UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Lauri Haverinen

CYBER SECURITY OF SMART BUILDING ECOSYSTEMS

Master's Thesis
Degree Programme in Computer Science and Engineering
August 2022

Haverinen L. (2022) Cyber Security of Smart Building Ecosystems. University of Oulu, Degree Programme in Computer Science and Engineering, 74 p.

ABSTRACT

Building automation systems are used to create energy-efficient and customisable commercial and residential buildings. During the last two decades, these systems have become more and more interconnected to reduce expenses and expand their capabilities by allowing vendors to perform maintenance and by letting building users to control the machines remotely. This interconnectivity has brought new opportunities on how building data can be collected and put to use, but it has also increased the attack surface of smart buildings by introducing security challenges that need to be addressed. Traditional building automation systems with their proprietary communication protocols and interfaces are giving way to interoperable systems utilising open technologies. This interoperability is an important aspect in streamlining the data collection process by ensuring that different components of the environment are able to exchange information and operate in a coordinated manner. Turning these opportunities into actual products and platforms requires multi-sector collaboration and joint research projects, so that the buildings of tomorrow can become reality with as few compromises as possible. This work examines one of these experimental project platforms, KEKO ecosystem, with the focus on assessing the cyber security challenges faced by the platform by using the well-recognised MITRE ATT&CK knowledge base of adversary tactics and techniques. The assessment provides a detailed categorisation of identified challenges and recommendations on how they should be addressed. This work also presents one possible solution for improving the detection of offensive techniques targeting building automation by implementing a monitoring pipeline within the experimental platform, and a security event API that can be integrated to a remote SIEM system to increase visibility on the platform's data processing operations.

Keywords: Intelligent building, cyber-physical structures, operational network, threat intelligence, threat modeling

Haverinen L. (2022) Älyrakennusekosysteemien kyberturvallisuus. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 74 s.

TIIVISTELMÄ

Taloautomaatiojärjestelmiä käytetään energiatehokkaiden ja muokattavien rakennusten luomisessa sekä kaupallista että asuinkäyttöä varten. Viimeisten parin vuosikymmenen aikana nämä järjestelmät ovat kehittyneet yhä verkottuneemmiksi, jotta laitetoimittajat pystisivät käyttämään uusia ominaisuuksia kuten etäohjausta ylläpitokulujen laskemiseksi. Tämä verkottuneisuus on luonut uusia mahdollisuuksia rakennusten tuottaman datan keräämiseen ja hyödyntämiseen, mutta se on samalla kasvattanut laitteiden ja älyrakennusten hyökkäyspinta-alaa. Perinteiset rakennusautomaatiojärjestelmät suljettuine yhteyskäytäntöineen ja rajapintoineen ovat väistymässä avoimia teknologioita hyödyntävien yhteentoimivien järjestelmien tieltä. Tämä yhteentoimivuus on tärkeässä roolissa tiedonkeruuprosessin virtaviivaistamisessa varmistamalla että ympäristön eri osat pystyvät kommunikoimaan ja toimimaan koordinoidusti. Näiden mahdollisuuksien muuttaminen valmiiksi tuotteiksi vaatii monialaista yhteistyötä ja yhteisiä tutkimushankkeita, jotta tulevaisuuden rakennuksista tulee totta mahdollisimman vähin kompromissein. Tässä työssä tarkastellaan yhtä kokeellista hankealustaa, KEKO ekosysteemiä, kiinnittäen erityistä huomiota alustan kyberturvallisuushaasteisiin käyttäen yleisesti tunnettua MITRE ATT&CK -viitekehystä, joka sisältää tietoa vihamielisten toimijoiden käyttämistä taktiikoista ja menetelmistä. Turvallisuusarvio tarjoaa yksityiskohtaisen listauksen havaituista haasteista ja kuinka niihin voisi vastata. Työssä esitetään myös yksi vaihtoehto rakennusautomaatiojärjestelmiä kohtaan toteutettujen vihamielisten toimien havainnoinnin parantamiseen luomalla älyrakennusalustalle valvontakomponentti ja tietoturvatapahtumarajapinta, joka on mahdollista integroida SIEM-järjestelmään ja sitä kautta lisätä näkyvyyttä alustan dataprosessointiin.

Avainsanat: Älykäs rakennus, kyber-fyysinen järjestelmä, operatiivinen verkko, uhkatieto, uhkamallinnus

TABLE OF CONTENTS

FOREWORD

This work was carried out at Oulu University Secure Programming Group in collaboration with Netox Oy as a part of the Finnish-USA research collaboration Security and Software Engineering Research Center (S^{2}ERC).

Special thanks to my supervisor Teemu Tokola who tirelessly encouraged and supported me through this process.

Oulu, August 8th, 2022

Lauri Haverinen

LIST OF ABBREVIATIONS AND SYMBOLS

BAS	building automation system
IT	information technology
OT	operational technology
FMS	facility management system
BMS	building management system
EMS	energy management system
HVAC	heating, ventilation and air conditioning
AHU	air-handling unit
ATU	air terminal unit
VAV	variable air volume
ICS	industrial control system
NCS	networked control systems
SIEM	security information event management
IDS	intrusion detection system
IPS	intrusion prevention system
CPS	cyber-physical system
HMI	human-machine interface
ISO	International Standards Organization
OSI	Open System Interconnection
DPO	data protection officer
CTI	cyber threat intelligence
TTP	tactics, techniques & procedures
IOC	indicator of compromise
PACS	physical access control systems
HMI	human-machine interface
APT	advanced persistent threat
WSN	wireless sensor networks
MQTT	message queuing telemetry transport
API	application programming interface

1. INTRODUCTION

During the last couple of decades, the construction industry has been evolving at a rapid pace. The use of embedded systems in the form of building automation has increased in hopes of improving energy efficiency and providing more control over tenants' living and working environments. Building automation systems (BAS) are used to control the utilities and appliances based on input provided by different sensors. Constructions that utilise some of these advanced BAS are often referred to as intelligent or smart buildings. One of the key characteristics of the construction industry in the 21st century is the interconnectivity of these systems with the introduction of cloud-enabled real-time monitoring, management and controlling solutions. Remote interfaces are added to different building systems to let vendors perform maintenance remotely and offer a broader view on how the systems are operating. Internet-enabled interfaces also provide tenants more features by allowing them to control their residence preferences such as lighting and temperature to their liking. This kind of combination of information technology (IT) and operational technology (OT) can lead to a complex system architecture and introduce additional cybersecurity challenges.

1.1. Motivation

People spend the majority of their lives indoors. According to The National Human Activity Pattern Survey (NHAPS)[1] in 2001, U.S. citizens spend an average of 87 % of their time inside buildings. A study by Hussein, Paasonen & Kulmala[2] in 2012 shows that these results apply to Finland as well with the indoor time among Helsinki residents varying from 81% to 92%. This drives the construction industry to progress and adopt new technologies in the attempt to create more customisable, comfortable and energy efficient living and working environments.

Since the construction industry is progressing fast and adopting intelligent and networked systems in their structures, buildings are also producing vast amounts of data. Different industry sectors are interested in smart buildings as data collection platforms and building data is considered to be a valuable asset that has yet to be used to its full potential. In 2016, Bilal et al.[3] reviewed the state of big data in the construction industry and recognised its usefulness in several sub-domains within the construction industry, e.g. resource and waste optimisation, personalised services, facility management, building information modeling and energy management and analytics. Despite the fact that building data is considered to be a valuable asset, its utilisation in real world applications is lagging behind and authors have recognised several issues that should first be resolved. These topics include security, privacy and protection of the building data.

Because of the convergence of IT and OT devices and networks, the system environment of a smart building can be highly complex. There are no complete solutions for protecting and assessing the security of smart building ecosystems, so applicability of methods and guidelines used with environments with shared traits should be investigated. The cyber-physical nature of these environments suggests that for example industrial control systems (ICS) and smart grids share some similarities with smart building environments when it comes to use of automation systems and communication protocols, but the level of cybersecurity maturity is what really sets them apart. This is mostly because the systems are meant to be used in different types of network environments: ICS operate typically in separate protected networks that offer no connection to other networks or external systems, while one of the key features for advanced building automation systems is the ability to offer remotely accessible features. This can lead to increase in e.g. configuration-related issues that can cause unwanted exposure of system interfaces, which can also be seen in the report by National Cyber Security Centre of Finland (NCSC-FI)[4]. In 2019, NCSC-FI inspected 14 million IP addresses in 1500 networks to discover unprotected automation systems. These systems were divided into three categories: building automation, industrial automation and ICS. The results of this show that 968 (88%) of the found 1,102 unprotected exposed systems were related to building automation while 86 (8%) were part of industrial automation and only 48 (4%) were industrial control systems. The high portion of building automation systems in these statistics shows that there is demand for improving cybersecurity in building automation, and the challenge should be acknowledged during the industry transformation towards smart building ecosystems.

1.2. Study Design

To understand the cyber security requirements of a modern cloud-enabled smart building ecosystem, this work describes the system architecture of a typical building automation environment, presents how different types of systems operate in it and examines the objectives of construction industry's move towards data-driven building management with the help of a recent microservice-based building data platform, KEKO[5]. KEKO ecosystem is an experimental building data platform developed by a consortium consisting of several major companies and organisations in Finland operating on different industry sectors. A widely recognised cyber threat intelligence framework, MITRE ATT&CK[6], is used to perform threat modeling on the ecosystem to identify the most relevant cyber security challenges and the building data management process is evaluated based on General Data Protection Regulation (GDPR)[7]. In the attempt to improve the overall security awareness of the smart building environment, a Security Information and Event Management (SIEM) system is applied to monitor the microservice architecture of the platform. Results from threat modeling process are applied to define required data sources and create a set of rules that allow detecting possible system misuse incidents. The SIEM system applicability with this kind of environment is tested by running a fabricated service which emulates adversarial actions targeted towards a building automation system by producing falsified measurements that attempt to alter how the system operates. These results are also used to define recommendations for improving the security preparedness of future data-centric buildings.

2. THREAT MODELING

When developing any software, system or network, it is important to acknowledge all the different ways malicious actors could attempt to misuse it. Threat modeling is a process used to gain understanding on how the system could become compromised. In 2014, Shostack[8] published a book in which he provides motivation for using threat modeling for assessing security of a system. According to the author, the process can be used to discover vulnerabilities in software even before development, understand the security requirements of a system, improve the quality and security of developed products and address issues that would otherwise remain unacknowledged. There are several strategies for threat modeling that offer various approaches for identifying and addressing threats on different levels of operation, but generally the threat modeling process follows steps described by Myagmar, Lee & Yurcik[9]:

1. Characterizing the system: gaining understanding of the system architecture.
2. Identifying assets and access points: recognise valuable assets, motivation of adversaries and possible paths they might take.
3. Identifying threats: use gathered information to enumerate adversaries' potential goals and investigate ways to mitigate them.

In addition to these steps, OWASP[10] recognises the importance of the retrospective evaluation to detect possible oversights in the created threat model.

This chapter introduces the methodology of two threat modeling frameworks: ATT&CK by MITRE[6] and STRIDE by Microsoft[11]. Although the threat modeling process itself with both of these follows the steps mentioned above, the strategies have different focuses and they are often applied for different purposes.

2.1. MITRE ATT&CK

In 2013, MITRE[12] began to categorise known adversary behaviour to create a knowledge base that could be used to provide behavioural insight to adversary emulation. They published their first Adversarial Tactics, Techniques and Common Knowledge (ATT&CK) model, which focused on Windows enterprise environments. Since then, ATT&CK has been expanded to include different types of environments and the current version, ATT&CK v11, supports systems such as macOS and Linux enterprise networks, mobile platforms, cloud services and ICS. The framework can be used to assess the effectiveness of security measures used in a system by emulating adversary, offering additional information to cyber threat intelligence (CTI), recognising defensive gaps and implementing behavioural analytics to monitoring systems. The effectiveness of the framework is based on five principles as described in MITRE's white paper[13] from 2017:

1. Include post-compromise detection: if adversary bypasses defensive mechanisms of their target network environment, knowledge of their actions can help to minimise the damage caused by the incident.

2. Focus on behaviour: although indicators of compromise (IOC) like file hashes, signatures and IP addresses are used extensively by network defense mechanisms to detect malicious activity, it is easier for adversaries to alter them rather than changing their behaviour patterns to avoid detection.

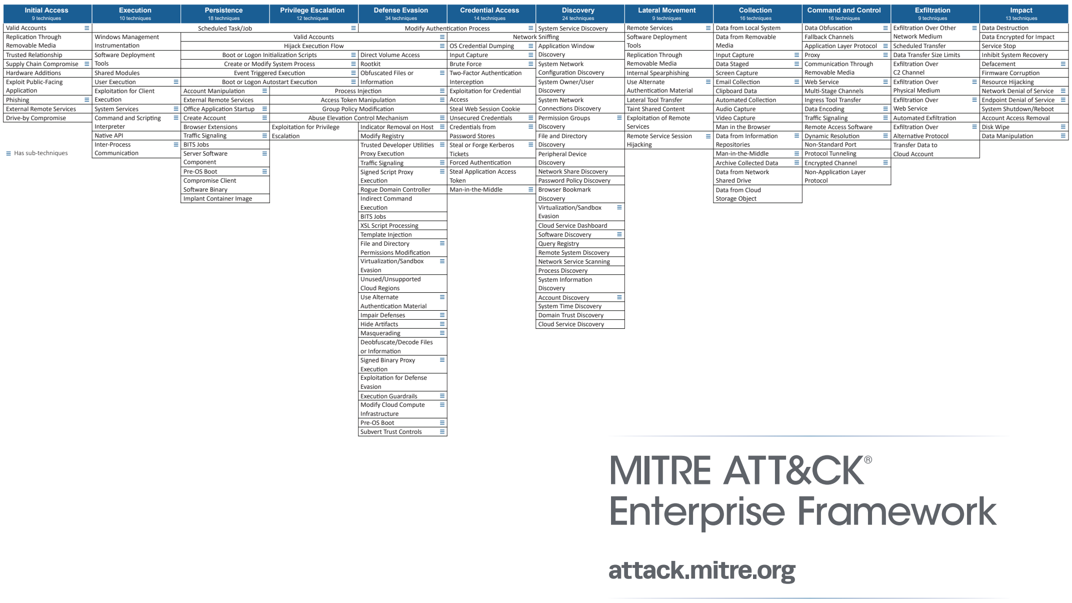
3. Use a threat-based model: knowledge of threat actor behaviour helps to specify required defense strategies.

4. Iterate by design: adapt methodology based on recent discoveries on adversary behaviour and actions.

5. Develop and test in a realistic environment: emulating threat in a realistic environment allows to detect alterations of techniques specific to that target environment.

2.1.1. ATT&CK matrix

The known tactics, techniques and procedures (TTP) in the ATT&CK framework and the relationship between them are visualised with a matrix. An overview of the ATT&CK v8 for Enterprise matrix is presented in Figure 1[14]. The ATT&CK for Enterprise matrix contains the information for different types of enterprise network environments. These include e.g. Windows, macOS and Linux based network environments, different cloud service platforms and an expansion for preparatory techniques adversaries use to gain knowledge of their target before the initial attack. In addition to ATT&CK for Enterprise, MITRE has published matrices for mobile platform and ICS. The knowledge base is updated frequently and the most recent information and matrices are available in the MITRE ATT&CK website[6].



Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 34 techniques	Credential Access 14 techniques	Discovery 24 techniques	Lateral Movement 9 techniques	Collection 16 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Valid Accounts ≡	Scheduled Task/Job ≡	Scheduled Task/Job ≡	Scheduled Task/Job ≡	Modify Authentication Process ≡	Modify Authentication Process ≡	System Service Discovery	Remote Services ≡	Data from Local System	Data Obfuscation ≡	Exfiltration Over Other Network Medium ≡	Data Destruction
Replication Through Removable Media	Windows Management Instrumentation	Valid Accounts ≡	Valid Accounts ≡	Valid Accounts ≡	Network Sniffing	Network Sniffing	Software Deployment Tools	Data from Removable Media	Fallback Channels	Scheduled Transfer	Data Encrypted for Impact
Trusted Relationship	Software Deployment Tools	Hijack Execution Flow ≡	Hijack Execution Flow ≡	Hijack Execution Flow ≡	OS Credential Dumping ≡	Application Window Discovery	Replication Through Removable Media	Input Capture ≡	Application Layer Protocol ≡	Data Transfer Size Limits	Service Stop
Supply Chain Compromise ≡	Shared Modules	Boot or Logon Initialization Scripts ≡	Boot or Logon Initialization Scripts ≡	Direct Volume Access	Input Capture ≡	System Network Configuration Discovery	Internal Spearphishing	Data Staged ≡	Proxy ≡	Exfiltration Over C2 Channel	Inhibit System Recovery
Hardware Additions	User Execution ≡	Create or Modify System Process ≡	Create or Modify System Process ≡	Rootkit	Brute Force ≡	System Owner/User Discovery	Use Alternate Authentication Material ≡	Screen Capture	Communication Through Removable Media	Exfiltration Over Physical Medium ≡	Defacement ≡
Exploit Public-Facing Application	Exploitation for Client Execution	Event Triggered Execution ≡	Event Triggered Execution ≡	Obfuscated Files or Information ≡	Two-Factor Authentication Interception	System Network Connections Discovery	Lateral Tool Transfer	Email Collection ≡	Web Service ≡	Exfiltration Over Web Service ≡	Firmware Corruption
Phishing ≡	System Services ≡	Boot or Logon Autostart Execution ≡	Boot or Logon Autostart Execution ≡	Process Injection ≡	Exploitation for Credential Access	Permission Groups Discovery ≡	Taint Shared Content	Clipboard Data	Multi-Stage Channels	Automated Exfiltration	Resource Hijacking
External Remote Services	Command and Scripting Interpreter ≡	Account Manipulation ≡	Process Injection ≡	Access Token Manipulation ≡	Steal Web Session Cookie	File and Directory Discovery	Exploitation of Remote Services	Automated Collection	Ingress Tool Transfer	Exfiltration Over Alternative Protocol ≡	Network Denial of Service ≡
Drive-by Compromise	Native API	External Remote Services	Access Token Manipulation ≡	Group Policy Modification	Unsecured Credentials ≡	Peripheral Device Discovery	Remote Service Session Hijacking ≡	Audio Capture	Data Encoding ≡	Transfer Data to Cloud Account	Endpoint Denial of Service ≡
	Inter-Process Communication ≡	Office Application Startup ≡	Group Policy Modification	Abuse Elevation Control Mechanism ≡	Credentials from Password Stores ≡	Network Share Discovery		Video Capture	Traffic Signaling ≡		System Shutdown/Reboot
		Create Account ≡	Abuse Elevation Control Mechanism ≡	Indicator Removal on Host ≡	Steal or Forge Kerberos Tickets ≡	Password Policy Discovery		Man in the Browser	Remote Access Software		Account Access Removal
		Browser Extensions	Exploitation for Privilege Escalation	Modify Registry	Forced Authentication	Browser Bookmark Discovery		Data from Information Repositories ≡	Dynamic Resolution ≡		Disk Wipe ≡
		Traffic Signaling ≡		Trusted Developer Utilities Proxy Execution ≡	Steal Application Access Token	Virtualization/Sandbox Evasion ≡		Man-in-the-Middle ≡	Non-Standard Port		Data Manipulation ≡
		BITS Jobs		Traffic Signaling ≡	Man-in-the-Middle ≡	Cloud Service Dashboard		Archive Collected Data ≡	Protocol Tunneling		
		Server Software Component ≡		Signed Script Proxy Execution ≡		Software Discovery ≡		Data from Network Shared Drive	Encrypted Channel ≡		
		Pre-OS Boot ≡		Rogue Domain Controller		Query Registry		Data from Cloud Storage Object	Non-Application Layer Protocol		
		Compromise Client Software Binary		Indirect Command Execution		Remote System Discovery					
		Implant Container Image		BITS Jobs		Network Service Scanning					
				XSL Script Processing		Process Discovery					
				Template Injection		System Information Discovery					
				File and Directory Permissions Modification ≡		Account Discovery ≡					
				Virtualization/Sandbox Evasion ≡		System Time Discovery					
				Unused/Unsupported Cloud Regions		Domain Trust Discovery					
				Use Alternate Authentication Material ≡		Cloud Service Discovery					
				Impair Defenses ≡							
				Hide Artifacts ≡							
				Masquerading ≡							
				Deobfuscate/Decode Files or Information							
				Signed Binary Proxy Execution ≡							
				Exploitation for Defense Evasion							
				Execution Guardrails ≡							
				Modify Cloud Compute Infrastructure ≡							
				Pre-OS Boot ≡							
				Subvert Trust Controls ≡							

≡ Has sub-techniques

Figure 1. MITRE ATT&CK v8 for Enterprise matrix.

In the ATT&CK matrix, the title for each column represents the adversary's tactic or goal. They portray the reason for adversary's actions, and each contains several techniques and sub-techniques for achieving the goal. ATT&CK for Enterprise categorises the tactics in the following way:

- Initial access: includes techniques for adversaries to gain a foothold within their target system.
- Execution: methods for executing adversary-controlled code on target systems.
- Persistence: techniques that help adversary gain persistent access to a system.
- Privilege escalation: obtaining higher level permissions in the target network.
- Defense evasion: ways for adversaries to stay undetected.
- Credential access: accessing legitimate system credentials.
- Discovery: techniques for adversaries to collect knowledge of their target system environment.
- Lateral movement: methods for accessing other systems within the same target network.
- Collection: enumerate information and files to be exfiltrated from the system.
- Command and control: techniques for controlling the target system after gaining persistent access.
- Exfiltration: ways to add, modify, remove or transfer files in and out of the target system.
- Impact: techniques for altering operational processes of the target.

Different objects in each column describe the techniques to achieve the tactical objective. For instance, if the adversary attempts to gain initial access to the environment, ATT&CK for Enterprise recognises nine main techniques with some additional sub-techniques that they might use which are represented in Figure 2[6]. All of these techniques do not apply for all types of systems or threat groups, so it depends on the type of the target environment which techniques are taken into consideration. The difference between ATT&CK matrices for different domains is in the use of different tactics and techniques.

Techniques and sub-techniques contain the information of different procedures, the concrete actions, that adversaries use as part of each technique. They often include using legitimate system functions for malicious purposes, so detecting them based on IOC alone may prove to be challenging. Monitoring the event sequences for behaviour-based analysis can provide additional information and allow the detection of that type of malicious activity.

To make the behaviour-based analysis possible, ATT&CK framework contains information about known adversaries and threat groups. This information is collected from threat intelligence reports disclosed by public and private organisations, and combined to create known behaviour patterns and attack sequences. The groups include intrusion sets, threat actors or campaigns with specific targets, but the main

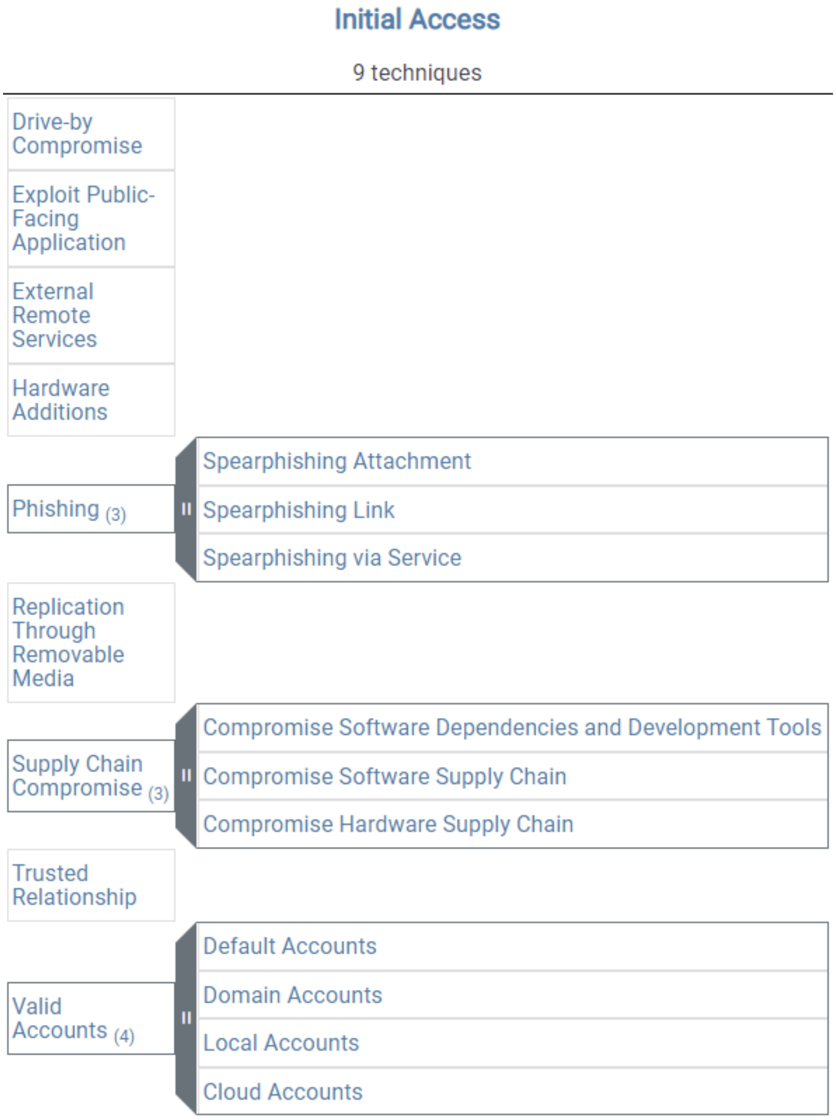
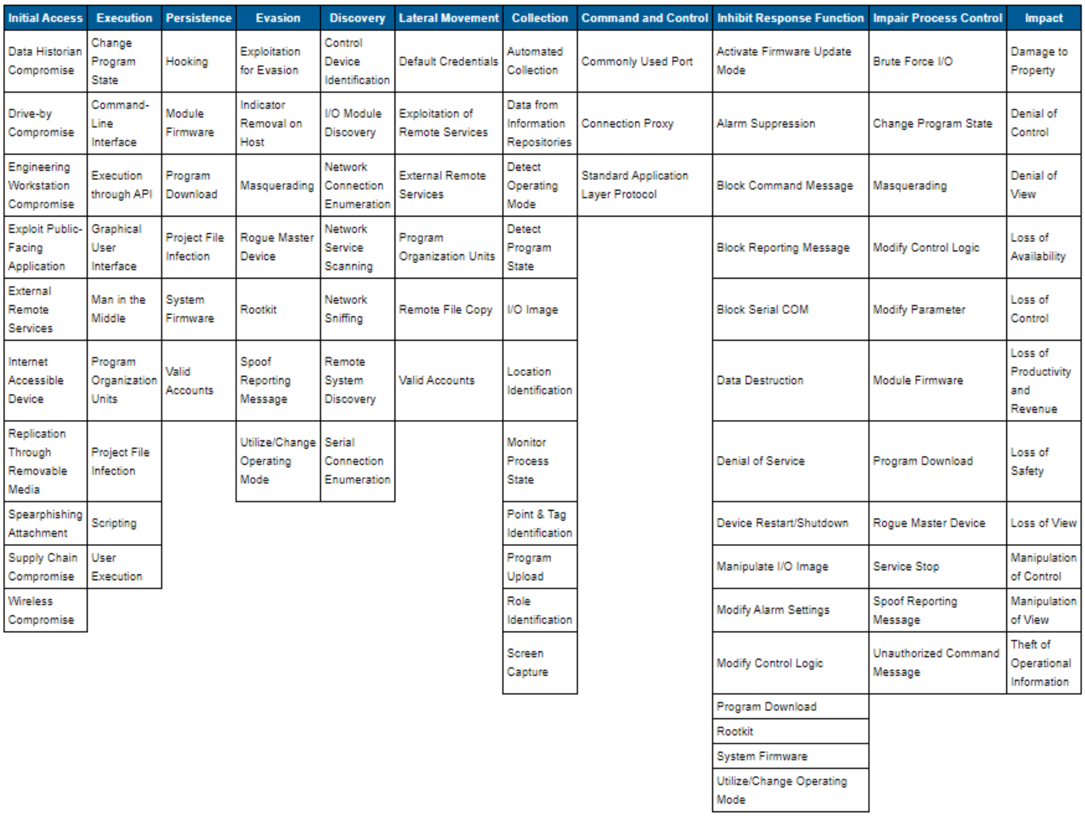

Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3)
Spearphishing Attachment
Spearphishing Link
Spearphishing via Service
Replication Through Removable Media
Supply Chain Compromise (3)
Compromise Software Dependencies and Development Tools
Compromise Software Supply Chain
Compromise Hardware Supply Chain
Trusted Relationship
Valid Accounts (4)
Default Accounts
Domain Accounts
Local Accounts
Cloud Accounts

Figure 2. MITRE ATT&CK for Enterprise - Initial Access with sub-techniques.

focus of ATT&CK is on Advanced Persistent Threats (APT). They are well-resourced adversary actors that have the ability to use sophisticated methods to compromise target environments. According to Tankard[15], the terms used to describe these threats refer to the use of advanced exploits and their goal of maintaining long-term presence on the target system. MITRE ATT&CK v8 recognises 109 different APT groups with some of them containing additional associated actors. Some examples of these groups are APT28, a group that has been active since 2004 and is associated with Russia's Main Intelligence Directorate's (GRU) Unit 26165, and the Lazarus group which is associated with the North Korean government[6].

2.1.2. ATT&CK for ICS

In June 2020, MITRE expanded their ATT&CK framework by publishing the matrix for ICS environments[16]. It focuses on threats targeting physical processes and operational environments which are controlled and managed with network-connected human-machine interfaces (HMI). The main idea of the ATT&CK for ICS is the same as with ATT&CK for Enterprise: provide framework and knowledge base for existing threat group behaviour for the platform in question. The ATT&CK matrix for ICS is depicted in Figure 3[17].



Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

Figure 3. MITRE ATT&CK for ICS matrix.

When comparing ATT&CK for ICS to its enterprise counterpart, there are some differences with both the tactics and techniques recognised. ATT&CK for ICS does not consider privilege escalation, credential access and exfiltration to be commonly used tactics with incidents concerning industrial environments. Instead, the matrix presents two new tactical goals for adversaries: inhibiting response function and impairing process control. The first one includes techniques used to prevent automatic safeguards such as safety or quality assurance functions from operating. The second one consists of techniques used to disturb and manipulate how controllers operate leading to effects in the physical environment. The cyber-physical traits shared by both the building automation systems and ICS environments indicates that some threats may also exist on both platforms, so TTPs included in the ATT&CK for ICS matrix can be relevant for smart building environments as well.

2.2. Microsoft STRIDE

The second approach to threat modeling introduced here is called STRIDE. It was created by then Microsoft employees Kohnfelder and Garg[18] in 1999 to help their colleagues to "identify potential vulnerabilities in your product during a security analysis". Unlike MITRE ATT&CK, STRIDE is not a knowledge base of threats per se, but more of a developer focused mnemonic for all of the things that could go wrong in security. The name STRIDE is an acronym of six major threat categories: spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. Each of these threats target some system property that can be described using information security principles CIA (confidentiality, integrity, availability) triad or AAA (authentication, authorization, accounting) model. Nweke summarises these two principles well in his article[19] stating that the CIA triad describes the goals for cyber security and AAA actualises these objectives.

- Spoofing can be described as impersonating someone or something else to perform actions that might otherwise be unavailable. Adversaries can spoof e.g. processes or files to make a system run malicious code, machines by modifying data on various levels of network stack to appear legitimate system on that network or people in the form of phishing attacks. Spoofing violates the authentication property, because the techniques used in these types of attacks attempt to circumvent the mediating authority's confirmation of the system's legitimacy.

- Tampering threats include the techniques to modify data without permission. For example, adversaries may try to modify files containing operational information to alter system's activities or tamper with the network to read and edit packets transported from a reliable source. This threat violates the integrity of the data.

- Repudiation is the act of either honestly or deceptively claiming that someone else did or is the responsible for some actions. These threats typically appear on the business layer, and the repudiating party is not necessarily an attacker, but rather a person or an employee who could either truthfully or deceivingly e.g. claim to be a victim of a fraud. These threats violate the accountability principle, and they can be mitigated using appropriate logging and authentication mechanisms.

- Information disclosure threats cause unwanted exposure of data to parties that are not authorised to see the information leading to violation of the confidentiality of the system. Adversaries can try to leak information from multiple sources, such as running processes, data stores and communication between network or system components.

- Denial of service threats aim to prevent valid users from using or accessing resources and services breaking the availability of the system. Adversaries may either actively attempt to block the use of resources by e.g. creating and sending enough requests to flood the network or create more persistent attacks by e.g. encrypting the data or absorbing all available memory so other processes can not use it.

- Elevation of privilege infringes the authorization property of the system by allowing unauthorized users to obtain and operate on higher level of permissions. Shostack[8] recognises two main methods for elevating privileges: corrupting a process and taking advantage of authorization failures. Adversaries can use these techniques by e.g. exploiting vulnerabilities in software components.

Table 1 is derived from the STRIDE chart created by Shostack[8] and it summarises the STRIDE threat categories, properties they violate and some examples of typical incidents.

Table 1. The structure of STRIDE.

Threat Category	**Violated Property**	**Definition**	**Examples**
Spoofing	Authentication	Impersonating a legitimate user or resource to access a system	IP, ARP & DNS spoofing, forging email sender address, renaming files and processes with names of existing ones
Tampering	Integrity	Modifying data without permission	Modifying code running with the same level of privilege, editing local or remote files to run malicious code
Repudiation	Accountability (Non-repudiation)	Claiming that some other party is responsible for actions	Saying that someone else modified a file, claiming not clicking on a phishing email link
Information disclosure	Confidentiality	Exposing resources to someone who does not have the permission to view it	Extracting operational information from error messages or logs, reading contents of exposed data stores, reading data from network traffic
Denial of service	Availability	Denying the use of a service or a resource	Absorbing all available memory so other processes can not use it, flooding network with malformed packets
Elevation of privilege	Authorization	Allowing users to perform actions they are not authorised to do	Corrupting process with input that is not handled properly, exploiting vulnerable authorization implementation

3. SMART BUILDINGS

Modern smart building ecosystems are complex ensembles of physical and logical elements attempting to operate in harmony to improve efficiency and increase the comfort of building users. To be able to detect the security requirements of these environments, one must first understand how they operate. This chapter summarises briefly the architecture of a generic modern smart building ecosystem from the system perspective and presents an overview of common elements in existing system architectures.

3.1. Definition

On the concept level, intelligent and smart buildings have been around for decades. The definitions used to describe them vary from the constructions containing some automated controls for utilities to self-aware entities in science fiction literature. In 2014, Buckman, Mayfield and Beck[20] reviewed some often used definitions for intelligent and smart buildings in academic, industrial and popular literature, and concluded that smart buildings “integrate and account for intelligence, enterprise, control, and materials and construction as an entire building system, with adaptability, not reactivity, at the core, in order to meet the drivers for building progression”. The authors identify energy, efficiency, longevity, comfort and satisfaction to be the progression drivers. Although this may be true, the industry may also be driven forward by the desire to create new business opportunities with service-based building operation model. By using features that require additional maintenance such as supplementary equipment or constant software updates, vendors are able to keep building owners as customers even long after the building project is completed.

Terms smart building and intelligent building have sometimes been used interchangeably, but Buckman et al. recognise an important difference between them: adaptability instead of reactivity. Smart buildings are capable to gather more information from a wider range of sources than intelligent buildings and use this data to provide more adaptable environment for tenants. This adaptability could mean e.g. predicting occupancy rates for different spaces during different times of year and adjusting the utilities to accommodate it.

3.2. System Environment

The ecosystem of a smart building consists of physical structures, OT systems, IT systems, various networks, sensors and actuators. The basis of the system design described in this section is derived from a book by Sinopoli[21] published in 2009. The description is expanded by including later advancements in building automation system development[22], and by dividing the on-premise devices and networks into three domains: field domain, automation domain and management domain. In addition to these, there is a fourth one, the external domain, which consists of remote systems that typically communicate with management systems. An overview of how these domains set in a typical smart building ecosystem is presented in the Figure 4.

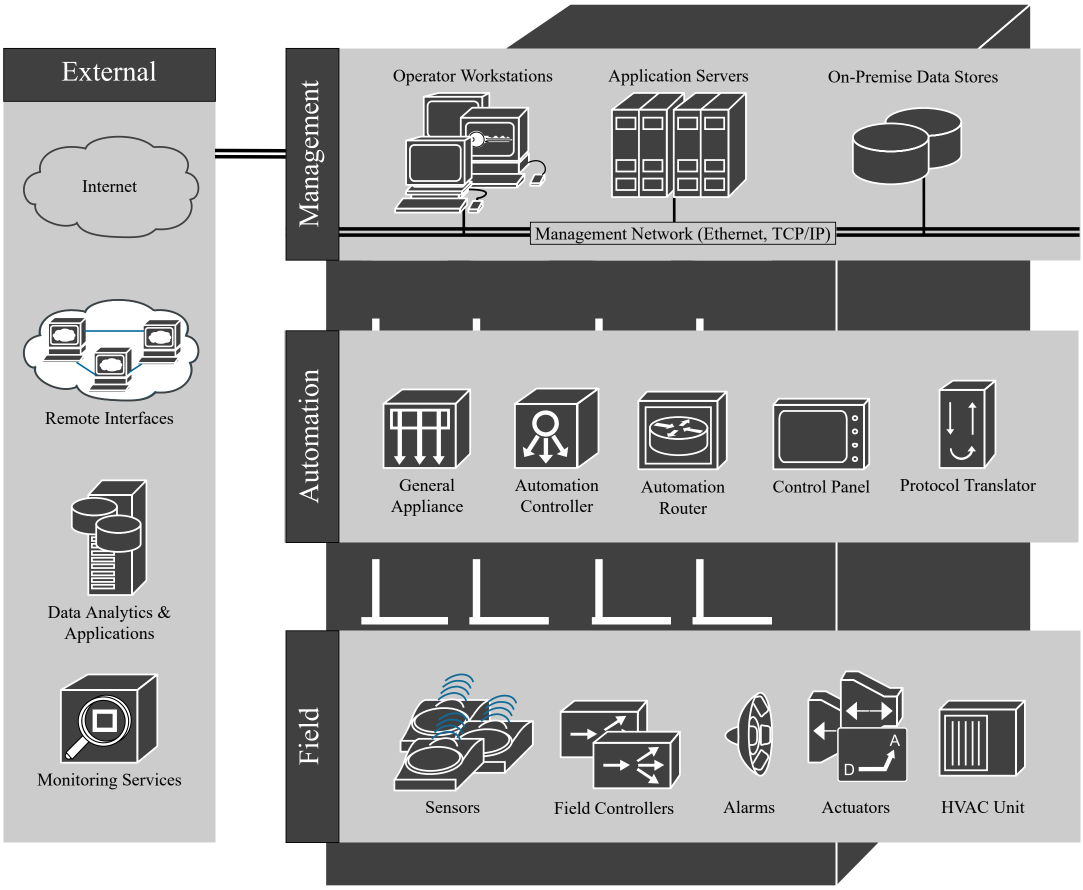


Figure 4. Overview of smart building system domains.

Systems within the smart building ecosystem can operate on multiple domains. The field domain includes the equipment that either measures or controls the physical properties in the building, e.g. temperature, humidity and occupancy. They send sensor data and receive instructions from devices in the automation domain, in which the data is processed, stored and forwarded to the upper level systems operating in the management domain. Management systems then create overviews of this information for building operators or send the data for remote services for data analysis or monitoring purposes.

3.2.1. Network Infrastructure

Traditionally, building system vendors have networked their entire systems and units separately. This means that each building system has utilised their own separate proprietary networks. These networks can be divided to operational technology (OT) and information technology (IT) networks based on the devices connected to them and what communication protocols they use. Management domain devices typically communicate via the core IT network, the backbone. This is a standard Ethernet-based wired TCP/IP network that provides interconnectivity between building systems from different vendors. Some workstations and management devices in this network can also offer connectivity to the Internet. Automation domain systems can also use this network to communicate with management systems, or alternatively they can be connected to the management servers directly. Direct connections typically follow the structured cabling concept familiar from the IT network, but there may be some variance in the protocols used. Communication between the field and automation domain devices typically happens in separate OT networks that utilise operational communication protocols such as BACnet, LonTalk, KNX or some vendor-specific proprietary protocol. A simplified overview of this type of traditional building automation network architecture is presented in Figure 5.

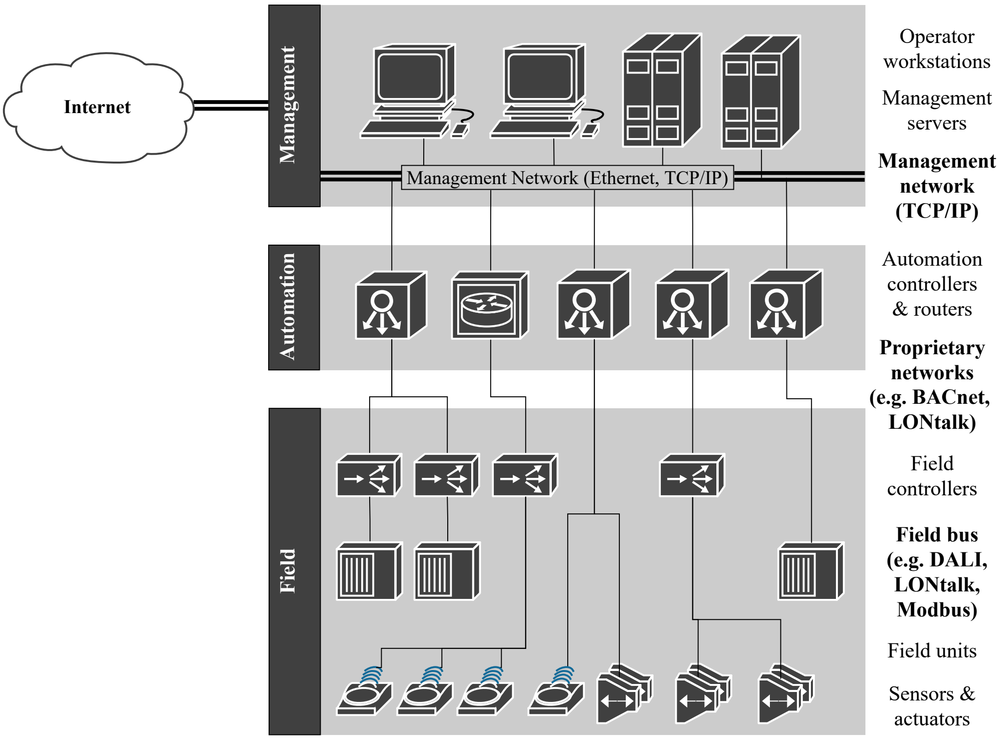

Internet
Management
Automation
Field
Management Network (Ethernet, TCP/IP)
Operator workstations
Management servers
Management network (TCP/IP)
Automation controllers & routers
Proprietary networks (e.g. BACnet, LONtalk)
Field controllers
Field bus (e.g. DALI, LONtalk, Modbus)
Field units
Sensors & actuators

Figure 5. Building automation network hierarchy.

Recently the network infrastructure of intelligent buildings has become more complex due to the increased amount of wireless sensor networks[23] (WSN). The hierarchy of building system networks still follows the same pattern as the one presented in Figure 5, but the amount of local wireless networks has increased. These networks consist of nodes which contain various sensors, have limited processing capability and and a wireless network interface for communicating with other nodes and a sink node that collects the data and is connected to a backbone network[24]. Nodes can also be connected directly to a gateway which may improve the interconnectivity with other networks. WSN utilises wireless communication protocols familiar from Internet of Things (IoT) devices and systems, e.g. LoRa, ZigBee, Wi-Fi and Bluetooth[23].

3.2.2. *Management Systems*

Facility management systems (FMS) are used to create a high level overview of the operation of the building. FMS operates on the management level of the ecosystem and it typically consists of a server and operator workstations running the facility management software. On-premise devices belonging to the FMS communicate using standard wired IP network, and the system is often connected to the public Internet allowing remote use. Some FMS can even be designed to be run on a remote server with only integration components running in the local network. FMS is used to assist with the business processes of the facility, and it offers features such as asset management and work order tracking.

Building management systems (BMS) are used to integrate different housing systems into jointly controlled ensembles. In other words, BMS creates an interface for monitoring, supervising and operating multiple systems with a single system. BMS is capable to collect, analyse and present data real-time from multiple systems and create historical summaries. It also offers interfaces between individual smart building systems and is capable of controlling different devices based on events and data received from other vendors' systems. While FMS focuses on the business processes, BMS is used on the operational side. From the infrastructural standpoint, BMS resembles FMS as the communication between the application server, workstations and automation domain devices takes place in the wired IP network.

Energy management system (EMS) offers insight on how the building is using energy. They can be either a standalone system or a software component in a BMS or FMS, and they are used to create interface for monitoring the energy usage of housing systems that use the most electricity. Visualisation of how these systems are located in the management hierarchy of a building automation environment is presented in Figure 6.

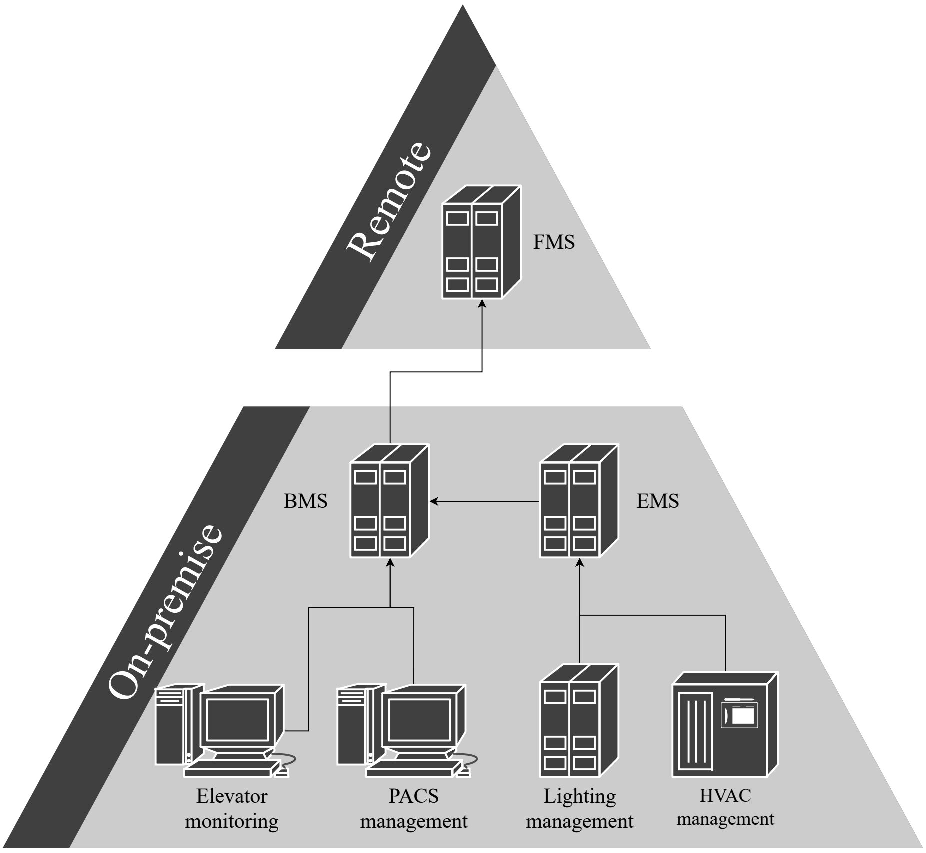


Figure 6. Management system hierarchy.

3.2.3. Heating, Ventilation and Air Conditioning (HVAC) Systems

Heating, ventilation and air conditioning (HVAC) systems are used to control the temperature, humidity and ventilation of the building. A typical HVAC system includes boilers or other heating units, air conditioning units (A/C), air-handling units (AHU), air terminal units (ATU) and variable air volume equipment (VAV). HVAC operates in three domains: management, automation and field. The management level of a HVAC system consists of a server and operator workstations that are connected to a wired IP network. It provides functions such as control interface for the whole HVAC system, historical data storage and creation of alarms and reports from lower level devices.

The management server is connected to the automation level system controllers via the same IP network. These controllers have multiple communication interfaces, and they operate the field level devices either directly or via field controllers. System controllers are able to store and process data provided by lower level devices, and they can operate independently if they are unable to communicate with the management server. They can also communicate with other system controllers in a peer-to-peer manner. The communication between system controllers and field controllers takes place in an Ethernet-based OT network using industrial communications protocols such as BACnet or LonTalk[25] and it allows field controllers to communicate with each other as well.

In addition to field controllers, the field domain contains sensors and actuators including thermostats, pressure sensors and humidity sensors. Field controllers operate the actuators based on the commands received from the system controllers. On this level, the devices communicate with analog or digital signals.

3.2.4. Lighting Systems

Lighting is one of the most costly building system when it comes to energy consumption. This motivates constructors and building owners to integrate intelligent lighting control systems in new buildings. A typical system consists of a server running the control software and hosting a website or a web API to access it, operator workstation, system controllers, relay panels, occupancy sensors and dimmer switches. The server, workstation and system controllers are connected via wired IP network. Relay panels operate the specific lights based on predefined rules or input from dimmer switches, occupancy sensor or system controller, and they communicate with system controllers using some industrial communications protocols (e.g. BACnet, LonTalk, ModBus, DALI). Lighting systems often interface with other building systems to enable features such as emergency lighting during fire alarm.

3.2.5. Physical Access Control Systems (PACS)

Security system is a term used to describe a set of multiple different systems that take care of the physical security in a building. Physical access control systems (PACS) are a type of security systems that are used to control the building users' access to building and its resources and to prevent and detect physical breaches. Their use is vital for the

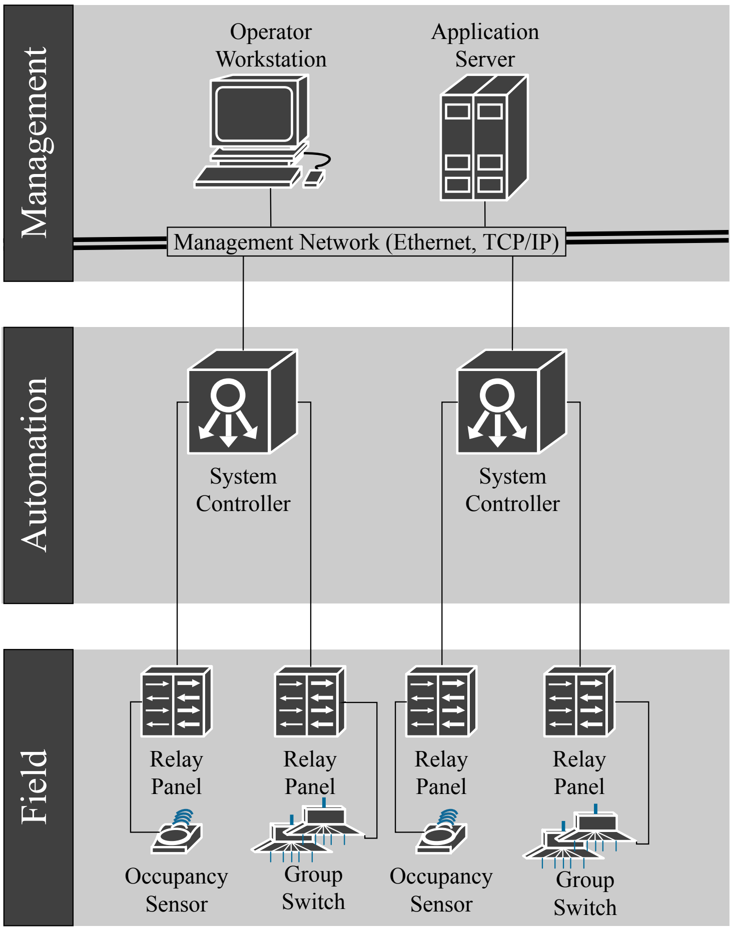


Figure 7. Typical lighting control system.

physical security of a building. A typical PACS consists of a server or a host containing the system's database of valid users and logs, operator workstation for monitoring the system, system controllers that are connected to the server and peripheral devices such as card readers, door position switches and alarms. The authentication of a building user typically happens with a physical key device, such as an RFID tag or a key card. When the key is presented to a reader, the information is sent to the server which verifies the request based on stored information and then instructs the system controller to either allow or deny access. These events are logged on the server and they can be used to provide information for other building systems about tenants' location within the building. Communication between the server, operator workstation and system controllers or control panels can take place in an Ethernet network that offers the interconnectivity with other building systems, e.g. elevator control system. A typical process of authenticating a building user is presented in Figure 8[26].

3.2.6. Video Surveillance Systems

Surveillance systems are a type of a security system that provides video surveillance service of the activities happening on-premise for local and remote use. They have the following functions: capturing video feed of an important location, transmitting

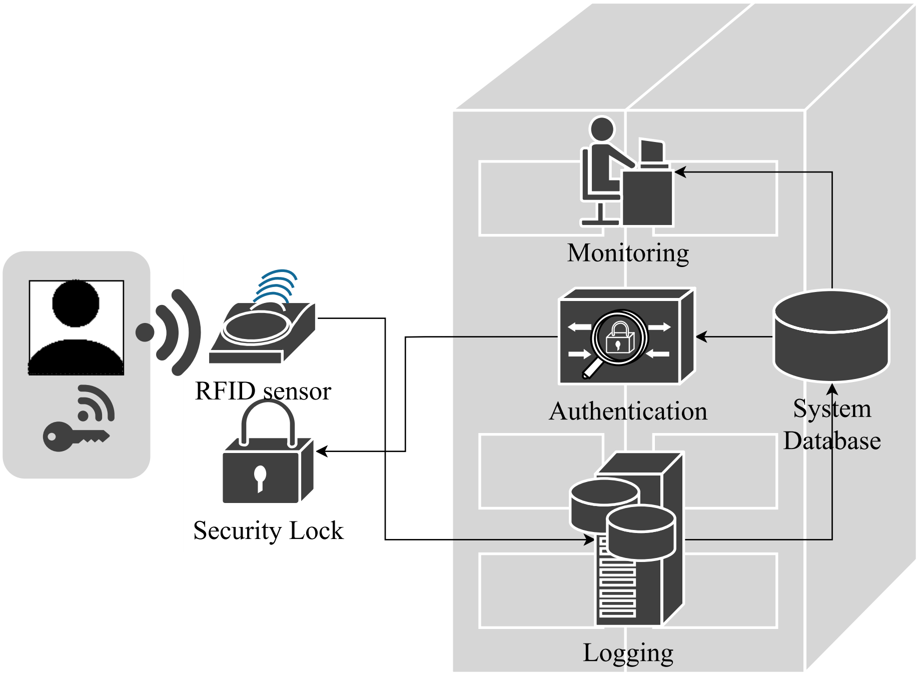


Figure 8. Typical physical access control system.

the feed to security control center, processing the video, recording it and presenting it to security personnel on a monitor. Modern video surveillance systems consist of multiple cameras capable of communicating via IP network, a video server that receives, processes and stores the feed and an operator workstation that is used to view the feed with the client software or via web browser. The communication between the cameras and video server can either take place in a wireless LAN or it can leverage the existing wired IP network. IP-based communication also allows operator to control the cameras directly on their workstation.

3.2.7. Elevator Control Systems

Passenger elevators are electrically powered structures used to move building users vertically between floors. They consist of five main components: the hoisting machinery, the elevator car or cabin, the elevator control system, the control interface and elevator monitoring system. The hoisting machinery is placed either in a separate machine room located above or below the elevator shaft or inside the hoistway, and depending on the type of the elevator, it uses either hydraulic cylinder or electric motor with counterweight and steel belt to raise and lower the elevator car based on the commands received from the control system. The control system is responsible for controlling the acceleration and speed of the car between floors, keeping the elevator leveled and dispatching cars based on input from the control interface. The control system consists of elevator drive controlling the hoisting mechanism and a set of programmable logic controllers (PLC) that read inputs from the control interface and create commands for the elevator drive. The interface consists of user accessible control panels inside the elevator cars, call buttons next to the elevator doors and a separate communication interface used with the elevator monitoring system. The communication between most of these systems utilises fieldbus, but the monitoring system can also exchange information with external systems using the IP network. The monitoring system receives operational status information from the control system and observes the performance of the elevator system continuously. It consists of a workstation or a server running the monitoring software and it allows the service personnel to see the elevator status and perform remote diagnostics and maintenance through a web interface. The monitoring system can also be used to integrate the elevator system to other management systems or data collection platforms in the smart building ecosystem. It is also possible to run the monitoring system on a remote server, and many elevator vendors are providing the remote monitoring to their customers using the Software-as-a-Service (SaaS) business model.

3.2.8. Fire Alarm Systems

Regarding the life safety of tenants, fire alarm systems are one of the most important set of systems in a building environment. In commercial buildings, a basic fire alarm system typically consists of smoke and heat detectors, water flow switches, manual alarm switches and notification devices. All these components are connected to a fire alarm control panel with a structured cable connection using some industrial communication protocol. The control panel is also connected to the backbone network to communicate with other building systems in case of a fire to e.g. tell PACS to unlock doors for people to escape or make HVAC decrease or increase the airflow in certain areas to suppress fire or eject smoke. This connection is also important for monitoring the system, remote testing and creating external alerts for local fire and rescue authority via Internet. The connection between fire alarm systems and central station receivers uses a dual-destination IP receiver address for improved redundancy.

3.3. Communication Protocols

Since smart building environments can have systems designed by multiple different vendors, it is important that they are able to operate together and communicate with each other. Traditionally building automation system vendors have implemented and used their own, proprietary communication protocols with dedicated wiring, but to enable the intercommunication between these systems, smart buildings should use open and standardised communication protocols[27][28]. Favouring standardised open protocols can also be seen as one way of future-proofing the ecosystem so the functionalities of the platform do not become dependant on any single company and their proprietary technologies. If some functionality of the smart building depends on a single vendor that for some reason discontinues support to the used systems, the feature could stop working at all. One recent example of this is a smart home appliance company Insteon[29], which in April 2022 had its cloud-connected devices suddenly stop working. Week after the incident, a representative of the company released a statement where they explained that the company was not able to find new funding solutions causing them to shut down their cloud services. This made their cloud-based features like remote control, automatic scheduling, and mobile application non-functional, also breaking the configuration and control for new or reset devices. Although the company utilised proprietary protocols in their local device-to-hub communications, the open source community has managed to use the network connection of the hub to develop integration with Home Assistant[30], an open source smart home platform, enabling basic device management via local TCP/IP network. This of course is not a desired scenario for any stakeholder involved in a smart building platform, so the risks should be acknowledged already in the planning phase of the building project.

4. SMART BUILDINGS AND CYBER SECURITY

Construction industry's rapid shift towards more information-driven intelligent housing systems has caused the threat landscape of smart buildings to expand. Due to their cyber-physical nature, smart buildings can be susceptible to attacks targeting either the automation systems or the physical environment that the systems measure and alter. However, the increased attack surface does not necessarily correlate with increase in the cyber resilience of these systems. Improving the security features is rarely seen to increase the profitable value of a product when compared to adding other functionalities, which can cause the resilience of a system to lag behind other features. This chapter describes the current state of smart building cyber security, presents some of the recent incidents and their impacts, and reviews some related research efforts that aim to improve the security preparedness of future smart building environments.

4.1. Attack Surface

Although the utilisation of networked automation systems may offer a wide variety of benefits for building owners, users, operators and tenants, it also brings some additional cyber security issues to recognise. The convergence of IT and OT environments increases the complexity of these ecosystems and can cause the attack surface to grow when compared to traditional building automation environments, so it is important that the security of these systems is taken into consideration. Khaund[31] recognises the following attack surfaces in smart buildings:

- Building users
- Remote access
- Physical access
- Integration platforms
- Wireless access

All of these are potential targets for malicious parties who aim to abuse the building's systems. Since network-enabled features are incorporated into various building systems, cyber security incidents may occur in any of the four smart building system domains described in 3.2. Traditionally, OT system security has lagged behind due to reasons including legacy issues caused by the long lifespan of OT systems and the use of proprietary technology[32]. Although the use of standardised open protocols and interfaces has recently become more common to achieve the interoperatibility between devices developed by different vendors[27], the way of thinking about the system security has not evolved at the same pace. In order to improve the overall cyber security preparedness of these environments, it is important to acknowledge the gap between IT and OT systems in the context of system security. The gap consists of key differences of system characteristics between these two types of systems as recognised by McBride, Schou & Slay[33] and it includes attributes such as what kind of information is processed in the system, how does that information affect other systems, what effects does system malfunction have and how long is the system lifecycle. From this prespective, smart buildings are similar to industrial computer systems. Smart building ecosystems also share resemblance with internet of things (IoT) environments, since both consist of sensors, actuators and various IT assets. The cyber-physical nature of these systems makes them terrifying targets for cyber attacks, since incidents can easily cause also physical damages.

4.2. Attacker Profiles

There are several types of bad actors with diverse resources committing attacks against information systems. They are fueled by various motives, including thrill seeking, revenge, economical motivation or reasons concerning social, political and ideological issues[34]. Some actors may also be interested in the reputation they could gain from identifying vulnerabilities in publicly used systems and software. Whatever the motivation behind these bad actors is, it is often related to their level of expertise, available resources and targeted organisations.

There are different categorisations available of the different types of actors in the cybercrime space, and Meyers, Powers & Faissol[35] have created a taxonomy summarised in Table 2 consisting of eight types of adversaries that can be categorised based on their skills, motivation and what methods they use to reach their goals.

The least sophisticated adversary group, script kiddies, are novices with limited skills that use existing free tools and exploits to introduce themselves to the hacker subculture. They are typically motivated by revenge or pursue of new sensations to fight boredom. Although the actors in this group have typically been able to perform attacks with only relatively low effects, the increasing capabilities of freely available tools and paid services, such as botnet-for-hire, have increased the novices' ability to perform large-scale attacks against organisations.

Hacktivists are a group of adversaries that perform defacement or denial of service attacks against public or private organisations to promote their own social, political or ideological views. Although their technical skill set can also be considered low, they are typically more capable than novices. Cyber punks have somewhat similar skills to hacktivists, but they are motivated by personal reasons such as gaining reputation, thrill seeking or earning money by placing cryptocurrency miners in vulnerable systems.

Insiders are one of the most formidable groups of adversaries to companies because of their capability to use legitimate privileges to access and abuse systems. This group is typically motivated by monetary reasons or frustration and revenge, and although they can have a varying level of technical skills, they can use their knowledge on the organisation and its systems to steal, modify and destroy crucial information. They can also work together with more technically skilled adversaries to share insider information that could be used to create customised exploit kits and malware to target the organisation. Coders and writers are one adversary group that would benefit from this information. They create tools used by other groups, like novices, and they are motivated by revenge, economical reasons or gaining reputation in hacking community. Bad actors in this group can also create e.g. video game exploits that are then sold to players.

Table 2. Adversary taxonomy

Category	Skill set	Motivation
novice (script kiddie)	existing tools and exploit kits	revenge, thrill seeking
hacktivists	existing tools and known vulnerabilities in defacement and denial of service attacks	social, political, ideological
cyber punks	existing tools, known vulnerabilities and some custom scripts	economical, thrill seeking, reputation
insiders	abusing insider privileges	revenge, economical
coders, writers	customised scripts, malware and automated tools	revenge, economical, reputation
old guard (white hat, ethical hackers)	varying level of sophistication, automated tools, novel vulnerabilities and custom exploits to identify and report weaknesses	non-malicious, economical, reputation
professionals (black hat)	sophisticated attacks by exploiting novel vulnerabilities	revenge, economical
cyber terrorists	potentially state-backed actors targeting nation states with attacks with varying level of sophistication	ideological, political, economical

Old guard and professionals are groups that both have high technical skills and capability to identify and exploit novel vulnerabilities. The main differences between these two groups are their motivation and how they utilise their findings. Old guard, also referred as ethical hackers or white hat hackers, are individuals who attempt to abuse systems to identify weaknesses and vulnerabilities without performing criminal actions. They then report their findings to the organisations or authorities responsible for the systems to let them fix the issues before they are exploited by malicious parties. They are motivated by self-education, gaining reputation as a security tester or earning money from a possible bug bounty program enforced by the target company. The professionals, often referred as black hat hackers, operate on the other side of the law. They can be motivated by revenge or, similarly to ethical hackers, economical reasons, but instead of reporting their findings to the organisation with the vulnerable systems, black hats can use their findings to perform attacks themselves or sell them to other malicious actors. Depending on the target systems, these zero-day vulnerabilities can reach high prices in the black market.

One of the malicious groups that can buy these zero-day vulnerabilities is cyber terrorists. They are typically well-resourced groups with varying technological skills and they have a large scale of potential targets ranging from influential individuals to nation states. The motive behind their attacks is often based on ideological, political or economical objectives of their sponsor or backer. This backer could for example be a hostile country that wants to destabilise or disrupt operations of enemy government organisations.

Concerning smart buildings, there can be different types of public and private organisations using them. Depending on their role and connections to other organisations and governmental authorities, adversaries of all the previous categories may be interested in targeting them. In order to prevent and mitigate their attacks, the threats introduced by the use of networked building systems need to be recognised.

4.3. Threat Taxonomy

As mentioned in section 4.1, smart buildings share similarities with ICS and IoT environments. The resemblance between these system environments mean that there are also similarities in how they can potentially be abused and compromised, and according to Kaspersky[36], the threats faced by smart buildings include same ones experienced by other computer systems. In 2017, European Union Agency for Network and Information Security (ENISA) published a study[37] which discusses threats faced by internet-connected control and automation systems and IoT environments in the context of critical information infrastructure. The study presents the following threat categories concerning these environments. A summary of this threat taxonomy is presented in Table 3.

The first category, nefarious activity, consists of threats in which attackers abuse the systems to disturb their operations or make them operate in a malicious way. These threats include different types of malware, vulnerability exploitation for gaining access in a system, multi-stage attacks targeting specific targets, distributed denial of service (DDoS), malicious devices, attacks on user privacy and modification of information produced or processed by the systems. These threats can affect a wide range of environment components.

The second threat category consists of threats with the objective to collect sensitive or operational information from the environment. These include man in the middle (MitM) attacks used for eavesdropping and intercepting network communications and information flow, hijacking the communication between two network nodes by acting as a legitimate host, passively obtaining information on connected devices and the network environment, and repeatedly sending valid messages in the network to cause malfunction or crash of a targeted system.

The threats in the third category include different types of outages. They can occur for several reasons, both accidentally or intentionally and they can be caused by malfunction or failure in network systems, infrastructure devices or required support service. Outages can affect the operations of the whole environment.

The effects of the fourth category, damage or loss of IT assets, can lead to losing or leaking sensitive information to unauthorised parties. Theft of these systems could allow malicious actors to reverse engineer proprietary systems and thoroughly examine them to gain additional knowledge on how a target system could be exploited.

The fifth category, failures and malfunctions, focuses on errors in the system environment. It consists of system vulnerabilities and failures caused by third party systems and components related to the environment. Software bugs, security weaknesses and configuration errors can cause risk to the network environment and allow bad actors to perform malicious actions in it.

Because of the cyber-physical qualities and components of these environments, they can be susceptible to disasters in the physical world. The sixth category includes natural and environmental disasters, and they can cause physical damage to these systems.

Physical attacks including vandalism and sabotage can also cause physical damage to these systems. Tampering with the devices can also lead to unauthorised access e.g. via unprotected physical network interfaces.

Table 3. Threat taxonomy of IoT systems in critical infrastructure

Category	Threats
Nefarious activity / Abuse	Malware Exploit kits Targeted attacks DDoS Counterfeit by malicious devices Attacks on privacy Modification of information
Eavesdropping / Intercpetion / Hijacking	Man in the middle IoT communication protocol hijacking Interception of information Netwok reconnaissance Session hijacking Information gathering Replay of messages
Outages	Network outage Failures of devices Failure of system Loss of support services
Damage / Loss (IT assets)	Data / Sensitive information leakage
Failures / Malfunctions	Software vulnerabilities Third party failures
Disaster	Natural disaster Environmental disaster
Physical attacks	Device modification Device destructions (sabotage)

To expand on this, the study also presents some threat scenarios and their importance levels calculated from the negative impact each attack could have in a real-life incident. These scenarios describe what types of actions malicious parties could take in order to compromise the system. A summary of them is presented in Table 4[37].

The first attack scenario focuses on listening the traffic between control device and physical device it is controlling. With these kind of eavesdropping attacks, malicious parties can extract operational information about systems which can later be used in more severe attacks. The main impact of this kind of incident is the leakage of potentially sensitive operational data.

In the second scenario, the attacker modifies the configuration of the sensors to force them to provide unexpected data to other systems. If some systems are purely dependent on these specific sensors, falsified data may cause power surges, malfunction or breakdowns. For example, modifying smoke detector's data could cause building's fire alarm to go off and automated sprinkler system to start. This could cause damaged or even destroyed property.

The third scenario presents an incident, in which attackers manipulate the physical actuators' configuration. Impacts of this depend highly on what kind of actuator they are targeting and could range from system malfunctioning to damaged property.

In the fourth scenario, attackers are trying to gain control over the administrative system which could lead to entire system being compromised. The attack vector could be some remote maintenance interface that uses default or otherwise weak passwords. If attackers were successful, they could have full control over the building's systems.

The fifth scenario describes the situation where attackers manage to exploit protocol vulnerabilities to gain privileged unauthorised access to control systems. This could be a repercussion of the first presented scenario and is usually part of a larger-scale attack against the systems. Successful attackers could create paths to building systems which they are able to use later, and even if they are unable to do so, they might be able to force the system to malfunction or crash.

In the sixth scenario, the attackers have already gained either physical access to the system control unit console or have accessed a control interface via network and they are injecting commands into the system. This could give them ability to move laterally in the network and gain access to other systems as well.

Stepping stone attacks in the scenario seven can be thought as a continuity to scenario six. If attackers have gained unauthorised access to a system, they might be able to initialise the next sets of attacks from it.

The attack scenario eight includes some BAS becoming a part of a larger botnet which is then used to make distributed denial-of-service (DDoS) attack. The DDoS attack itself might targeted against the building's systems, which could cause a flood of malicious network traffic and system crashes.

In the ninth attack scenario, attackers tamper with the building's power sources by either physically modifying the cabling or using malware to alter the received power levels. If BAS thinks there is not enough power to run some systems, they could be turned off as a result.

The tenth scenario, ransomware, has become a very common problem among consumers and businesses. Ransomware is a type of malware which encrypts the victim's data making it and possibly the whole system unusable. After attackers have infected the system with a malware, they try to blackmail the building owner or the system supplier. This type of attack can be used against different types of targets making it relevant in the context of smart building ecosystems as well.

These scenarios are not just theoretical examples, but several cyber attacks have occurred that utilised smart building systems. During the first half of 2019, 37,8%

Table 4. IoT attack scenarios in the context of critical infrastructure

Attack Scenario	Importance level
Against the network link between controller(s) and actuators	High - Crucial
Against sensors, modifying the values read by them or their threshold values and settings	High - Crucial
Against actuators, modifying or sabotaging their normal settings	High - Crucial
Against the administration systems of IoT	High - Crucial
Exploiting protocol vulnerabilities	High
Against devices, injecting commands into the system console	High - Crucial
Stepping stones attacks	Medium - High
DDoS using an IoT botnet	Crucial
Power source manipulation and exploitation of vulnerabilities in data readings	Medium - High
Ransomware	Medium - Crucial

of smart building systems that used Kaspersky's security solutions faced a malware attack[36]. One noticeable successful malware attack happened during 2017 in Austria, where attackers managed to infect Hotel Jaegerwirt's digital key management system with ransomware[38]. Hotel's owner was forced to pay the attackers to restore the access to the system.

Attacks against smart buildings can also lead to massive data breaches, as shown by the 2013 Target data breach incident[37]. The breach happened, because a third party HVAC vendor with remote access to BAS had their credentials stolen. Attackers managed to exploit the BAS and leverage it to gain access to the store network. This incident resulted in theft of 40 million credit and debit card accounts and could have been avoided with correct security implementations.

In 2013, two security researchers were able to gain access to the building management system of Google's Australian office[39]. The BMS had a vendor-created administrative account using default credentials for allowing remote access and maintenance, and although there was an update available which disabled the account, the BMS at Google office was still unpatched. Because the system was no updated, there was also additional vulnerabilities which the researchers could have been exploited to gain access to other systems in the same network.

5. SMART BUILDINGS AND PRIVACY

Modern smart buildings generate, collect and process vast amounts of data to perform their daily operations. This data is seen as a valuable asset for other industry sectors as well, and there is a high demand for smart building data research. Its utilisation is however lagging behind due to issues concerning security, privacy and protection of the building data[3]. This chapter introduces how building data is collected and processed and what issues can be recognised based on the current legislation.

5.1. Data Collection

Since smart buildings operate on more data-driven basis and offer more information sources than traditional buildings equipped with automation systems, several industries see smart buildings as data collection platforms. Building data is considered to be a valuable asset for many business areas but has yet to be used to its full potential. In 2016, Bilal et al.[3] reviewed the state of big data in the construction industry. Authors acknowledge that although building data is useful in several sub-domains within the construction industry such as resource and waste optimisation, personalised services, facility management, building information modeling and energy management and analytics, its utilisation and the adoption of modern big data analysis methods is lagging behind. This indicates that there is a high demand for smart building data research.

Modern buildings rely more and more on intelligent systems to provide customisable and personalised living and working conditions, and in order to develop more intelligent automation systems and platforms that have the ability to take advantage of building data, the information must be collected and processed properly. One of the suggested models for harnessing the capabilities of big data in smart buildings was introduced by Linder et al.[40] in 2017. The authors proposed a scalable system called BBData, which they describe as a Web of Buildings. In the model, instead of collecting and processing data in single building's management systems for use in that same building, data from all connected buildings is collected and processed in cloud platform and then provided for buildings' applications to use. This approach allows to scale the ecosystem from individual buildings to neighbourhoods or entire smart cities, but it also introduces issues that arise from the large amount of collected data, including processing resources, interoperability of appliances from different vendors and privacy.

Building data can be used to develop services for different types of users, including building owners, tenants, building system vendors and maintenance personnel. Building owners may be interested in waste optimisation and improved energy consumption management while tenants could benefit from automated environmental controls based on weather forecasts and their personal preferences. One of the main enablers for this is user-profiling, which is already used by several industry sectors to obtain knowledge of user or customer behaviour and create more personalised experiences. In their article from 2009, Schiaffino and Amandi[41] discuss about a typical user-profiling process. Authors define a user profile as "a description of someone containing the most important or interesting facts about him or her", and more specifically in the context of software application user, profile contains "essential information about an individual user". Authors recognise the following six different application domains for user-profiling:

1. Adaptive systems: user profiles are used to make systems behave differently for different users
2. Intelligent agents: user profiles are used to provide active personalised assistance
3. Intelligent tutoring systems: user profiles are used to guide learning process based on individual's prior knowledge and learning style
4. E-commerce: customer profile is used to create personalised offers and suggestions
5. Knowledge management systems: employee's skills and knowledge is used to assign them to suitable role
6. Recommender systems: user profile contains preferences on what type of subjects the user is interested in to create personalised recommendations

In the context of this work, adaptive systems is the most interesting domain since smart buildings aim to improve living comfort of their users by providing them environment personalisation. User profiles should contain information about individual's behaviour patterns, interaction preferences, contextual information and some individual characteristics. These can be obtained either explicitly from the users themselves or implicitly by observing users' actions.

Smart building user-profiling typically focuses on improving buildings' energy-efficiency by creating profiles for specific spaces or rooms instead of building users. Room profiles are used to predict the occupation and utilisation rate of a room or an area and use that data to control building's systems. One of these studies was performed by Barbato et al.[42] in 2009. Researchers used wireless sensor network consisting of temperature sensors, lighting sensors and passive infrared (PIR) sensor modules to detect movement in five rooms and create occupation patterns and daily profiles. These profiles were used to predict when the rooms are going to be in use. Their prediction algorithm managed to achieve a very low false prediction rate of only 2.55% or less per room for a simulated period of 300 days. Agarwal et al.[43] continued the work in 2010 by creating a more affordable system for occupation detection. Researchers used door sensors and passive infrared (PIR) sensor modules to detect when office rooms were in use. Researchers managed to achieve similar detection rates with less sensors. The same approach has been developed further to create a Fuzzy Inference System (FIS), which can be used to recognise activities in an office room environment[44]. This makes it possible to extract single user activity logs from relatively small amount of office building data. Although this seems like a great development, it presents one the main issues concerning building data collection: what data can be collected without affecting individual privacy?

5.2. Privacy Concerns

As Bilal et al.[3] and Linder et al.[40] have also recognised, the main pitfalls include security, privacy and protection of the building data. Privacy and the use of personal data has been a major concern during the recent years, and it has resulted in reformation of data protection regulations and creation of new legislation. The most noteworthy of the recent regulations is the General Data Protection Regulation (GDPR)[7] enforced by the European Union since 2018. GDPR creates restrictions on processing any information that by itself or in combinations with some other information can be used to identify a person and it gives the data subject the right to request and review any information related to them. The GDPR is restrictive when it comes to this personal data, but trying to circumvent the regulation by collecting only seemingly non-personally identifiable data is not necessarily the solution for individual privacy. There are no specific and exhaustive listings of what is and what is not considered as personal data, and some have suggested that the categorisation is already outdated and should be abandoned[45]. It has been shown[46, 47, 48] that it is possible to uniquely identify and profile individuals from datasets containing non-identifiable information by enriching the data with other non-personal or some publicly available information, indicating that the separation between personal and non-personal data is not obvious.

Related to smart buildings, Holm[49] describes three typical applications where seemingly trivial data is often used and how it can create privacy concerns. The first of these is the use of environmental sensor data and electricity consumption in optimising the energy-efficiency of the building. For instance, indoor air carbon dioxide level changes can be used for detecting occupancy in certain spaces. This information can be used to automatically decrease heating and ventilation when the space is unused. It also tells when the user of the space is present, and if the space has a specific user such as resident of an apartment or office user, the occupancy information is considered personal data. The same goes for electricity consumption in a certain space, as it may reveal what appliances are used at what times. By itself this information may seem trivial, but pairing it with additional information could create identifiable data.

The second case Holm brings up is related to the use of modern security systems that allow the use of personal keys and biometric information, such as fingerprints and facial recognition. Since the data used by these systems can be used to identify a person, it should be acknowledged that it is seen as personal data so the collection and processing should be done in accordance with the GDPR.

The third application is related to human health and how smart buildings can be used to steer tenants towards more healthy lifestyle. The use of mobile and wearable solutions like fitness trackers and smart phone applications that monitor user activity have become common in the recent years, but these mobile solutions can have limitations when it comes to monitoring person's overall health. Smart building sensors can be used to complement these limitations, and they can be used to e.g. record and examine snoring and breathing problems during sleep, monitor the quality of exhaled air and analyse human waste in an automated manner and produce overviews and visualisations of person's health. This uses and produces data that is personal and contains potentially sensitive information.

With these applications in mind, it can be said that almost any measurable data containing variables that are affected by human presence or interaction should be considered personal data if the space can in any way be linked to specific identities. It should be acknowledgedm that this introduces restrictions on collecting and processing the data.

5.3. Maintaining Building Data Privacy

All data collection operations should have legitimate and explicit purpose, and effective measures should be taken to protect the data and individual privacy. Cejka, Knorr & Kintzler[50] have identified the following measures that can be used to maintain individual privacy with smart appliances:

1. Data minimization
2. Opt-out
3. Explicit opt-in
4. Data anonymisation
5. Data aggregation
6. Minimal and local data storage
7. Data security and technical data protection
8. Data sovereignty and earmarking
9. Data obfuscation
10. Data control and customer incentive

Although their work concerned mainly smart meters in smart buildings, the practices can be applied to other data collection operations as well. Data minimisation means that only the minimal data required to provide the service or functionality should be collected. Sensor readings should be collected as rarely as possible to avoid forming additional information, e.g. occupancy information if the sensor's main purpose is to measure air quality. Building users should also have the possibility to refuse data collection as much as possible, and collecting any data that is not necessary for the systems to operate should require an additional agreement from the user. Living or working space customisation based on identified individual preferences should also not be made automatically without a separate consent from the user.

All data should be anonymised so that any identifying information is removed from it. If it is necessary for a service's required functionality to identify separate individuals from the data, it should be pseudonymised so that it takes more than reasonable effort to break the pseudonymity and identify the individual. Data should also be aggregated as soon as possible so that individual measurements can not be separated from it. If the purpose of data collection is to gain overview of the environment, information from individual sensors should only be used to calculate combined measuring points.

Depending on the type of the data, there may be legal obligations to store it for a fixed period of time, but typically building data should be stored only for the time it takes to achieve the objective of the data collection. Storing it for excessive time periods can increase the impact of a possible data theft while also increasing the cost of data storage. Local data storage should also be preferred for retaining the data to avoid transferring it unnecessarily and insecurely, and to keep the amount of parties processing the data to their minimum. If the building systems utilise centralised data collection platform, sufficient technical measures need to be implemented in order to protect the data against unauthorised access and modification. These measures should be upgradable and also maintain accountability so that it is possible to identify who has accessed or modified the data.

Complementing the building data with additional information or sharing it to third parties should require an explicit consent from the data subject. If it is necessary to have third parties access the data for e.g. research purposes, any sensitive information should be obfuscated so it becomes useless for malicious actors. Data subjects should be made aware of what data is being collected and who can access it. They should also be able to review the correctness of the collected information and remove any personally identifiable data. To maintain building users' interest in consenting to data collection, they should be provided with some type of incentive rather than keeping them as part of it out of necessity. One option is to present them overviews based on their data, that would allow them to monitor the data collection and see how their behavior affects different variables. These could include e.g. calculating ecological footprint or monetary value for electric energy consumption.

The GDPR introduces several restrictions for collecting personal data, and although previous practices help to operate in conformance to them, there are other aspects that need to taken into account as well. The regulation also requires one of six lawful basis to be fulfilled for collecting the personal data. These include 1) consent from the data subject, 2) contractual obligations with the data subject, 3) legal obligations to the legislative body, 4) necessity to protect life, 5) performing a task for public interest which has a definitive legal basis, or 6) legitimate interest. In addition, the GDPR recognises special categories of personal data that can reveal sensitive information about a person, including e.g. their ethnicity, political or religious views, genetic information and biometric data. Collecting and processing this data may require complying with additional regulations depending on the type of the information and the legislation of the country of operation.

For distributing responsibilities, GDPR recognises three types of roles in data collecting and processing organisations. Data Protection Officer (DPO) is an expert in their field who monitors the compliance with data protection regulation and provides advice to employees and management concerning it. Data controller is a person or a company who defines the purposes and practices for data collection and controls the collection and processing of it. Data processors process the data on behalf of the data controller. In smart building ecosystems, where there can be several companies collecting data from the environment and providing services for building users, challenges may arise concerning how these roles are distributed. When building data is collected in a centralised manner, there may be legal ambiguities on determining which stakeholder should assume the main responsibility[51]. Whatever the approach is, it is important to acknowledge this issue and define a comprehensive data governance strategy in which the responsibilities and obligations are fairly divided among all stakeholders.

6. KEKO SMART BUILDING ECOSYSTEM

In 2020, seven major companies and organisations in Finland began the development of KEKO Ecosystem[5], a platform for future smart buildings that would allow the collection and utilisation of building data by integrating facility management, housing systems, sensor networks and other building automation systems into a single ecosystem. The project consortium consists of KONE, Nokia, YIT, Caverion, Halton, VTT and Netox that each bring their own area of expertise to the project. As the main objective for the project was to define the requirements, features, and investigate different technologies that could be used in a complete, production-ready platform, the final KEKO ecosystem architecture is still taking its shape. This chapter describes the work-in-progress version of the environment, the KEKO Experimentation Platform (KEPPI), how it aims to improve the existing intelligent building environments and what changes may be coming in later stages of the project. An overview of the ecosystem is presented in figure 9.

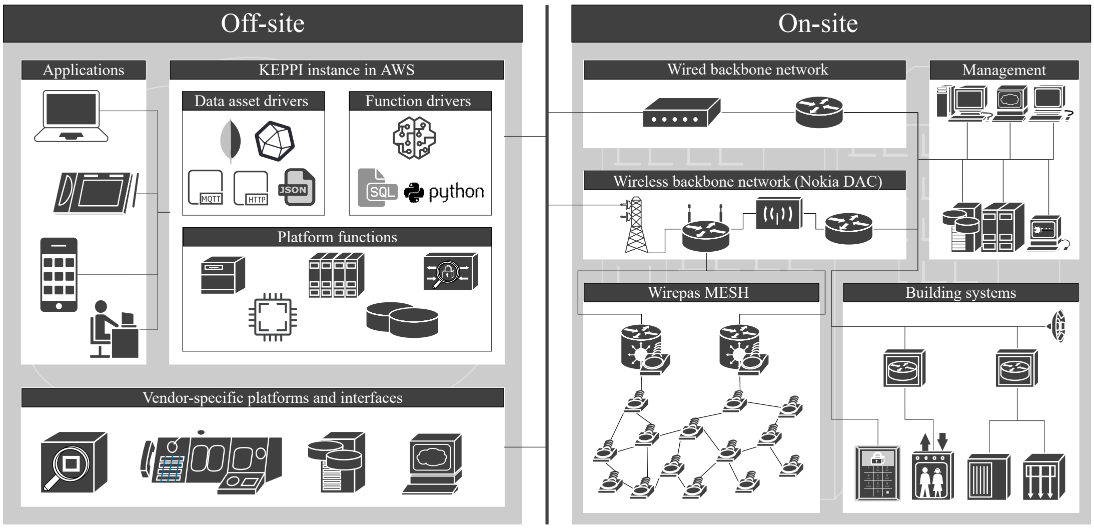


Figure 9. Overview of the ecosystem.

In chapter 7, MITRE ATT&CK is applied to this environment to identify the most relevant potential cyber threats, and although the threat modeling was performed on the current architecture of KEPPI, the changes that the final platform may undergo during its different development phases are taken into consideration. These results may also be used to guide the design process of the final ecosystem to identify the most relevant threat vectors and include the appropriate measures to mitigate the threats. In addition, a Security Information and Event Management system was implemented to monitor the building data collection of this microservice-based architecture of the platform in such a way that it can be applied in the final ecosystem as well.

6.1. KEKO Experimentation Platform, KEPPI

As mentioned in the beginning of the chapter, KEPPI is the work-in-progress version of the KEKO smart building ecosystem platform. The main functions of the platform are to collect and aggregate data streams from different building systems and to create interfaces so that the data can be accessed by local and remote services and applications. The platform allows vendors and admissible developers to create and run data processing pipelines on it using the data provided by the sensors, building systems and external sources to develop applications for improving the experience for end-users, e.g. tenants, building owners and facility managers. These services can include for instance real-time indoor navigating with digital signage, predicting and adjusting room temperature based on external weather data or estimating occupancy rates in different parts of the building.

6.2. System Design

KEPPI is built using WhereOS[52], a cloud operating system which enables rapid service integration and application prototyping using the actual building data sources in the environment. WhereOS is built on top of Apache Spark[53] and it uses the Spark stack together with Apache Hive[54] as the execution engines for the applications in the platform. The basic features of this stack enable the use of SQL queries to work with structured data and Spark DataFrames, but WhereOS extends these capabilities with the use of two types of drivers: function drivers and data asset drivers. Function drivers can be used to add new processing features to extend the standard SQL data managing capabilities, e.g. Python scripting and training machine learning model containers. Data asset drivers can be added to utilise additional data formats (e.g. GeoJSON, BIM file formats and sensor byte streams), external databases (e.g. MongoDB and InfluxDB) and data transmission protocols (e.g. HTTP and MQTT). KEPPI utilises several of these drivers for collecting data from building systems, storing it in a suitable format, processing it by training and utilising machine learning models, and creating APIs for providing the data to be used externally.

Although the KEPPI platform itself is run on a single remote server instance, it is designed using a container-based microservice architecture. Containers are discrete virtualised runtime environments that are run on top of a single operating system. They are a lightweight, scalable and rapidly deployable solution for isolating separate services when compared to other virtualization methods[55]. Essentially, KEPPI is a collection of containerised applications running on a single server, each of them providing different services which are responsible for either some specific function of the platform itself, e.g. user authentication or logging, or processing the operational data received from some building systems and external sources. This container-based microservice architecture offers scalability and expandability to the platform and allows it to be run in a distributed manner.

The primary method for transmitting data in and out of KEPPI is via its SSL encrypted HTTP APIs. The outbound APIs expose endpoints for reading various information and sensor measurements, e.g. floorplan, rooms, indoor air quality measurements, elevator status, energy usage and access control rules from the ecosystem with HTTP requests. Inbound APIs are used to bring new data sources or to integrate tenant service applications to the platform. Developers can create additional endpoints for their own containerised services to share and receive data using the API.

Data sources that do not produce streaming data are integrated to the system using these APIs.

The second method for transferring data is using a MQTT message broker. MQTT (Message Queuing Telemetry Transport) is a publish-subscribe protocol that can be used to stream real-time data over network. The protocol is often used with WSN and it can be run over TCP/IP network. In the target ecosystem, streaming data sources like sensors and other measurement devices are integrated using an on-premise sink device on the edge of the network and a MQTT message broker component in KEPPI. Figure 10 visualises the operations of the platform.

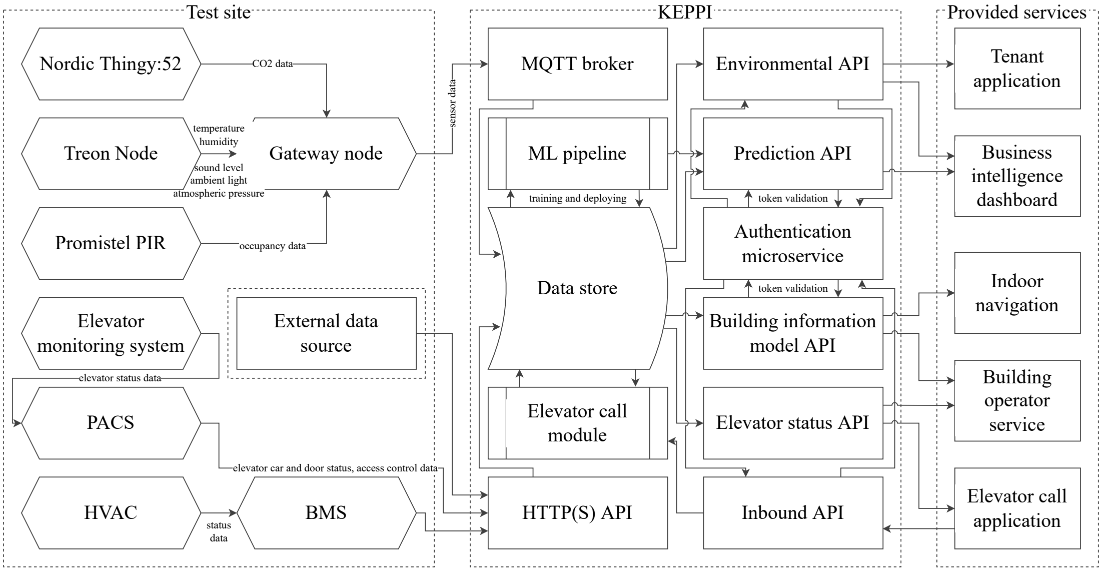


Figure 10. Diagram of the ecosystem.

6.3. Sensors and Data Collection

The building and sensor data consists mainly of time series measurement data collected from the on-premise WSN. This network consists of Treon Nodes, Nordic Thingy:52's and Promistel PIR sensors that are using Wirepas Mesh networking protocol. The protocol has similarities with Bluetooth LE and it has support for transmitting and receiving BLE Beacon messages, but unlike BLE Mesh which uses flooding, Wirepas Mesh uses cost-based routing for communication between nodes. The experimental KEKO site contains also Nokia Digital Automation Cloud (DAC), a private LTE network that can be used to create sensor networks with wider coverage area. In the test site, the DAC is used as a wireless backbone network.

In the sensor network, nodes transmit their measurement data wirelessly to gateway nodes located on the edge of the mesh network. Gateways then send MQTT publish messages with the most recent information from each node every five minutes to the back-end systems using the local private LTE network. After KEPPI receives the data, it parses the message and stores it in InfluxDB running on a separate container on the same server instance. Other services and pipelines running in the ecosystem can then query this data from the database and use it in their operations. The same database also contains, at least from the monitoring point of view, the more interesting measurement data that is provided by HVAC and BMS. This data may be more interesting because the building systems can utilise the same measurements in their control loops to alter their operations as well. Building systems can provide the measurement data to KEPPI by using MQTT message brokers similarly to sensor networks, inbound or outbound HTTP endpoints, or if they are using proprietary protocols or data formats, they might require a specialised driver in the platform to fetch the data.

7. ASSESSMENT

As described in the chapter 2, the threat modeling approach used in this work began with characterization of the system. The container-based modular architecture presented in section 6.2 is run on a remote server hosted by a third party hosting provider, so it is important to recognise the presence of both the local OT networks of automation systems and the hybrid IT architecture of management systems and KEPPI in this ecosystem. The asset identification and threat assessment is done with the help of MITRE ATT&CK for ICS[17], MITRE ATT&CK for Enterprise[6] and ENISA Baseline Security Recommendations for IoT in the Context of Critical Information Infrastructures[37].

7.1. Identifying Assets and Access Points

For identifying the valuable assets in the target environment, IoT asset taxonomy proposed by ENISA[37] is used as the starting point. Based on their asset group taxonomy, ICS asset listing in MITRE ATT&CK for ICS[17] and KEPPI system architecture, the relevant asset groups listed in table 5 can be defined. These assets may act as an intermediate or final targets for adversaries depending on their motivation and objective.

Building automation and sensor devices form the interface between real-world events and digital space by sensing or affecting some physical world attributes. In this ecosystem, this category contains automation systems introduced in chapter 3, like HVAC , PACS & Philips Hue lighting systems and sensor networks consisting of sensor nodes like Nordic Thingy:52, Promistel PIR and Treon Node. Promistel PIR is a motion sensor that is used to detect occupancy of certain area. It is compatible with the Wirepas Mesh network. Thingy:52 is a multi-sensor developed by Nordic Semiconductor that can measure temperature, humidity, air pressure, light, color, orientation, motion and CO2 concentration level. In addition, the sensor has a digital microphone and it offers Bluetooth connectivity and is configurable over-the-air (OTA) using its Bluetooth API. Treon Nodes offer similar measurement capabilities with additional HAL-sensor for measuring magnet proximity. Together with Treon Gateways, these nodes form the Wirepas Mesh network that is used to connect all sensors in the test site area to the backbone network.

Communications category includes the different networks in the ecosystem. In the KEPPI environment, this includes all the wired networks used by building systems, the WSN using Wirepas Mesh protocol and Nokia Digital Automation Cloud (DAC) which provides a wireless backbone with private LTE network for the sensors.

The infrastructure group includes routers, gateways and other network devices used to create different networks in the environment. This includes Nokia DAC access points, edge server and Treon Gateways which form and connect the backbone Wirepas Mesh network to KEPPI platform.

The platform and back-end group consists of web-based services and the cloud infrastructure required to operate them. In the experimentation platform, these include the Amazon Web Services Elastic Computing node running the different parts of the platform, WhereOS instance managing the drivers and the KEPPI instance itself.

The decision making group includes the inbound APIs that bring external data to KEPPI and data pipelines that process data that is used in automating some building systems (e.g. temperature, humidity).

Applications and services include all KEPPI APIs, and third party applications using them.

The information group consists of all the data that the platform collects from the building sensors, receives from external sources and produces from data processing pipelines. Some examples of this in KEPPI are measurements from temperature and humidity sensors, weather forecasts from some external providers, and occupancy predictions from data processing pipeline.

Table 5. Asset groups in smart building ecosystem

Asset group	Description	Ecosystem components
Automation devices & sensors	housing system hardware and software components, operator devices, controllers, sensors and actuators	BMS, HVAC, Nordic Thingy:52, Treon Nodes, Philips Hue, Promistel PIR
Communications	networks and protocols	Bluetooth LE, Wirepas Mesh, private LTE network (NDAC)
Infrastructure	network devices, e.g. routers, switches, gateways and protocol translators	Treon Gateways, NDAC radio access points & edge server
Platform & back-end	web services, user interfaces, cloud infrastructure	KEPPI instance, web UI, WhereOS, AWS instance
Decision making	data processing pipelines and control loops	KEPPI environmental data processing
Applications & Services	data analytic and visualisation systems, tenant services, management interfaces	KEPPI outbound APIs, data processing pipelines
Information	local and remote data historians, logs, sensor measurements	KEPPI sensor data, container logs, AWS logs

After the valuable assets have been recognised, the next step is to recognise how and why bad actors might show interest in them.

7.2. Applying ATT&CK

As described in the section 2.1.1, ATT&CK organizes actions taken by malicious parties into Tactics, Techniques and Procedures. Tactic is the high level milestone of an attacker which can be achieved by performing different techniques that consist of technical procedures, like exploiting a vulnerability in a used software component.

7.2.1. Techniques

The assessment with ATT&CK begins by going through relevant techniques one at a time, identifying assets relevant to it and recognising the coverage of it in the monitoring process. Assessing the whole environment like this requires extensive resources and knowledge of the environment, so in the scope of this work, some compromises must be made. The role of building automation systems has typically been the initial access point into the system environment, so the focus in this work is in ATT&CK's tactic Initial Access (TA0108). It consists of techniques used by adversaries to gain the initial foothold in an environment. Successful detection and mitigation of these techniques can prevent any further system abuse making this potentially the most critical stage of any offense against a system environment. ATT&CK for ICS includes the following techniques for this stage:

- Drive-by Compromise (T0817)
- Exploit Public-Facing Application (T0819)
- Exploitation of Remote Services (T0866)
- External Remote Services (T0822)
- Internet Accessible Device (T0883)
- Remote Services (T0886)
- Replication Through Removable Media (T0847)
- Rogue Master (T0848)
- Spearphishing Attachment (T0865)
- Supply Chain Compromise (T0862)
- Transient Cyber Asset (T0864)
- Wireless Compromise (T0860)

In addition, the following techniques from ATT&CK for Enterprise can be recognised as relevant for networked building system environments:

- Hardware Additions (T1200)
- Trusted Relationship (T1199)

- Valid Accounts (T1078)

In total, MITRE ATT&CK can be used to identify 15 techniques that can be used to gain the initial foothold. The framework also presents information on how these techniques can be mitigated and detected. As discussed in chapter 4.1, there are five primary attack surfaces in smart buildings: building users, remote access, physical access, integration platform and wireless access. Building users represent different groups of people, including tenants utilising smart building services, operators managing the building and appliance vendor personnel with elevated privileges to some systems. Physical and wireless access contains the physical equipment and local wireless networks of the environment, and remote access consists of the interfaces and services that can be used via networks. Integration platform represents the KEPPI instance and all local and remote components required to run it. When these attack surfaces are combined with different asset groups present in the ecosystem, it becomes apparent that there are several attack paths for adversaries to take to compromise the ecosystem as visualised in Figure 11. Figures 12-25 present how these attack paths are leveraged in each corresponding ATT&CK technique.

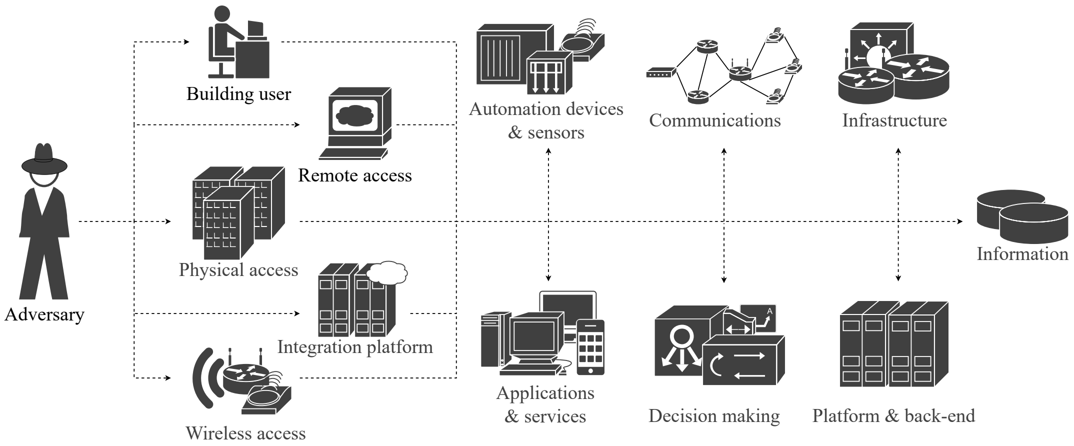


Figure 11. Attack paths in smart building ecosystems.

The first of these is drive-by compromise (T0817), which can occur if a user visits a malicious website during a regular browsing session. This technique targets human operable devices, so in the context of this environment, potential targets include management workstations connected to internet. The technique can be mitigated by isolating application processes in the target devices (M0948), using exploit protection mechanisms to detect and block software exploitation (M0950), restricting use of web content like JavaScript and browser extensions (M0921), and keeping the software updated (M0951). To detect this technique, ATT&CK suggests four useful data sources: application (DS0015), file creation (DS0022), network traffic (DS0029) and process creation logs (DS0009). The technique can also be used for attempting to gain access via trusted third parties.

Public-facing applications (T0819) can contain weaknesses that attackers may try to exploit to gain access. The attack surface should be minimised by exposing only the required applications, but since KEPPI uses exposed web APIs as its main method for publishing and receiving data, there are several exposed services that are required by the platform making this technique highly relevant. The targets

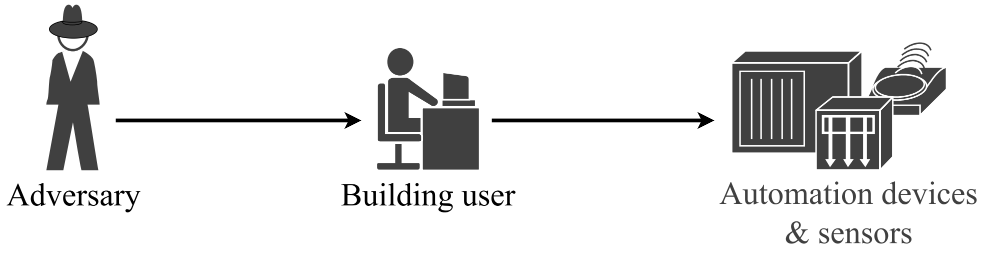


Figure 12. Drive-by-compromise (T0817).

for this include automation device remote management interfaces, network device management interfaces, KEPPI platform and API, and unintentionally exposed back-end services. Mitigations for this technique include isolating application processes (M0948), exploit protection (M0950) i.e. using web application firewall (WAF) to filter the HTTP traffic, segmenting the public-facing servers from the rest of the network (M0930), keeping systems updated (M0951), scanning the public-facing systems regularly for vulnerabilities (M0916) and following the least-privilege policy for service account management (M0926). Monitoring application logs (DS0015) and network traffic logs (DS0029) can be used to detect this technique.

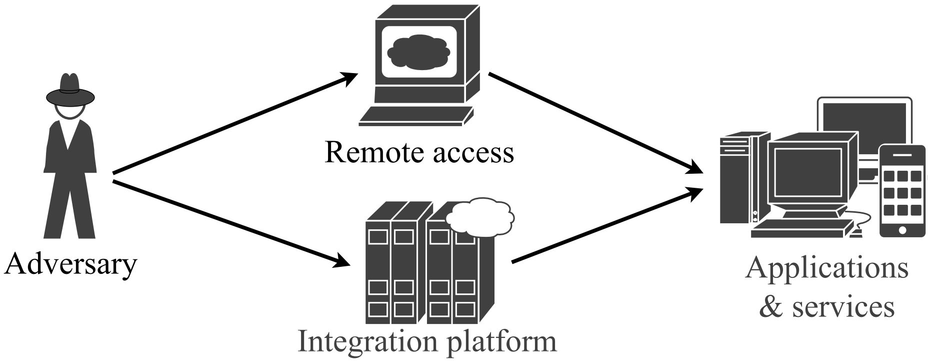


Figure 13. Public-facing applications (T0819).

Due to its distributed architecture, some components of the KEPPI platform are run on remote systems. Malicious actors can attempt to abuse them to gain initial access or move laterally within the environment. Exploitation of remote services (T0866) can target the platform's remote data historians, infrastructure management systems, application servers operating as middleware between client applications and KEPPI platform, and other remote services used to interact with systems remotely. Some mitigations to this exist, including application isolation (M0916), disabling unnecessary features (M0942), identifying exploitation behaviour with security applications (M0950), network segmentation (MM0930), keeping permissions and access minimised (M0926), collecting CTI (cyber threat intelligence) to identify potential threats (M0919), keeping systems updated (M0951), and scanning the systems for known vulnerabilities and exposed services regularly (M0916). Network traffic (DS0029) and application logs (DS0015) can help to detect this type of abuse.

External remote services (T0822) allow users to connect to internal networks from external locations. For attackers, these can provide a path for accessing internal network resources from external networks. An example of this could be a VPN access granted to building appliance vendor for remote maintenance purposes. The

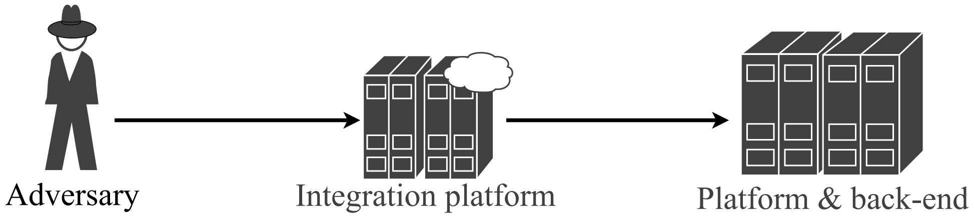


Figure 14. Exploitation of remote services (T0866).

mitigations include account use policies (M0936), disabling or removing rarely used features and programs (M0942), limiting access to resources over network (M0935), enforcing secure password policies (M0927) and multi-factor authentication within organisation and with partners (M0932), segmenting network (M0930), and making sure that access and user accounts are managed actively (M0918). This technique can be detected by monitoring the application logs (DS0015), network traffic (DS0029), and logon sessions (DS0028) for relevant resources.

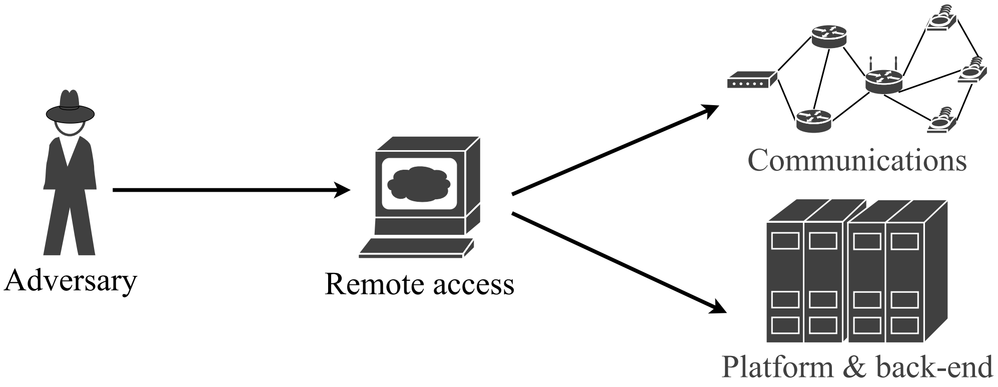


Figure 15. External remote services (T0822).

Another potential way for attackers to gain access in internal networks is by abusing directly internet accessible devices (T0883). Some systems in the building may require remote access for maintenance purposes, and if they are connected without adequate protection mechanisms, they may become the initial point of access for malicious parties. This technique concerns automation devices and infrastructure systems. The main mitigation for this technique is to segmenting the network (M0930) so that internal systems cannot be accesses remotely and auditing that there truly are none of these systems exposed. To detect this technique, logon session metadata (DS0028) and network traffic flow (DS0029) can be used as data sources.

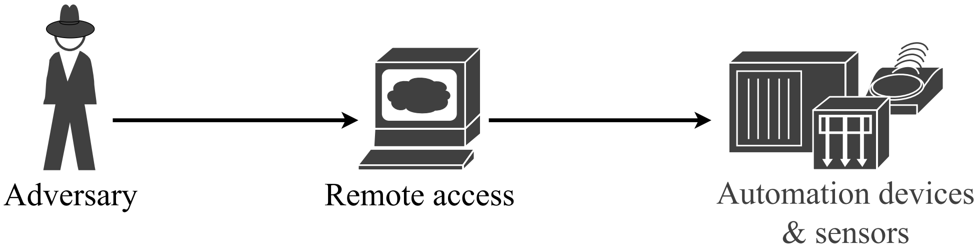


Figure 16. Internet accessible devices (T0833).

Similarly to external remote services (T0822), remote services (T0886) in internal networks are often used to interact with systems remotely. These services include e.g. SSH, RDP, and SMB, and attackers may try to take advantage of inadequately configured or outdated services to access internal systems. In the KEPPI platform, this technique concerns the platform's components that are running remotely accessible services, e.g. back-end systems, remote data historians, and operator workstations. Mechanisms like access management (M0801), authorization enforcement (M0800), network traffic filtering (M0937), human user authorization (M0804), network allowlists (M0807), network segmentation (M0930), enforcing password policies (M0927), process and device authentication (M0813), and user account management (M0918) can be used to mitigate the technique. For detecting it, several data sources can be used. These are command execution logs (DS0017), logon session creations (DS0028), network share access logs (DS0033), network traffic flows (DS0029), and process creation logs (DS0009).

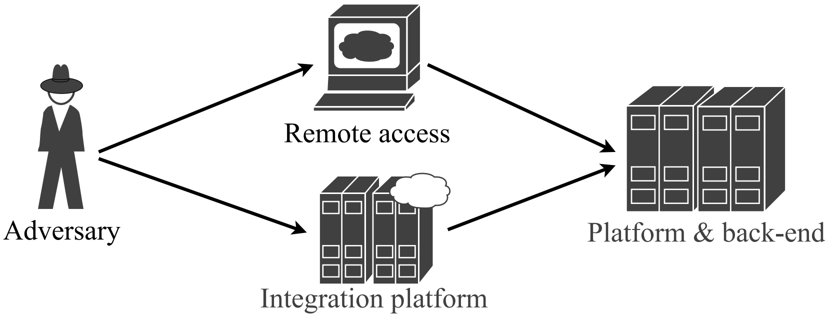


Figure 17. Remote services (T0886).

Replication through removable media (T0847) is a technique used to access systems and networks separated from the IT networks by placing malware to removable media either manually or automatically from an infected system, and inserting it to a system in the separated environment. This technique can be used to infect systems even in air-gapped networks, but it requires a user with physical access to the system. This can be achieved with e.g. social engineering techniques on personnel. In the building environment, this technique concerns automation systems, operator workstations and infrastructure devices, and it can be mitigated by disabling automated execution of software on removable media (M0942), limiting physical access to the systems (M0934), and preventing the use of removable media on these systems with OS configurations (M0928). The technique can be detected by monitoring drive creation (DS0016), file access and creation (DS0022), and process creation logs (DS0009).

Adversaries can introduce rogue masters (T0848) to automation system environment to impersonate legitimate control devices and disrupt communications between actual devices. Rogue systems can be used to capture network traffic, send false data or disrupt communication. In networked building automation environment, this technique can target the automation systems, their communication channels, network infrastructure devices, and sending fraudulent data can affect decision making process. The mitigations for this technique include using secure network protocols that verify the authenticity and integrity of communications (M0802), filtering network traffic (M0937) and using network allowlists (M0807) so that only authorised control messaged from known hosts are accepted, segmenting the network (M0930), and

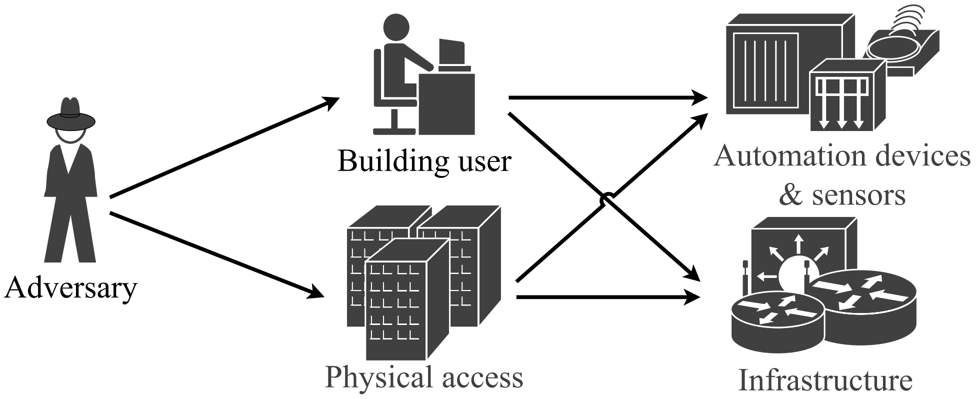


Figure 18. Replication through removable media (T0847).

authenticating all messages between the devices (M0813). Application logs (DS0015), network traffic (DS0029), and event alarms from operational databases (DS0040) can be used to detect this technique.

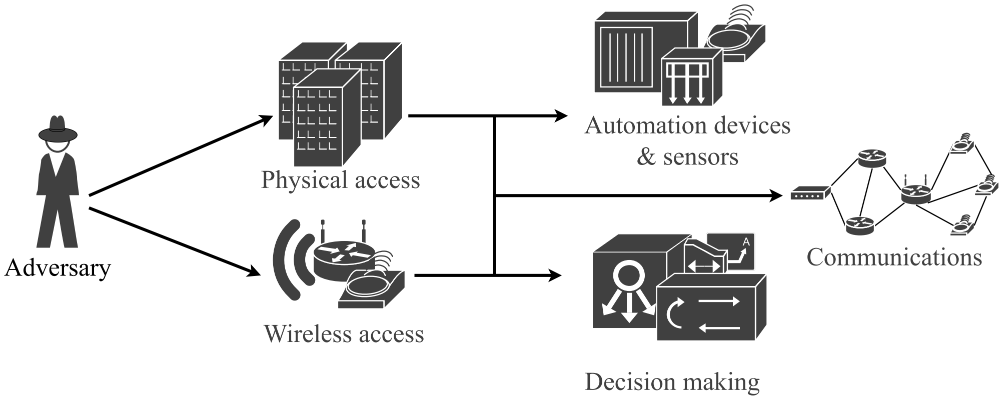


Figure 19. Rogue masters (T0848).

Spearphishing attachment (T0865) is a social engineering technique that consists of sending malware as an email attachment to specifically selected person or group of people, who open and execute the attachment in a vulnerable system. Similarly to drive-by compromise, this technique targets human operable devices so it concerns the management and operator workstations connected to internet. Mitigations include using antimalware software on the devices (M0949), intrusion prevention systems to block malicious traffic (M0931), restricting access to email and web-based content (M0921), and training users to identify social engineering techniques (M0917). Application (DS0015) and network traffic logs (DS0029) can be used to help detecting this technique.

Supply chain compromise (T0862) is a technique in which bad actors compromise or replace some component of the target system before it is introduced to the target environment. The component can be either hardware or software based, and the compromise can occur at any stage during the supply chain. This technique concerns primarily automation devices and their firmware and updates, communications protocols and their dependent libraries, infrastructure systems, back-end services with their dependencies, and applications and services using third party libraries. Although there may not be simple and effective responses against this technique, it can be

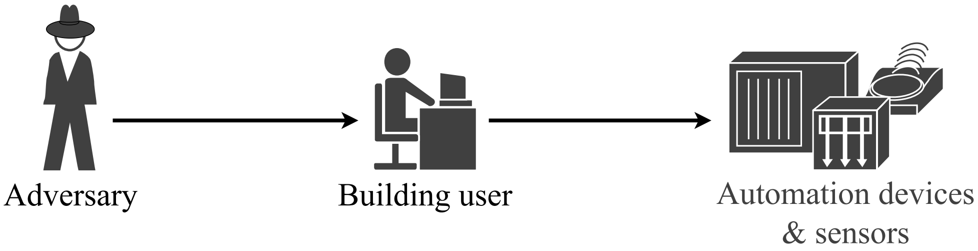


Figure 20. Spearphishing attachment (T0865).

mitigated to some extent by performing comprehensive audits in the environment (M0947), utilising digital signatures to verify the integrity of components (M0945), enforcing a supply management program to validate the trustworthiness of parties in the supply chain (M0817), defining a patch management process for keeping the systems updated (M0951), and scanning the systems regularly for vulnerabilities (M0916). Detecting manipulated systems can be challenging, but some methods for it include verifying the integrity of distributed binaries by comparing their hashes and signatures to ones received directly from the software provider (DS0022), and comparing the behaviour of the component against known baseline behaviour (DS0013).

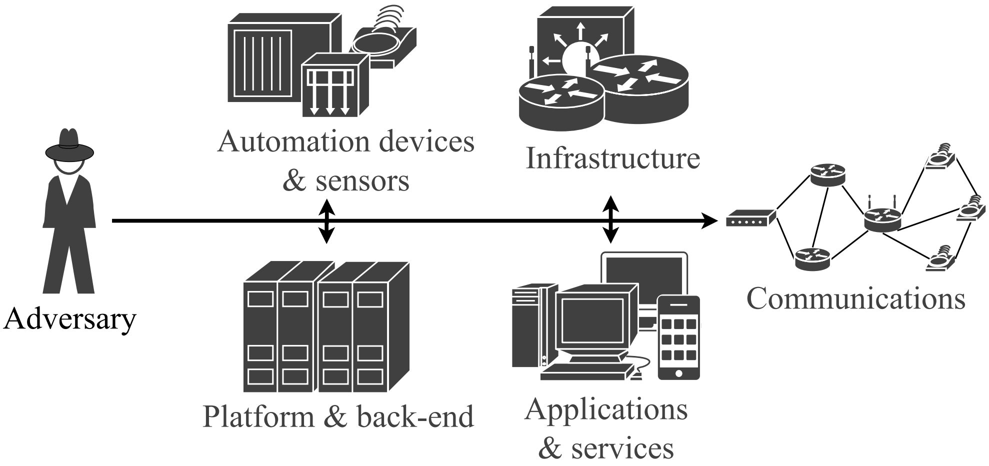


Figure 21. Supply chain compromise (T0862).

Transient cyber assets (T0864) are systems that are used in networks temporarily, and they are typically introduced to the environment by authorised personnel. The systems can include e.g. maintenance laptops and tablets used to locally perform configurations and updates to automation systems that are not permanently connected to external networks and do not offer remote maintenance interfaces. If these devices are used in other environments, they may become compromised leading to unauthorised access to building system network or leakage of operational information. The threat to the smart building environment can be mitigated by using antimalware solutions on these devices (M0949), verifying the integrity of them by using e.g. Trusted Paltform Module (TPM) (M0947), encrypting the storage on the devices to prevent information leakage (M0941), segmenting the network (M0930) and keeping the systems updated (M0951). ATT&CK does not recognise specific data sources that could be used to detect this type of technique, but based on the similarity to some other techniques, application logs (DS0015), network traffic (DS0029) and operational databases (DS0040) could potentially help in this effort.

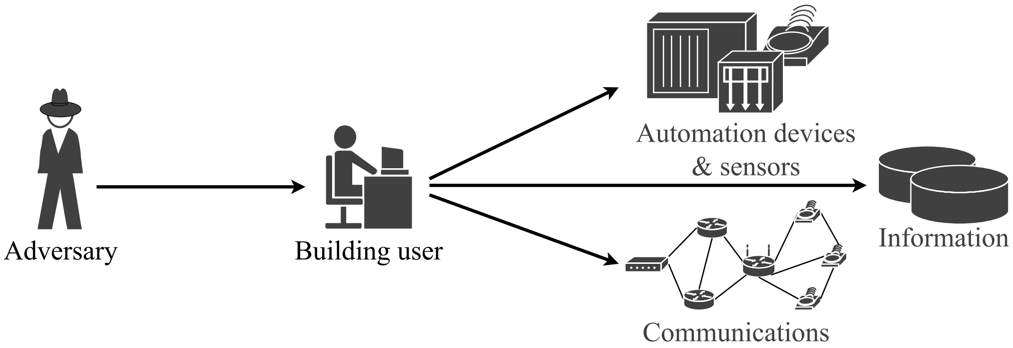


Figure 22. Transient cyber assets (T0864).

Wireless compromise (T0860) is a technique where malicious actors compromise a wireless device or exploit insecure wireless communication protocol to gain unauthorised access to a wireless network. Since smart building environments can contain multiple wireless networks, this technique can be considered highly relevant. Bad actors can use the access to communication channels to issue malicious commands or send falsified data to other systems. To mitigate this technique, the authenticity of communications should be verified by utilising secure network protocols (M0802) and requiring device authentication (M0813) instead of relying blindly on the received information and commands, encrypting the network traffic to prevent eavesdropping (M0808) and minimising the wireless signal propagation area (M0806). Monitoring application logs (DS0015), logon session information (DS0028) and network traffic flow (DS0029) can aid in detecting this technique.

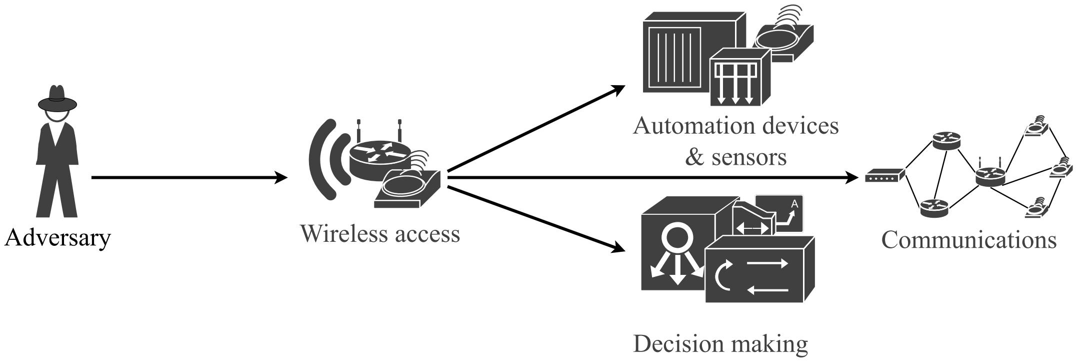


Figure 23. Wireless compromise (T0860).

Similarly to removable media, malicious parties can use hardware additions (T1200) to gain access to a target environment. They can include malicious computer accessories and networking devices, and unlike the removable media, these additions can have computing or networking capabilities. They can be used to add abusable functionalities to automation systems, operator workstations and network infrastructure devices and bad actors can use them to e.g. inject keystrokes, eavesdrop and modify network communications or access systems remotely from unmonitored network. The ATT&CK for Enterprise recognises two mitigations for this technique, including limiting hardware installation (M1034) and access to resources over network

(M1035). However, in smart building environments it is also important to recognise that limiting physical access (M0934) can also be a viable method for preventing this technique. The data sources that can be used in detection include application logs (DS0015), drive creation logs (DS0016) and network traffic flows (DS0029).

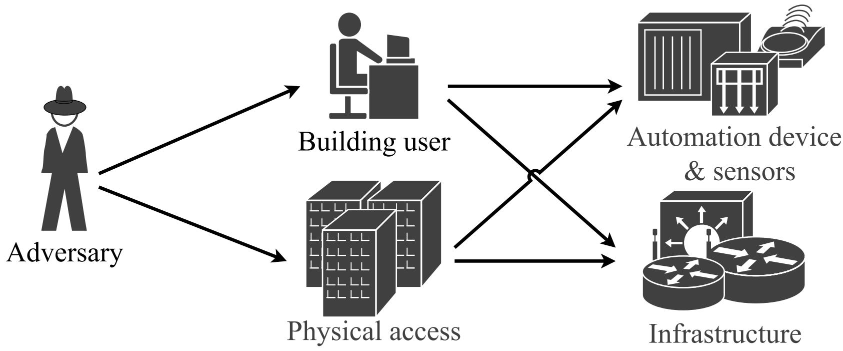


Figure 24. Hardware additions (T1200).

There can be a large group of organisations and companies involved in a smart building ecosystems. Bad actors may try to abuse the trusted relationship (T1199) between them to gain access to a target environment by leveraging valid accounts (T1078) that any of these parties may have. Service contractors and housing appliance vendors can often have elevated access permissions to maintain and manage the automation systems, but if their credentials for accessing the building platform becomes compromised, they can be abused by malicious parties to access systems in the internal network. This can be potentially mitigated by segmenting the network (M1030), keeping control over the user accounts used by third parties (M1052), ensuring that systems store credentials securely (M1013), changing default credentials (M1027), reviewing the permissions of different accounts regularly (M1026), removing unused accounts (M1018) and training users on secure principles (M1017). User authentication logs (DS0002), application logs for management systems (DS0015) and logon session information (DS0028) can be used as data sources for detecting this technique.

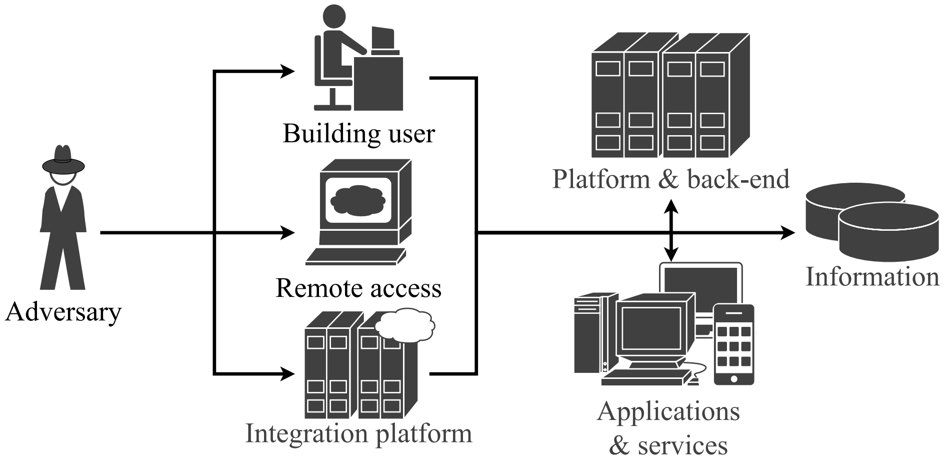


Figure 25. Trusted relationship (T1199) and valid accounts (T1078).

A summary of how these techniques are related to asset groups and what data sources can be used to detect them is presented in table 6.

Table 6. ATT&CK Initial Access Techniques related to asset groups.

Asset group	Relevant Techniques	Detection sources
Automation devices	Exploit Public-Facing Application (T0819) Exploitation of Remote Services (T0866) Internet Accessible Device (T0883) Replication Through Removable Media (T0847) Rogue Master (T0848) Supply Chain Compromise (T0862) Transient Cyber Asset (T0864) Wireless compromise (T0860) Hardware Additions (T1200)	DS0009, DS0013, DS0015, DS0016, DS0019, DS0022, DS0028, DS0029, DS0040
Communications	External Remote Services (T0822) Rogue Master (T0848) Supply Chain Compromise (T0862) Wireless compromise (T0860)	DS0013, DS0015, DS0022, DS0028, DS0029, DS0040
Infrastructure	Exploit Public-Facing Application (T0819) External Remote Services (T0822) Internet Accessible Device (T0883) Rogue Master (T0848) Supply Chain Compromise (T0862) Wireless compromise (T0860) Hardware Additions (T1200)	DS0013, DS0015, DS0016, DS0019, DS0022, DS0028, DS0029, DS0040
Platform & back-end	Exploit Public-Facing Application (T0819) Exploitation of Remote Services (T0866) Remote Services (T0886) Supply Chain Compromise (T0862) Trusted Relationship (T1199) Valid Accounts (T1078)	DS0002, DS0009, DS0013, DS0015, DS0019, DS0022, DS0028, DS0029, DS0033
Decision making	Rogue Master (T0848)	DS0015, DS0029, DS0040
Applications & Services	Drive-by Compromise (T0817) Exploit Public-Facing Application (T0819) Exploitation of Remote Services (T0866) Remote Services (T0886) Replication through removable media (T0847) Spearphishing Attachment (T0865) Supply Chain Compromise (T0862) Hardware Additions (T1200) Trusted Relationship (T1199) Valid Accounts (T1078)	DS0002, DS0009, DS0013, DS0015, DS0016, DS0019, DS0022, DS0028, DS0029
Information	External Remote Services (T0822) Remote Services (T0886)	DS0015, DS0028, DS0029

7.2.2. Summary and Observations

Based on the summary presented in table 6, two data source categories are present in detecting techniques against any asset group: application logs (DS0015) and network traffic (DS0029). Application logs can contain information about various metrics, errors and alerts, and they are provided by the applications themselves instead of the operating system or the platform they are running on. Network traffic log contains information about the data transmission taking place across a network either in a summarised form as a netflow or as a captured raw data. This information an be provided by the network devices or separate network sensors. It makes sense that these data sources are important, because application-specific logs can provide detailed information that can be useful in detecting whether systems are behaving as expected and network flow can reveal if any malicious traffic is taking place in the environment. At its current state, there was no data collection or monitoring implemented for either of these data source types, and adding them afterwards would have required extensive access to the system environment which was not possible due to the platform being already operational and being used by other members of the project. For future stages of the ecosystem, this deficiency should be acknowledged.

In addition to application logs and network traffic, there are also other data sources that should be utilised to form a comprehensive monitoring implementation for the automation environment. These include information on running processes (DS0009), sensor health telemetry data (DS0013), drive creation and modification logs (DS0016), service and daemon execution logs (DS0019), file creation and access logs (DS0022), logon session creation logs (DS0028), and operational databases (DS0040). With the exception of process creation, the same data sources can also be recognised to be useful in detecting incidents concerning the communications and the network infrastructure devices. Sensor health information would be a useful data source for detecting misbehaving field equipment. This information is typically stored in the systems own operational database, but it could also be provided to KEPPI similarly to sensor measurements. The operational databases of the WSN and automation systems can also contain other information that would be beneficial in detecting security incidents, such as the asset catalog of legitimate devices and the security keys that the Wirepas Mesh uses to authenticate all sensors. At its current state, KEPPI did not have access to this information, so a deeper integration of these operational databases and the KEPPI platform should be considered.

Collecting data on logins, processes, daemons, files and drives requires running separate logging components on the automation systems that would be capable to transmit all the relevant events to the monitoring system. In enterprise environments, the typical method for achieving this is running a log shipper on each endpoint device which sends the event data directly to a log collector component of the SIEM system. This approach can be applied for the operator workstations and management servers, but it requires access to all of these systems and the permissions to run the logging component. In the smart building ecosystem, KEPPI could potentially act as the log collector, which would also allow the data to be used to improve the platform's security event pipeline. These capabilities did not exist in the current ecosystem, but they should be implemented for future designs.

For detecting incidents concerning the back-end infrastructure and services provided by it, information should also be collected from user logins (DS0002) and remote storage resources (DS0033). Since user authentication to services provided by the KEPPI platform is handled by a service running on the same server, collecting authentication events for it should be simple to implement. KEPPI could also be used as the log collector to provide this information to the monitoring infrastructure. The server event logs should also be collected to detect unauthorised access and usage on the server running the integration platform, but these events should be sent directly to the log collector component of the monitoring system. If the server running KEPPI becomes compromised, adversaries could modify and disturb the event collection. For other back-end systems and the cloud infrastructure, it is also possible to utilise existing tools provided by the cloud platform. Since KEPPI is running on AWS cloud platform, the monitoring methods provided by it should be added to the monitoring implementation to gain visibility of the system environment. AWS CloudTrail[56] creates audit log containing information of all user activities, management events and performance metrics. It should be kept in mind, that if the finished KEKO instance is operated either in local environment or on a different provider's platform, the available tools may change. Whatever the case may be, the back-end infrastructure should be included to the monitoring scope.

Addressing the issues described here after the platform is made operational is challenging and they should be acknowledged alredy in the design phase. The following section describes one proposed method that can however improve the visibility in the platform operations at some scale by implementing a security event pipeline within the platform and integrating it to a SIEM system.

7.3. SIEM integration

Security Information and Event Management (SIEM) systems have been traditionally used in security operations centers (SOC) to monitor and manage security-related events and information received regarding different assets in an enterprise IT network environment. These assets include workstations, different types of servers, networking devices and data stores and they are monitored using network sensors and log collectors. Flow and log collectors normalise and aggregate the data collected from various sources and transmit it to event processor which analyses the data based on a predefined set of rules. If events matching some rules are discovered, the system executes a response action that is defined along the rule by e.g. notifying SOC analyst of a possible adversarial behaviour. Events and logs are then stored for a predetermined time.

SIEM system is one of the most important tools in SOC, and its capabilities are often the defining element when measuring the effectiveness of a SOC[57]. Modern SIEM systems are capable to collect and process large amounts of data and perform analysis based on detected behaviour patterns, so usage of additional, non-standard data sources could improve the detection capabilities of SOCs. With critical infrastructure and smart grid environments, improving the situational threat awareness by implementing a SIEM system has been studied by others previously[58], so it might be possible to apply a similar system for smart building ecosystem as well. The following sections describe the implemented monitoring interface for KEPPI and how it is integrated with an IBM QRadar SIEM system for collecting and processing security event data.

7.3.1. KEPPI Monitoring Module

At its current state, the ecosystem could potentially provide system logs from the back-end systems operating in the system and from the server instance usage. This would require a full access to the back-end systems for configuring them, but since the platform was already operational and providing services to other consortium partners, gaining such access was not possible. These log sources could be used to detect threats originating from the IT systems in the ecosystem, but they are not necessarily required to detect threats concerning the operational system components. To enable the detection of possible OT asset misuse incidents in the ecosystem, a security event collection component was implemented in KEPPI. The implementation began by designing the main requirements for the module, which included the following:

1. validating data produced by sensors and building systems
2. validating data received from tenant applications
3. identifying anomalous data
4. creating events from detected anomalies
5. storing the events for predefined time period
6. providing event information as time series data

The most interesting data that was already produced by the building systems was environmental measurements provided by the local WSN. Since these measurements can be used to automatically adjust the ventilation and air conditioning in the building, falsified information could cause e.g. overheating the indoor air or even lead to system malfunction. The first challenge in designing the monitoring component comes with the validation of this data and at what point in the processing pipeline it should be done. Because the monitoring component is added on top of the existing prototype environment already providing data for other services, modifying existing pipelines is not an option. With a separate event collection pipeline, it is possible to query entries from the measurement database based on predefined conditions or use some advanced anomaly detection algorithms for examining the time series data and create events if anomalous patterns are identified. In this monitoring module prototype, the incident detection is based on testing if the measurements exceed predefined threshold values or do not comply with the expected data format. If relevant outliers are identified, an event is created.

The purpose for the second function is to validate the data received from HTTP APIs used for external data sources and tenant applications. The platform allows vendors and building owners to create APIs that can receive data or commands from external sources, but this also means that these APIs can be misused. The KEPPI HTTP API driver uses token-based authentication, meaning that all requests to the

API must contain a valid alphanumeric string. At its current state, the platform trusts requests which provide a valid authentication token. However, if some of these tokens were to become compromised, attackers could instrument malicious requests and send them to the API successfully. Since the token validation is performed by the back-end containers operating the HTTP API driver, authentication logs can not be used as a data source without access to the back-end instance. This means that authentication logs can not be used here for identifying security events. In addition, separating maliciously crafted API calls from valid ones by using KEPPI data alone is challenging. To add this functionality, the contents of incoming API requests are checked so that 1) they contain only the required data fields, and 2) the data itself follows the expected format (e.g. alphanumeric string or timestamp). If anomalous requests are identified, an event is created and stored on the platform.

After an event is detected, it is stored in an InfluxDB database. Each entry contains the timestamp of the event, what type of event has occured and information on the affected asset. To use the event data with external systems, it needs to be accessible outside the ecosystem. As described in section 6.2, the primary method for transmitting data out of KEPPI is via HTTP API, and since the used SIEM has some existing support for fetching information from HTTP REST APIs, it was selected to be the method for transmitting event data from the platform to the SIEM system. This functionality was added by using the platform's HTTP API driver, which also enabled the use of same token-based authentication as other data exposing APIs on the platform. The event API requires two parameters to present event data from specified time interval: start_time and end_time. When these timestamps are defined in the HTTP request, events that have timestamps between start_time and end_time are returned using JSON format. An example of the API response is presented in listing 7.1.

```json
[
  {
    "time": "2021-06-01T11:15:00.000Z",
    "asset_type": "sensor",
    "asset_id": "A127622",
    "event_source": "wirepas/1111111/building12",
    "event_type": "invalid_value",
    "measurement": "monitoring_event"
  },
  {
    "time": "2021-06-01T11:20:00.000Z",
    "asset_type": "elevator_call_api",
    "asset_id": "elev_123",
    "event_source": "wirepas/1111111/building3",
    "event_type": "invalid_api_field",
    "measurement": "monitoring_event"
  }
]
```

Listing 7.1. Monitoring API JSON response

7.3.2. IBM Qradar

The implemented event API was tested with IBM Security QRadar SIEM[59]. QRadar is a SIEM system that can aggregate security-related information and events from several sources, analyse the data based on a set of predefined rules and anomaly detection, and combine this information to create an alert when offensive actions are detected. The alert contains information about all events related to the incident to help the SOC analyst to identify the origin of this offense.

QRadar can collect information by using a software component called Device Support Module (DSM). DSM parses the received data, converts it to a standardised event format and passes it on to Event Processor, which analyses the event by using Custom Rules Engine (CRE). If the event matches to some of the predefined rules, an action corresponding to that rule is executed. QRadar supports DSMs for various applications, network devices and cloud platforms, but due to the experimental nature of KEPPI, there were no existing integrations.

QRadar can receive information from several types of sources using different protocols. One of these is Universal Cloud Rest API protocol[60], which is an outbound protocol that can be used to actively fetch event information from remote REST APIs with HTTP requests. The protocol can be configured to a specific target API with a custom workflow XML document, which defines the required parameter values for the API and the process of retrieving and parsing events. The created workflow document that was used to add the KEPPI event API as a custom log source for SIEM can be found in Appendix 2. After the events are received by SIEM, the information gets parsed and the CRE can analyse the events to find incidents based on predefined rules.

7.3.3. Other implemented components

To test the event API, additional API mimicking the functionality of an elevator call interface was added in KEPPI. The elevator API resembled a service that is used by building users with a web or mobile application allowing them to call an elevator to their floor. The API was able to receive HTTP requests that contained unique identifier for the individual user, information on the used application, what elevator is called and what floor it should arrive to. The forged elevator call API was created using the HTTP API driver of the platform and it used token-based authentication similarly to platform's other HTTP APIs. If a request contains a valid authentication token, the information in it is validated to follow the expected format. User IDs, application and elevator identifiers, and possible floors for each elevator were defined in the API definition, and if there was any unexpected data fields present in the request, the elevator call was not forwarded to a pipeline acting as an elevator compatibility module. Instead, the validation result was used to create an event in the event database.

In addition to the elevator call API, the measurement database containing temperature sensor readings was duplicated in KEPPI and a service mimicking a sensor node was added. The fake node sent modified sensor data with the intention to disrupt the building system operations and the copied database was used to store these measurements so the testing would not affect the operations of the actual building.

8. EVALUATION

This chapter presents the evaluation of the proposed monitoring pipeline and event API, what limitations can be recognised from this approach and how they could be overcome in the future.

8.1. Event Collection Pipeline

As described in 7.3.1, there were six requirements for the monitoring pipeline. The compliance with the first, validating data produced by sensors and building systems, was tested by implementing the fake sensor service that irregularly produced unexpected measurement data. The fake sensor could depict e.g. a legitimate node that has been compromised and is being used to inject malformed data in an attempt to disrupt the building platform operations. The anomalous data was detected with the added sanity checks that verified that the data received from the sensor was in the correct and expected format and that the numerical measurement data was also between the predetermined threshold values.

Fulfilling the second requirement, validating data received from tenant applications, was tested by sending malicious requests to the elevator call API. It was assumed that the attacker had access to a valid API token that they had discovered from e.g. an insecure tenant application using the API. Similarly to sensor data, the received requests were tested with simple sanity checks to detect if any unexpected values were received. Although the monitoring pipeline was able to detect incorrect values in requests and prevent their submit to the elevator compatibility module, it would be useful for investigative activities to include the HTTP API driver logs to the created events.

The third requirement, identifying anomalous data, was fulfilled by creating an asset catalog in KEPPI that contained identifiers for all legitimate devices and resources, defining the accepted formats and values for each used attributes and adding the sanity checks to the test data provided by the pipelines. Although the entries with unexpected data, such as unknown device ID or measurements containing non-numeric characters, were detected, adding these checks afterwards to all data collection pipelines would not be possible without disrupting the data collection operations of the building. An alternative method was also tested where the received information was stored in the platform's database and queried periodically afterwards. Although it was possible to detect and remove the unexpected entries with this approach, the data could have already been used by some other service. In order to also prevent the use of malicious data, all data collection pipelines should contain the logic to verify the received information and submit any irregularities to the monitoring pipeline for further checks and event creation.

The rest of the requirements were creating events from anomalies, storing the events for a predefined time period and providing them to the external SIEM system. These were fulfilled by creating the event pipeline that received alerts from the test services and created events based on them, stored the events in a separate database instance running on the platform and published them using the event API. The InfluxDB driver running on KEPPI allowed to define how long the database entries are stored, so it is possible to change the period from hours to years depending on the needs and resources of the user organisation. The SIEM system stores the security logs as well, so it is possible to use a shorter retention time for the events in KEPPI. To satisfy the last requirement, the event API returns a list of detected events based on their time of occurrence. The API takes two timestamps defined in the HTTP request and uses them to query the events that have taken place during that time period. These are then returned as a response to the request. Similarly to the fake elevator API, data verification checks are performed for received requests before they are processed. If requests with unexpected values are received, the request is not processed further and new event entry is created.

In addition to complying with these requirements, the event detection was expected to honor user privacy and comply with the data regulation policies. The monitoring pipeline utilised only the information that would have been collected from the building environment anyway, so additional data sources that could be used to violate individual privacy were not used. The data processing was also minimised and the created events were stored on the platform and in SIEM for a minimal time. Although no data was collected explicitly for monitoring purposes, building users should be made aware that the building data is also processed by a security monitoring system since the monitoring system takes advantage of it.

8.2. Limitations

Although the requirements defined in the beginning of the monitoring pipeline implementation were mostly met, there are limitations to the capabilities of the implemented solution concerning the coverage of the ecosystem assets. This is due to the limited access to the back-end systems and infrastructure. Although the event pipeline was capable to detect modified and potentially malicious contents from the data received from building sensors and KEPPI services, the information might not be useful by itself. Without visibility to the back-end systems and the IT infrastructure running the services, it is not possible to determine the origin of an incident, investigate events further or in some cases to separate whether an event was created because of a purposeful misuse or a malfunctioning device. In addition, implementing any mechanisms that could prevent security incidents from occurring was not either sensible or possible to do on the platform. For example, implementing rate limiting to the service APIs could have been achievable, but the increased computational tasks on the server in case of a DDoS attack could disrupt the server operations on their own making the implementation a nonsensical option. Instead, the feature should be included in the HTTP API driver and effective measures should be utilised on the infrastructure level. This would however require extensive access to the environment, which again was not a possibility.

Concerning the validation of the data received from building systems and services and the event creation, the logic for identifying anomalies could be improved in the future. In the current state, the data was validated with mostly sanity checks and comparisons with the expected data format, but the KEPPI platform would also allow to run a machine learning container that could use anomaly detection algorithms for detecting outliers in time series data. Here however, the thresholds, expected data formats and asset listings were all defined separately to each measured attribute by examining the produced data and forming a picture of what type of data the system would produce in a normal state. Implementing this comprehensively to all data collection pipelines would require collecting sensor health telemetry from the WSN and integrating a resource and asset management components from all data-providing systems to KEPPI so that the information of legitimate devices was accessible and usable in the platform. In addition, the logic for detecting anomalies that are worthy of an event should be included in each data collection pipeline. Correlating the events with the relevant information from the IT infrastructure would then happen in the SIEM system. This is also when the true value of the implemented monitoring pipeline would come up as an addition to an already existing monitoring infrastructure that covers the typical IT assets of the ecosystem.

8.3. Reflective Observations

Collaborating with a major external project with several stakeholders representing different industry sectors can be helpful in identifying realistic challenges faced by different organisations. However, it can also introduce challenges for working on a relatively narrow topic within the scope of the whole project. These include for example dependency of other ongoing activities and the possibility to use project resources. If the topic of a subproject is highly dependent on results or outcome of some other ongoing task that faces delays, the subproject will likely be delayed as well. For working on subprojects such as theses, it is important to acknowledge these risks in early phases and determine other possible methods for achieving the desired outcome. It is also important to monitor the status and progress of these dependencies during the work and act immediately if any of the risks are turning into reality. During this work, this did not succeed. For future stages of the KEKO project and in any projects with a large consortium, it would be essential that the available resources are defined clearly before determining a final topic for subprojects and starting the work. Although it is and should be possible to adapt to changing circumstances, achieving the best possible research results will be challenging if the changing situation steers the project too far from the starting point.

8.4. Recommendations for the Future

When considering the cyber security of the experimental KEKO ecosystem, few key suggestions and recommended actions can be derived from the discussion in chapters 7 and 8. The first of them is defining the role of the KEPPI platform more clearly in the context of cyber security and monitoring of the ecosystem. The monitoring pipeline presented in this work shows that the platform can be used as a centralised event collector for smart building assets, but at its current state the event collection capabilities are limited. If KEPPI was to be used as the sole source for security-related events concerning the smart building assets, it should collect and store more operational information from the building systems, including device health telemetry and a catalog of legitimate assets and resources. Introducing this operational information to the platform could improve the detection of anomalous events and enable the utilisation of the platform's data processing pipelines in detecting these anomalies. Alternatively, if KEPPI is to be kept separate from any additional security monitoring systems, the system environment should include event collectors that are able to collect the operational information from sensor and automation equipment and provide it to the external monitoring system.

Secondly, it is important that the back-end systems and the IT network infrastructure are included in the monitoring scope. The back-end monitoring should include logs from both the server platform where KEPPI is running (currently in AWS) and the services provided by the KEPPI instance (e.g. status information of the data processing pipelines and the authentication logs of KEPPI APIs). Depending on the role of the KEPPI platform in the production version of the ecosystem, it could be used to provide the platform's own service logs to the monitoring system. The server platform events and logs should however be supplied to the security monitoring system via different route. For instance, AWS offers CloudWatch and CloudTrail that can be used to integrate the platform logs directly to an external SIEM system. For on-premise instances, it is possible to utilise OS logs and separate endpoint monitoring components provided by the SIEM system to collect the relevant events. In addition to these system logs, network sensors should be utilised to monitor the communications taking place in the ecosystem network infrastructure. Identifying malicious communications requires visibility to the networks.

The third general recommendation is that the security aspect should be taken into account in early stages of each step in the ecosystem development. This starts already from the beginning of designing the system and data flow architecture. The zero trust model should be applied ecosystem-wide by validating all users and interactions, verifying the integrity of communications and devices, and enforcing least-privilege access controls. Although this was on some level taken into account in individual components such as the WSN of the current version of the KEKO ecosystem, the data collection process performed by the experimental platform should be improved to follow the same principles as well. The integrity of the building data should be verified to prevent the use of malicious data in any building operations. The permission management should also be improved to follow the least-privilege principle. API tokens should be specific to certain APIs and the platform users should only be able to access and modify their own data processing pipelines. In addition, platform users should also have read-only access specifically to the data that their pipelines require.

The final recommendation is that the transparency of the ecosystem should be improved. Based on the observations during this work, this concerns both the technical components of the ecosystem and the level of technical documentation available. Regarding the technical components, the ecosystem should favour open protocols and technologies. Sensors and building systems that utilise proprietary technologies create unnecessary dependencies to specific vendors or equipment, and security through obscurity should never be considered a valid security measure. Mostly this is just a reminder due to the limited information on the automation systems utilised in the test site, which relates to the second mentioned aspect of the transparency. Although the documentation for utilising and creating services on the experimental platform was adequate, information about the underlying platform and existing assets had to be collected from other sources. This is also related to the first recommendation and the level of ecosystem integration in the context of cyber security. Although most of the assets were inferential from the available documentation and the collected building data, for example the make and model of the BMS and HVAC used in the test site were left unclear. This makes it impossible to assess how any features specific to their platforms would impact the security of the ecosystem. Although this is most likely caused by the experimental and changing nature of the current ecosystem, this should be acknowledged in the future.

9. CONCLUSIONS

The main goals for this work were to examine the system architecture of modern smart building ecosystems and investigate the different cyber security challenges that are introduced when both IT and OT assets are present in a building automation system environment. This knowledge was then used together with the MITRE ATT&CK knowledge base to identify different cyber security threats against an experimental version of an upcoming smart building ecosystem, KEKO. This work also presents recommendations on acknowledging identified security concerns and one potential solution for integrating the building data collection systems with a SIEM system to improve the awareness of the OT environment operations by implementing a monitoring pipeline and a security event API within the platform.

By itself, the presented implementation is able to detect only events that originate from the data processing pipelines of the environment or from the APIs created on the platform. This leaves the IT infrastructure of the ecosystem unmonitored, and detecting all of the techniques discussed in this work would require additional visibility to the IT side of the infrastructure. This needs to be taken into account for future stages of the project. When combined with an additional monitoring infrastructure that covers the typical IT assets of the ecosystem, the implemented monitoring pipeline could be used to gain additional information about security incidents and their effects to the cyber-physical systems. The proposed solution could also be applied to detect attacks that utilise novel side channels such as thermal or acoustic measurements, although at this time this may sound a little too far-fetched.

During this work, it became obvious that although there are a lot of differences between the IT and OT environments, the two are converging on each other. Whilst these systems are utilising more and more of the same technologies, there is still a wide gap between the IT and OT in the context of system security. To narrow down the gap, different industries must act together to address these issues. The KEKO Ecosystem project is a great example of these joint endeavours that makes it possible for industry operators to learn from each other. If there is one thing that all industry sectors should learn from these projects, it is that the security of a system should never be just an afterthought. There will always be someone who will try to abuse it.

10. REFERENCES

[1] Klepeis N.E., Nelson W.C., Ott W.R., Robinson J.P., Tsang A.M., Switzer P., Behar J.V., Hern S.C. & Engelmann W.H. (2001) The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. Journal of Exposure Science & Environmental Epidemiology 11, pp. 231–252.

[2] Hussein T., Paasonen P. & Kulmala M. (2012) Activity pattern of a selected group of school occupants and their family members in helsinki—finland. Science of the total environment 425, pp. 289–292.

[3] Bilal M., Oyedele L.O., Qadir J., Munir K., Ajayi S.O., Akinade O.O., Owolabi H.A., Alaka H.A. & Pasha M. (2016) Big data in the construction industry: A review of present status, opportunities, and future trends. Advanced engineering informatics 30, pp. 500–521.

[4] Traficom (2019), Suojaamattomia automaatiojärjestelmiä suomalaisissa verkoissa 2019. https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Suojaamattomia_automaatioj%C3%A4rjestelmi%C3%A4_suomalaisissa_verkoissa_2019.pdf.

[5] KEKO ecosystem project website. https://kekoecosystem.com/. Accessed: 2020-05-11.

[6] MITRE ATT&CK® website. https://attack.mitre.org/. Accessed 2020-11-19.

[7] GDPR - user-friendly guide to general data protection regulation. URL: https://www.gdpreu.org/.

[8] Shostack A. (2014) Threat modeling: Designing for security. John Wiley & Sons.

[9] Myagmar S., Lee A.J. & Yurcik W. (2005) Threat modeling as a basis for security requirements. In: Symposium on requirements engineering for information security (SREIS), vol. 2005, Citeseer, vol. 2005, pp. 1–8.

[10] OWASP, Application threat modeling. https://owasp.org/www-community/Application_Threat_Modeling. Accessed 2020-12-09.

[11] Shostack A. (2007, accessed October 29, 2020) STRIDE Chart. https://www.microsoft.com/security/blog/2007/09/11/stride-chart/.

[12] Strom B.E., Applebaum A., Miller D.P., Nickels K.C., Pennington A.G. & Thomas C.B. (2018) Mitre att&ck: Design and philosophy. Technical report .

[13] Strom B.E., Battaglia J.A., Kemmerer M.S., Kupersanin W., Miller D.P., Wampler C., Whitley S.M. & Wolf R.D. (2017) Finding cyber threats with att&ck-based analytics. The MITRE Corporation, Bedford, MA, Technical Report No. MTR170202 .

[14] MITRE ATT&CK® for enterprise matrix v8. https://attack.mitre.org/docs/attack_matrix_poster_2020_october.pdf. Accessed 2020-11-19.

[15] Tankard C. (2011) Advanced persistent threats and how to monitor and deter them. Network security 2011, pp. 16–19.

[16] Alexander O., Belisle M. & Steele J. (2020) MITRE ATT&CK® for Industrial Control Systems: Design and Philosophy. Technical report .

[17] MITRE ATT&CK® for industrial control systems website. https://collaborate.mitre.org/attackics/index.php/Main_Page. Accessed 2020-11-19.

[18] Kohnfelder L. & Garg P. (1999) The threats to our products. Microsoft internal report .

[19] Nweke L.O. (2017) Using the cia and aaa models to explain cybersecurity activities. PM World Journal 6.

[20] Buckman A.H., Mayfield M. & Beck S.B. (2014) What is a smart building? Smart and Sustainable Built Environment .

[21] Sinopoli J.M. (2009) Smart buildings systems for architects, owners and builders. Butterworth-Heinemann.

[22] Sinopoli J. (2016) Advanced technology for smart buildings. Artech House.

[23] Verma A., Prakash S., Srivastava V., Kumar A. & Mukhopadhyay S.C. (2019) Sensing, controlling, and iot infrastructure in smart building: a review. IEEE Sensors Journal 19, pp. 9036–9046.

[24] Stankovic J.A. (2008) Wireless sensor networks. computer 41, pp. 92–95.

[25] Krejčí R., Čeleda P. & Dobrovolnỳ J. (2012) Traffic measurement and analysis of building automation and control networks. In: IFIP International Conference on Autonomous Infrastructure, Management and Security, Springer, pp. 62–73.

[26] Yan P.F., Biuk-Aghai R.P., Fong S. & Si Y.W. (2007) Detection of suspicious patterns in secure physical environments. In: Fourth International Conference on Information Technology and Applications (ICITA 2007), Jan. 2007, v. 1, pp. 40–45.

[27] Wendzel S., Tonejc J., Kaur J., Kobekova A., Song H., Fink G. & Jeschke S. (2017) Cyber security of smart buildings. Wiley.

[28] Wang S. & Xie J. (2002) Integrating building management system and facilities management on the internet. Automation in construction 11, pp. 707–715.

[29] Insteon finally comes clean about its sudden smart home shutdown. https://arstechnica.com/gadgets/2022/04/insteon-finally-comes-clean-about-its-sudden-smart-home-shutdown/. Accessed 2022-05-17.

[30] Insteon - home assistant. https://www.home-assistant.io/integrations/insteon/. Accessed 2022-05-18.

[31] Khaund K. (2015) Cybersecurity in smart buildings: inaction is not option any more. A Frost & Sullivan Collaborative Industry Perspective .

[32] Mansfield-Devine S. (2019) The state of operational technology security. Network security 2019, pp. 9–13.

[33] McBride S.M., Schou C.D. & Slay J. (2020) A security workforce to bridge the it-ot gap. ? .

[34] Grispos G. (2019) Criminals: Cybercriminals. Encyclopedia of Security and Emergency Management , pp. 1–7.

[35] Meyers C., Powers S. & Faissol D. (2009) Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. Tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

[36] Kruglov K. (2019), Threat landscape for smart buildings. https://securelist.com/smart-buildings-threats/93322/. Accessed: 2020-05-18.

[37] European Union Agency for Cybersecurity (2017), Baseline security recommendations for IoT. https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/at_download/fullReport. Accessed: 2020-05-18.

[38] Hackers use new tactic at austrian hotel: Locking the doors. https://www.nytimes.com/2017/01/30/world/europe/hotel-austria-bitcoin-ransom.html. Accessed: 2020-05-18.

[39] Researchers hack building control system at google australia office. https://www.wired.com/2013/05/googles-control-system-hacked/. Accessed: 2021-04-16.

[40] Linder L., Vionnet D., Bacher J.P. & Hennebert J. (2017) Big building data-a big data platform for smart buildings. Energy Procedia 122, pp. 589–594.

[41] Schiaffino S. & Amandi A. (2009) Intelligent user profiling. In: Artificial Intelligence An International Perspective, Springer, pp. 193–216.

[42] Barbato A., Borsani L., Capone A. & Melzi S. (2009) Home energy saving through a user profiling system based on wireless sensors. In: Proceedings of the first ACM workshop on embedded sensing systems for energy-efficiency in buildings, pp. 49–54.

[43] Agarwal Y., Balaji B., Gupta R., Lyles J., Wei M. & Weng T. (2010) Occupancy-driven energy management for smart building automation. In: Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building, pp. 1–6.

[44] Puteh S., Langensiepen C. & Lotfi A. (2012) Fuzzy ambient intelligence for intelligent office environments. In: 2012 IEEE International Conference on Fuzzy Systems, IEEE, pp. 1–6.

[45] Wachter S. (2019) Data protection in the age of big data. Nature Electronics 2, pp. 6–7.

[46] Gambs S., Killijian M.O. & del Prado Cortez M.N. (2014) De-anonymization attack on geolocated data. Journal of Computer and System Sciences 80, pp. 1597–1614.

[47] Narayanan A. & Shmatikov V. (2006) How to break anonymity of the netflix prize dataset. arXiv preprint cs/0610105 .

[48] Barbaro M. & Zeller T. J. (2006) A face is exposed for aol searcher no. 4417749. The New York Times URL: https://www.nytimes.com/2006/08/09/technology/09aol.html, accessed: 2020-08-02.

[49] Holm C. (2018) Smart buildings: Law and ethics. Scandinavian Studies in Law 65, pp. 257–268.

[50] Cejka S., Knorr F. & Kintzler F. (2019) Privacy issues in smart buildings by examples in smart metering. In: The 25th international conference and exhibition on electricity distribution (CIRED 2019), AIM, pp. 1–5.

[51] Chen J., Edwards L., Urquhart L. & McAuley D. (2020) Who is responsible for data processing in smart homes? reconsidering joint controllership and the household exemption. International Data Privacy Law 10, pp. 279–293.

[52] WhereOS. https://www.whereos.com/. Accessed: 2021-06-01.

[53] Apache Spark. https://spark.apache.org/. Accessed: 2021-05-15.

[54] Apache Hive. https://hive.apache.org/. Accessed: 2021-05-15.

[55] Singh V. & Peddoju S.K. (2017) Container-based microservice architecture for cloud applications. In: 2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE, pp. 847–852.

[56] Secure Standardized Logging - AWS CloudTrail - Amazon Web Services. https://aws.amazon.com/cloudtrail/. Accessed: 2022-06-16.

[57] Bhatt S., Manadhata P.K. & Zomlot L. (2014) The operational role of security information and event management systems. IEEE security & Privacy 12, pp. 35–41.

[58] Radoglou-Grammatikis P., Sarigiannidis P., Iturbe E., Rios E., Martinez S., Sarigiannidis A., Eftathopoulos G., Spyridis I., Sesis A., Vakakis N. et al. (2021) Spear siem: A security information and event management system for the smart grid. Computer Networks , p. 108008.

[59] IBM Security QRadar SIEM. https://www.ibm.com/qradar/security-qradar-siem. Accessed: 2022-05-23.

[60] Universal Cloud REST API protocol. https://www.ibm.com/docs/en/dsm?topic=configuration-universal-cloud-rest-api-protocol. Accessed: 2022-05-23.

11. APPENDICES

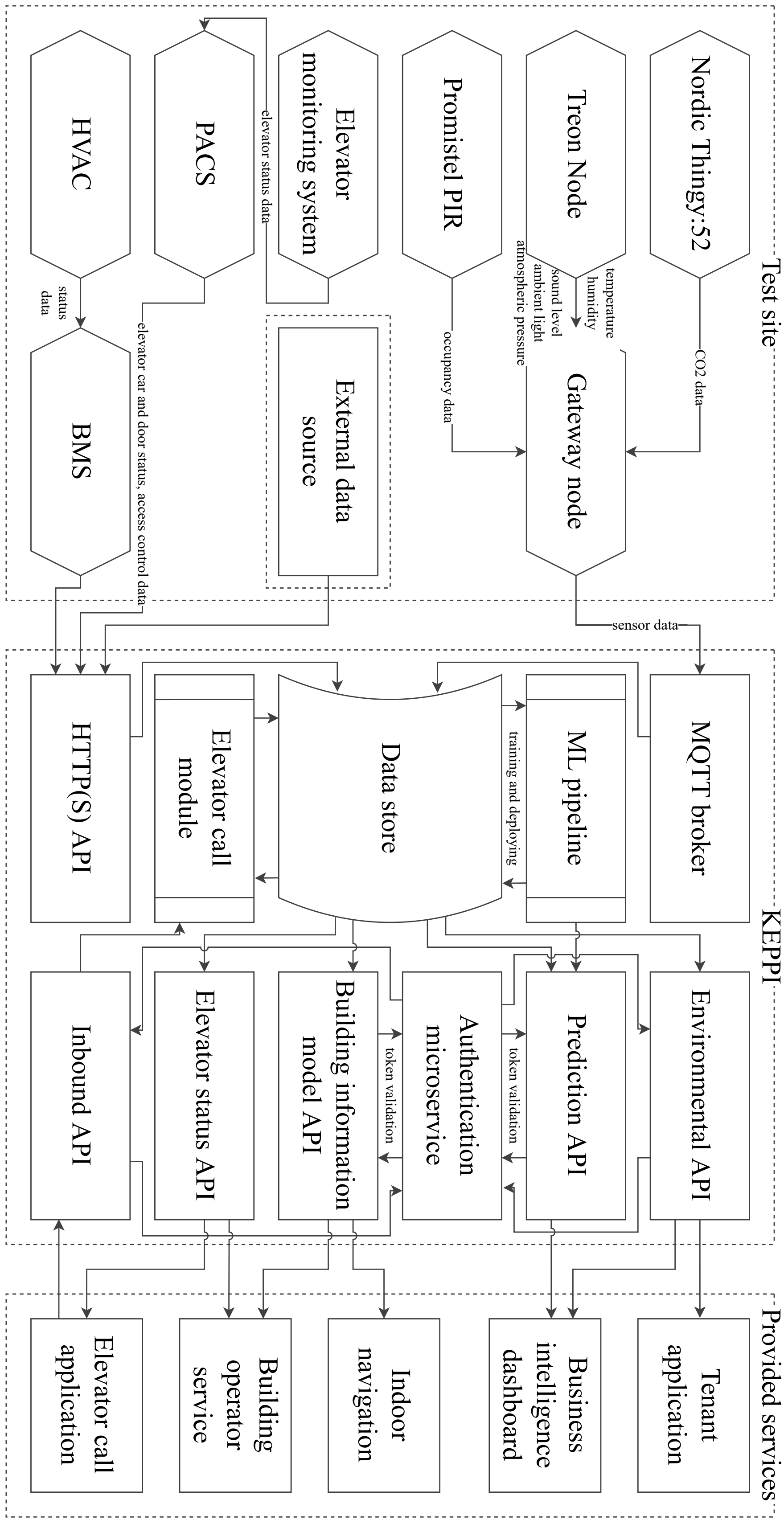
Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 34 techniques	Credential Access 14 techniques	Discovery 24 techniques	Lateral Movement 9 techniques	Collection 16 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Valid Accounts ≡	Scheduled Task/Job ≡			Modify Authentication Process ≡		System Service Discovery	Remote Services ≡	Data from Local System	Data Obfuscation ≡	Exfiltration Over Other Network Medium ≡	Data Destruction
Replication Through Removable Media	Windows Management Instrumentation	Valid Accounts ≡			Network Sniffing		Software Deployment Tools	Data from Removable Media	Fallback Channels	Scheduled Transfer	Data Encrypted for Impact
Trusted Relationship	Software Deployment Tools	Hijack Execution Flow ≡			OS Credential Dumping ≡	Application Window Discovery	Replication Through Removable Media	Input Capture ≡	Application Layer Protocol ≡	Data Transfer Size Limits	Service Stop
Supply Chain Compromise ≡	Shared Modules	Boot or Logon Initialization Scripts ≡		Direct Volume Access	Input Capture ≡	System Network Configuration Discovery	Internal Spearphishing	Data Staged ≡	Proxy ≡	Exfiltration Over C2 Channel	Inhibit System Recovery
Hardware Additions	User Execution ≡	Create or Modify System Process ≡		Rootkit	Brute Force ≡	System Owner/User Discovery	Use Alternate Authentication Material ≡	Screen Capture	Communication Through Removable Media	Exfiltration Over Physical Medium ≡	Defacement ≡
Exploit Public-Facing Application	Exploitation for Client Execution	Event Triggered Execution ≡		Obfuscated Files or Information ≡	Two-Factor Authentication Interception	System Network Connections Discovery	Lateral Tool Transfer	Email Collection ≡	Web Service ≡	Exfiltration Over Web Service ≡	Firmware Corruption
Phishing ≡	System Services ≡	Boot or Logon Autostart Execution ≡			Exploitation for Credential Access	Permission Groups Discovery ≡	Taint Shared Content	Clipboard Data	Multi-Stage Channels	Automated Exfiltration	Resource Hijacking
External Remote Services	Command and Scripting Interpreter ≡	Account Manipulation ≡	Process Injection ≡		Steal Web Session Cookie	File and Directory Discovery	Exploitation of Remote Services	Automated Collection	Ingress Tool Transfer	Exfiltration Over Alternative Protocol ≡	Network Denial of Service ≡
Drive-by Compromise	Native API	External Remote Services	Access Token Manipulation ≡		Unsecured Credentials ≡	Peripheral Device Discovery	Remote Service Session Hijacking ≡	Audio Capture	Data Encoding ≡	Transfer Data to Cloud Account	Endpoint Denial of Service ≡
	Inter-Process Communication ≡	Office Application Startup ≡	Group Policy Modification		Credentials from Password Stores ≡	Network Share Discovery		Video Capture	Traffic Signaling ≡		System Shutdown/Reboot
		Create Account ≡	Abuse Elevation Control Mechanism ≡		Steal or Forge Kerberos Tickets ≡	Password Policy Discovery		Man in the Browser	Remote Access Software		Account Access Removal
		Browser Extensions	Exploitation for Privilege Escalation	Indicator Removal on Host ≡	Forced Authentication	Browser Bookmark Discovery		Data from Information Repositories ≡	Dynamic Resolution ≡		Disk Wipe ≡
		Traffic Signaling ≡		Modify Registry	Steal Application Access Token	Virtualization/Sandbox Evasion ≡		Man-in-the-Middle ≡	Non-Standard Port		Data Manipulation ≡
		BITS Jobs		Trusted Developer Utilities Proxy Execution ≡	Man-in-the-Middle ≡	Cloud Service Dashboard		Archive Collected Data ≡	Protocol Tunneling		
		Server Software Component ≡		Traffic Signaling ≡		Software Discovery ≡		Data from Network Shared Drive	Encrypted Channel ≡		
		Pre-OS Boot ≡		Signed Script Proxy Execution ≡		Query Registry		Data from Cloud Storage Object	Non-Application Layer Protocol		
		Compromise Client Software Binary		Rogue Domain Controller		Remote System Discovery					
		Implant Container Image		Indirect Command Execution		Network Service Scanning					
				BITS Jobs		Process Discovery					
				XSL Script Processing		System Information Discovery					
				Template Injection		Account Discovery ≡					
				File and Directory Permissions Modification ≡		System Time Discovery					
				Virtualization/Sandbox Evasion ≡		Domain Trust Discovery					
				Unused/Unsupported Cloud Regions		Cloud Service Discovery					
				Use Alternate Authentication Material ≡							
				Impair Defenses ≡							
				Hide Artifacts ≡							
				Masquerading ≡							
				Deobfuscate/Decode Files or Information							
				Signed Binary Proxy Execution ≡							
				Exploitation for Defense Evasion							
				Execution Guardrails ≡							
				Modify Cloud Compute Infrastructure ≡							
				Pre-OS Boot ≡							
				Subvert Trust Controls ≡							

≡ Has sub-techniques

MITRE ATT&CK®
Enterprise Framework

attack.mitre.org

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Workflow name="KEPPI-testapi" version="1.0" xmlns="http://qradar.ibm.com/UniversalCloudRESTAPI/Workflow/V1">

    <Parameters>
        <Parameter name="host" label="Host" required="true" />
        <Parameter name="access_token" label="Access Token" required="true" secret="true" />
    </Parameters>

    <Actions>

        <!-- Initialize bookmark to one hour before current time if it is empty -->
        <Initialize path="/bookmark" value="${time() - (1 * 60 * 60 * 1000)}" />

        <!-- Format bookmark time to yyyy-MM-ddTHH:mm:ss.SSSZ which is used by KEPPI -->
        <FormatDate pattern="yyyy-MM-dd'T'HH:mm:ss.SSS'Z'" timeZone="UTC" time="${/bookmark}" savePath="/start_time_formatted" />

        <CallEndpoint url="https://${/host}/api/monitoring/testapi" method="GET" savePath="/get_events">
            <BearerAuthentication token="${/access_token}" />

            <!-- Get events after this -->
            <QueryParameter name="start_time" value="${/start_time_formatted}" omitIfEmpty="true"/>

            <!-- Get event before this -->
            <QueryParameter name="end_time" value="" omitIfEmpty="true"/>

            <RequestHeader name="Accept" value="application/json"/>
            <RequestHeader name="Content-Type" value="application/json"/>
        </CallEndpoint>

        <!-- Handle Errors -->
        <If condition="/get_events/status_code != 200">
            <Abort reason="${/get_events/status_code}: ${/get_events/status_message}"/>
        </If>

        <!-- Post Events, if any -->
        <If condition="count(/get_events/body) > 0">

            <!-- Add event type to each event and post it -->
            <ForEach item="/current_event" items="/get_events/body">

                <!-- Post the Event -->
                <PostEvent path="/current_event" source="${/host}" />

            </ForEach>

            <!-- Set bookmark to current time -->
            <Set path="/bookmark" value="${time()}"/>

        </If>

    </Actions>

    <Tests>
        <DNSResolutionTest host="${/host}"/>
        <TCPConnectionTest host="${/host}"/>
        <SSLHandshakeTest host="${/host}"/>
    </Tests>

</Workflow>
```


Test site
Nordic Thingy:52
Treon Node
Promistel PIR
Elevator monitoring system
PACS
HVAC
CO2 data
temperature
humidity
sound level
ambient light
atmospheric pressure
occupancy data
Gateway node
External data source
elevator status data
elevator car and door status, access control data
status data
BMS
sensor data
KEPPI
MQTT broker
ML pipeline
training and deploying
Data store
Elevator call module
HTTP(S) API
Environmental API
Prediction API
token validation
Authentication microservice
token validation
Building information model API
Elevator status API
Inbound API
Provided services
Tenant application
Business intelligence dashboard
Indoor navigation
Building operator service
Elevator call application