario. On the other hand, there is whitelist for clients who have access to the system. MS detects and forbids clients that act suspicious and are sending a lot of requests in short time.

4.9 Cost of Novel Architecture

It must be noted that deploying the proposed system architecture in a smart grid OT environment certainly introduces costs and impacts performance. Firewall, VPN tunnel, key vault and SIEM are tools that can be quite resource-intensive, depending on cloud vendor. Controls like authorization and encryption lead to higher CPU utilization which can be resolved with upgraded hardware or running additional micro-service instances.

5 Component Level STRIDE Analysis

Keeping in mind the nature of microservice architecture (e.g., components can change context or accessibility rapidly) and the fact that the attacker could be an insider, besides investigating threats for connection points on microservices trust boundary, STRIDE analysis for each component is also needed. That way, it can be determined how attacker can target each component. In this section, each component service is analyzed individually using the STRIDE methodology [25]. Risk for each threat is calculated following the Federal Information Processing (FISP) [26] and the European Network and Information Security Agency (ENISA) [27] standard, as shown on Table 1 and based on impact and likelihood. Impact, presented in Table 2, determines how negatively the exploitation of the threat would affect the business and clients. On the other hand, likelihood (Table 3) stands for the probability of the threat's exploitation by attackers.

Table 1
Risk matrix

	Impact					
Likelihood		Very high	High	Moderate	Low	Very low
	Very high	Very high	High	Moderate	Low	Very low
	High	Very high	High	Moderate	Low	Very low
	Moderate	High	Moderate	Moderate	Low	Very low
	Low	Moderate	Low	Low	Low	Very low
	Very low	Low	Low	Very low	Very low	Very low

Table 2
Impact level description

Impact	Description
Very high	Expected severe or catastrophic adverse effects on operation, assets, individuals: • Serious personnel injuries or loss of lives. • Long-lasting system unavailability and unusability resulting in blackouts. • Valuable asset destruction.
High	Any action which could lead to losing customers due to system unreliability and major financial loss: • The system loses capability to perform one or more of its primary functions (or connectivity with SCADA) and gives wrong calculations. • Leakage of secret customer information. • Expensive equipment damage.
Moderate	Serious adverse effect is expected: • Causing a significant degradation in the effectiveness of the system functions. • Losing customer trust. • Revealing information about system that can be used by competition (or attacker) to gain an advantage. • Unavailability of non-critical system components.
Low	Limited adverse effect is expected: • Minor damage to assets. • Financial loss.
Very low	Negligible adverse effects are expected which do not affect system performance.

Table 3
Likelihood level description

Likelihood	Description
Very high	An adversary is almost certain to initiate the threat event. This means that the system has serious security flaws that can be exploited, for example, if system is publicly available and anyone can use its functions or change data.
High	An adversary is highly likely to initiate the threat event. The system architecture has weak spots that an experienced adversary can leverage. Employee with higher privileges than necessary could be weak spot, either malicious (insider threat) or uneducated and tricked by an adversary.
Moderate	An adversary is somewhat likely to initiate the treat event. With great effort the attacker can obtain limited access to the system but still cannot endanger the system, so his motivation is low.
Low	An adversary is unlikely to initiate the threat event. Employees are loyal and well educated so the insider threats are not likely to happen. The system is protected on its perimeters, so if it comes to breach, attacker could not reach any sensitive information.
Very low	An adversary is highly unlikely to initiate the threat event. The system is protected from every aspect and could not be penetrated.

In the following subsections, analysis for each STRIDE threat (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) is given.

5.1 Spoofing

In spoofing communication from an unknown source is masked so it seems like it is coming from a known source. Service Fabric cluster does not have public endpoints and is not accessible from the public internet, it is reachable only through point-to-site VPN, authorized workstations that are in the utilities control center. The weakest link are employees who an adversary can potentially take advantage of using phishing, getting them to install malicious software or reveal secrets (e.g., their password). Another potential breach is to spoof node in microservice cluster by using stolen certificates. Outcome is same, an adversary could degrade the integrity of the system or reveal insight in system's operations.

If an adversary gains control over the NDS for example, he can send commands to field and leave catastrophic consequences like outages and blackouts. In the Table 2, this is marked as High impact.

This threat could not be exploited without getting insider's help because the workstations from which the system could be accessed are physically isolated and protected. Even if an attacker steals a password, he would need to pass MFA. Another scenario for attacker is to deploy additional node in cluster that could act like real service and send malicious data to other services. Because of strong system deployment rules, the presence of more people with different privileges is

required to add new node, meaning that one malicious insider could not perform this operation alone even with stolen certificates. Likelihood for this threat is Low, as it is unlikely that the adversary can initiate threat event (Table 3).

According to Risk matrix presented in Table 1, risk for this threat is ***Low***.

5.2 Tampering

This threat stands for changing of data (destruction, manipulation or alteration) through unauthorized channels. Communication between all system components is encrypted and protected so intercepting or tampering messages between services is worthless because without encryption key, those messages are meaningless to the adversary. Regarding databases, they are protected with RBAC, other files (e.g., log files) with ACL.

Impact is High because with this threat exploited attacker could gain insight in confidential client's information (Table 2). Compromising data from DMS or Historical database (e.g., attacker manipulates equipment's state) would result in wrong function calculation and would give false power grid status.

Likelihood for this threat exploit is Low according to Table 3 after applying proposed mitigation even if attacker is insider because communication and disks are encrypted and data is protected from unauthorized access. An adversary could not reach any sensitive information because each service implements appropriate authentication and authorization methods, encrypts messages and data.

Risk for this threat is ***Low***.

5.3 Repudiation

If the system lacks traceability, it is difficult to identify attacker who would perform malicious operations in the system. In the absence of proper auditing and logging controls, it would become impossible to implement any accountability in a system. There must be logging of all security events, user actions (e.g., successful and failed authentication), each service logs relevant activity. Constant monitoring of system is recommended through internal or external tools, like SIEM, which has automatic rules for incident reporting if suspicious system behavior or access occurred.

Impact for this threat is Very low because repudiation itself could not harm the system (Table 2). If this threat is exploited, tracing attacker is impossible and it is unknown if or when attack will be repeated.

Likelihood is Very low (Table 3). With implementation of logging and monitoring, set of events are defined whose processing can lead to instant responsible personnel informing if something is violated or the system is

breached. Also, log files are protected from deletion and unauthorized access, so even insider could not make them inaccessible if analysis after the breach is needed.

Risk for this threat is ***Very low***.

5.4 Information Disclosure

Secrets can be any sensitive information, such as connection strings, passwords, state of equipment in the field, etc. Algorithm for DMS functions is also considered sensitive business logic.

Impact is High as attackers could perform sabotage to make clients lose trust in utility as their electricity distributor and to switch to the competitor (Table 2). Sabotage could involve equipment destruction, topology change, disclosure of client confidential information, blackmail, etc. DMS function algorithm is important from the system performance aspect. With better algorithm, topology and load flow analysis gives more accurate results faster. For example, load flow results can be used to predict energy consumption in order to optimize the consumption demand ratio so having a better DMS function algorithm gives a competitive advantage. It is considered confidential business logic because it is developed or improved by the company's power engineers.

Likelihood for this threat is Low (Table 3). As with appropriate authentication and authorization implemented, all secrets are protected with RBAC and encryption, including databases. Keys and certificates are stored in key vault which brings additional layer of protection. In order for client application to gain access to the system it must be on whitelist which is monitored and controlled on daily basis.

Risk for this threat is ***Low***.

5.5 Denial of Service

If adversary launches Distributed Denial of Service (DDoS) attack he could provoke shutting down a service or network so that it is inaccessible to other clients. Besides service unavailability, increased load on the system could result in vertical machine scaling which can significantly increase cost.

Impact is High because if system is unavailable, utility is cut off from its consumers and unable to react to any potential outage (Table 2). Major financial loss is also possible because of larger number of service instances engagement than needed.

Likelihood for this threat is Low (Table 3) because with SIEM any anomalous behavior coming from user or device is detected and requests coming from that client are no longer processed because that client is taken of the whitelist. On the

other hand, the real threat is not malicious user, but endurance of the system. Employees can overload the system so that it becomes inaccessible. Similar situation can happen in case of storm during which a significant number of changes in the field occur in a short time. The system responds in these scenarios by increasing the number of instances so it will not become unusable.

Risk for this threat is ***Low***.

5.6 Elevation of Privilege

The attacker receives an account with fewer access rights and manages to obtain privileges with higher rights. Implementation of strong authentication methods relying on MFA, RBAC and ACL lowers the possibility for this threat to be exploited.

Reason for High impact (Table 2) is that if the attacker gets greater rights, he can create incidents in distribution network that will cause blackouts, gain insight to secret customer information, sabotage field crews, etc.

Likelihood is Very low (Table 3), because each service is protected with RBAC and user privileges are checked at each step. Users are required to authenticate with MFA before using any system component. List of users and user groups are visible only to administrators and any change that involves changing access to any resource requires administrator to re-authenticate.

Risk for this threat is ***Low***.

Conclusions

Securing mission critical systems, has always been a challenge and in most cases, a showstopper for shifting them to the cloud environment. The utilization of microservice-based architectures introduces multiple benefits regarding operational technology (OT) system performance. In addition, when deploying such system in the cloud environment, upfront investment and maintenance costs are lowered. The goal of this research was to analyze a novel microservice-based architecture for OT systems from the security point of view and to propose mitigation strategy which will lower the likelihood of threat exploitation. During our research of zero-trust principle, we came to the conclusion that security should be embedded in system architecture in the system design phase. Keeping that in mind, we developed a threat model, investigated and grouped external and insider threats, and proposed a novel architecture with appropriate mitigations for each threat.

First, the services were divided into three groups: core services and data storage, services for analytic functions and perimeter services. The communication networks groups serving these groups of services need to be segregated, where inbound and outbound rules are defined and message security protection level set

to encrypt and sign. All communication with outside systems should go through encrypted, point-to-site VPN tunnel and firewall. All data should be encrypted as well, including databases and data disks. For sensitive data storage, such as encryption keys or certificates, key vault is a suitable solution. A cloud-based identity provider needs to be deployed and used to authenticate users before granting system access. Anonymous access to the system is forbidden, strong password and multi-factor authentication (MFA) is necessary for all users. For node authentication, certificates are used. Role-based access control (RBAC) is implemented following defense in depth principles and authorization controls enabled. Activity logging and monitoring need to be planned carefully and appropriate alerts need to be defined to maximize the likelihood of early (cyber) attack detection.

Microsoft's STRIDE methodology was used to analysis the above-described smart grid OT architecture. The service-level risk analysis showed that the likelihood for threats exploitation is significantly reduced.

Future work will encompass comparing the proposed solution herein, with and without, the listed security controls and the calculation the exact overhead introduced, within different cloud environments.

References

[1] Macola, Ilaria Grasso: The five worst cyberattacks against the power industry since 2014, Power-Technology.com, April 2020

[2] D. Rosic, I. Lendak, and S. Vukmirovic, A Role-Based Access Control Supporting Regional Division in Smart Grid System, Acta Polytechnica Hungarica, Vol. 12, No. 7, pp. 237-250, 2015

[3] Dragoni, Nicola, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina., Microservices: yesterday, today, and tomorrow, Present and ulterior software engineering, pp. 195-216, 2017

[4] S. Stoja, S. Vukmirovic, N. Dalcekovic, D. Capko, B. Jelacic, Accelerating Performance in Critical Topology Analysis of Distribution Management System Process by Switching from Monolithic to Microservices, Rev. Roum 63, No. 3 (2018), pp: 338-343, 2018

[5] A. Pereira-Vale, E. B. Fernandez, R. Monge, H. Astudillo, G. Márquez, Security in microservice-based systems: A Multivocal literature review, Computers & Security, Vol. 103, April 2021

[6] N. Mateus-Coelho, M. Cruz-Cunha, L. Gonzaga Ferreira, Security in Microservices Architectures, Procedia Computer Science, Vol. 181, pp. 1225-1236, 2021

[7] Pereira-Vale A, Márquez G, Astudillo H, Fernandez EB, Security Mechanisms Used in Microservices-based Systems: A Systematic

Mapping, XLV Latin American Computing Conference (CLEI) IEEE, pp. 01-10, 2019

[8] Flora, José, Improving the security of microservice systems by detecting and tolerating intrusions, 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 131-134, IEEE, 2020

[9] de Almeida, Murilo Góes, and Edna Dias Canedo, Authentication and Authorization in Microservices Architecture: A Systematic Literature Review, Applied Sciences 12, No. 6 (2022): 3023, 2022

[10] T. Yarygina and A. H. Bagge, Overcoming Security Challenges in Microservice Architectures, 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), pp. 11-20, 2018

[11] Z. M. Yusop and J. H. Abawajy, Analysis of insiders attack mitigation strategies, Procedia-Social and Behavioral Sciences, Vol. 129, pp. 581-591, 2014

[12] Saxena, Neetesh, Emma Hayes, Elisa Bertino, Patrick Ojo, Kim-Kwang Raymond Choo, and Pete Burnap, Impact and key challenges of insider threats on organizations and critical businesses, Electronics 9, No. 9 (2020): 1460, 2020

[13] Nurse, Jason RC, Oliver Buckley, Philip A. Legg, Michael Goldsmith, Sadie Creese, Gordon RT Wright, and Monica Whitty, Understanding insider threat: A framework for characterising attacks, 2014 IEEE security and privacy workshops, pp. 214-228, IEEE, 2014

[14] Kim, Aram, Junhyoung Oh, Jinho Ryu, and Kyungho Lee, A review of insider threat detection approaches with IoT perspective, IEEE Access 8, pp. 78847-78867, 2020

[15] Agrafiotis, Ioannis, Jason RC Nurse, Oliver Buckley, Phil Legg, Sadie Creese, and Michael Goldsmith, Identifying attack patterns for insider threat detection, Computer Fraud & Security 2015, No. 7, pp. 9-17, 2015

[16] Ahmadvand, Mohsen, Alexander Pretschner, Keith Ball, and Daniel Eyring, Integrity protection against insiders in microservice-based infrastructures: From threats to a security framework, Federation of International Conferences on Software Technologies: Applications and Foundations, pp. 573-588, Springer, Cham, 2018

[17] Y. Sun, S. Nanda, and T. Jaeger, Security-as-a-service for microservices-based cloud applications, 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 50-57, IEEE, 2015

[18] Buck, Christoph, Christian Olenberger, André Schweizer, Fabiane Völter, and Torsten Eymann, Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust, Computers & Security 110 (2021): 102436, 2021

[19] de Weever, Catherine, and Marios Andreou, Zero trust network security model in containerized environments, University of Amsterdam: Amsterdam, The Netherlands, 2020

[20] S. Rodigari, D. O'Shea, P. McCarthy, M. McCarry, S. McSweeney, Performance Analysis of Zero-Trust multi-cloud, 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), pp. 730-732, IEEE, 2021

[21] I. Ahmed, T. Nahar, S. Sultana Urmi, K. Abu Taher, Protection of sensitive data in zero trust model, Proceedings of the International Conference on Computing Advancements, pp. 1-5, 2020

[22] Mehraj, Saima, and M. Tariq Banday, Establishing a zero trust strategy in cloud computing environment, 2020 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6, IEEE, 2020

[23] C. Buck, C. Olenberger, A. Schweizer, F. Völter, and T. Eymann, Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust, Computers & Security 110, 102436, 2021

[24] Microsoft MSDN documentation, the Threat Modeling tool https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-mitigations

[25] Microsoft MSDN documentation, the STRIDE Threat Model. https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

[26] New Standards for Security Categorization of Federal Information and Information Systems, NIST Federal Information Processing Standard (FIPS) 199, Feb. 2015

[27] European Network and Information Security Agency (ENISA), Annex II, Security aspects of the Smart Grid, https://www.enisa.europa.eu/