ТЕХНИЧЕСКИ УНИВЕРСИТЕТ -СОФИЯ

ФАКУЛТЕТ ПО ТЕЛЕКОМУНИКАЦИИ

КАТЕДРА “КОМУНИКАЦИОННИ МРЕЖИ”

маг. инж. Мариан Христов

ИЗСЛЕДВАНЕ НА ОБЛАЧНО-БАЗИРАНА СИСТЕМА ЗА НАБЛЮДЕНИЕ И АНАЛИЗ НА СЪБИТИЯ

А В Т О Р Е Ф Е Р А Т

на дисертация за придобиване на образователна и научна степен
"ДОКТОР"

Област: 5. Технически науки

Професионално направление: 5.3 „Комуникационна и компютърна техника”

Докторантска програма: „Осигурителна техника и системи”

Научни ръководители: доц. д-р инж. Мария Вълчева Ненова

доц. д-р инж. Елица Емилова Гиева

София, 2026г

Дисертационният труд е обсъден и насочен за защита от Катедрения съвет на катедра „Комуникационни мрежи“ към Факултета по телекомуникации на ТУ-София на редовно заседание, проведено на 06.04.2026 г.

Публичната защита на дисертационния труд ще се състои на 24.07.2026 г. (петък) от 13:00 ч. в конферентната зала на БИЦ на Технически университет – София на открито заседание на Научното жури, определено със заповед № ОЖ-5.3-35 от 23.04. г. на Ректора на ТУ-София в състав:

1. доц. д-р инж. Кирил Маринов Късев – председател
2. доц. д-р инж. Георги Руменов Цочев - научен секретар
3. проф. д-р инж. Станимир Михайлов Садинов
4. проф. д-р инж. Габриела Лъчезарова Атанасова
5. доц. д-р инж. Ивелина Стефанова Балабанова

Резервни членове:

1. проф. дн инж. Ивайло Иванов Атанасов
2. проф. д-р инж. Валентина Илиева Маркова

Рецензенти:

1. доц. д-р инж. Кирил Маринов Късев
2. проф. д-р инж. Габриела Лъчезарова Атанасова

Материалите по защитата са на разположение на интересуващите се в канцеларията на Факултет по телекомуникации на ТУ-София, блок № 1, кабинет № 1306.

Дисертантът е задочен докторант към катедра „Комуникационни мрежи“ на Факултет по телекомуникации. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. инж. Мариан Христов

Заглавие: Изследване на облачно-базирана система за наблюдение и анализ на събития

Тираж: 30 броя

Отпечатано в ИПК на Технически университет – София

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Съвременната цифрова трансформация, ускорена от Industry 4.0, води до все по-тясна интеграция между информационните технологии (IT) и оперативните технологии (OT), както и до масово въвеждане на IoT/IIoT устройства и отдалечена свързаност в Индустриалните мрежи. Това повишава ефективността, мащабируемостта и автоматизацията на процесите, но едновременно с това разширява възможните точки за атака и увеличава вероятността от киберинциденти с оперативен и физически ефект. За разлика от класическите IT среди, компрометиране на OT/ICS инфраструктура може да доведе не само до изтичане на данни или финансови щети, а и до прекъсване на технологични процеси, рискове за безопасността и потенциално застрашаване на човешкия живот.

В тази среди наблюдението, събирането и анализът на събития се превръщат в ключов фактор за киберустойчивост. На практика обаче индустриалните организации често се сблъскват с няколко системни предизвикателства. Първо, голяма част от OT устройствата не позволяват инсталиране на агенти, а активното сканиране може да наруши детерминизма на процеса, което налага пасивни, мрежово-базирани подходи за видимост. Второ, събитията и телеметрията идват от хетерогенни източници (OT сензори, мрежови устройства, защитни системи, логове от платформи и приложения), което усложнява корелацията и превръщането на детекциите в ясни, оперативни действия. Трето, дори когато съществуват механизми за детекция, често липсва надеждна и достатъчно бърза връзка между детекция и реакция - тоест своевременен канал за уведомяване и структуриран процес на реакция и неутрализиране за екипите по сигурността.

Допълнително предизвикателство е надеждността на самото наблюдение: при внедряване на решения за пасивен, без агент мониторинг възниква критичният въпрос “кой наблюдава системата за наблюдение?”. Загубата на видимост поради проблем в сензор, интеграция или инфраструктура може да остави средата незащитена точно в момент на настъпваща атака. Наред с това, съвременните извършители на злонамерени действия все по-често злоупотребяват с легитимни функционалности и процеси, за да прикриват злонамерени действия, - например firmware downgrade, маскиран като нормален update процес - което създава „симетрия“ между легитимни и нелегитимни операции и повишава сложността на детекцията, ако липсва контекстно-ориентирана аналитична логика.

Поради тези причини изследването и разработването на облачно-базирана система за наблюдение и анализ на събития е актуален научно-приложен проблем. Облачният подход предоставя възможности за централизация, мащабируемост, интеграция на множество източници на телеметрия, аналитика в реално време и автоматизация на реакцията. В същото време той изисква специфични решения за OT контекст: пасивност, висока надеждност на мониторинга, контекстно обогатяване на алармите и механизми за разграничаване на легитимни от злонамерени активности. Следователно, разработването и валидирането на архитектура и механизми за облачно-базиран мониторинг и анализ на събития, приложими към Индустриалните мрежи, представлява значим и актуален принос към повишаване на киберустойчивостта на критични среди.

Цел на дисертационния труд, основни задачи и методи за изследване

Настоящия дисертационен труд има за основна цел **разработването и изграждането на цялостна система за наблюдение и анализ на събитията в индустриални мрежи (OT/ICS), като се стреми да подобри нивото на сигурност на организациите чрез повишаване на мрежовата видимост и своевременно откриване на заплахи**. За постигане на тази цел е необходимо да се изпълнят следните задачи:

1. Идентифициране на заплахи и основни рискове в OT/ICS;
2. Анализ на съществуващите решения за справяне с тях;
3. Проучване на технологии за мониторинг и анализ на събитията в OT/ICS;
4. Разработване на система за наблюдение на OT/ICS базирана на резултатите от направеното задълбочено проучване;
5. Реализация на предложеното техническо решение в тестова и реална индустриална среда;
6. Оценка на ефективността на разработената система чрез практически експерименти, тестови подходи и реална работна функционалност;
7. Разработване на методология за оценка на риска в OT среди, базирана на разработеното техническо решение и установени практики и стандарти и областта.

Методологична основа

Методологията на изследването в дисертационния труд включва използване на аналитични и експериментално-приложни подходи. Аналитичният подход е приложен при прегледа на състоянието на проблема, формулиране на изисквания към облачно-базирана система за наблюдение и анализ на събития и определяне на критерии за оценка на ефективността ѝ в OT/ICS контекст.

Експериментално-приложният подход е използван при проектирането, реализирането и валидирането на предложените технически механизми чрез контролирани тестови сценарии и практическо изпълнение в тестова и/или реална Индустриална - производствена - среда. На база експерименти е оценена работоспособността на интеграцията между пасивния OT мониторинг (D4IoT), SIEM частта (Microsoft Sentinel) и автоматизациите (Azure Logic apps), както и ефективността на реализираните механизми за уведомления в реално време, health/operational мониторинг и version-aware детекция на firmware downgrade.

Научна новост

Научната новост в настоящия дисертационен труд е свързана с изследването и разработването на облачно-базирана система за наблюдение и анализ на събития в индустриални (OT/ICS) мрежи, съобразена със специфичните ограничения и изисквания на OT средата (пасивност, липса на агенти, високи изисквания за непрекъсваемост и безопасност). В дисертационния труд е извършен анализ на съвременните заплахи и системни слабости в OT/ICS, който служи като основа за дефиниране на архитектурни принципи и функционални изисквания към системата за мониторинг и анализ на събития.
Като научна новост се разглежда и разработването на многокомпонентен подход за повишаване надеждността на мониторинга чрез независими механизми за откриване на деградация или загуба на видимост. Допълнително, представена е аналитична логика, съобразена с версиите, - version-aware - която позволява разграничаване на легитимни фърмуерни промени от потенциално злонамерени firmware downgrade действия, въпреки високата прилика между процесите. Предложените подходи демонстрират практическа ефективност чрез валидиране в тестова и/или реална индустриална - производствена - среда и водят до по-бързо откриване и реакция при инциденти и до намаляване на нерелевантни

аларми, като по този начин се повишава устойчивостта на наблюдението и управляемостта на риска в OT/ICS инфраструктури.

Практическа приложимост

Всички разработени механизми и компоненти на предложената облачно-базирана система за наблюдение и анализ на събития са реализирани и валидирани чрез практически експерименти в тестова и/или реална индустриална - производствена - среда. Извършено е оценяване на ефективността на интеграцията между пасивния, без агент, OT мониторинг (D4IoT), SIEM частта (Microsoft Sentinel) и автоматизациите (Azure Logic apps), както и на допълнителни механизми за оперативно наблюдение и диагностика.

Предложените решения са приложими в реални Индустриални мрежи, където са налице ограничения за активни сканирания и инсталиране на агенти, и където се изисква надеждно уведомяване и бърза реакция при инциденти. Реализираните механизми могат да бъдат използвани както за оперативно известяване и начално разследване, така и за повишаване устойчивостта на мониторинга чрез откриване на деградация или загуба на видимост и разграничаване на високорискови събития (например фърмуерно понижаване) от рутинни промени.

Публикации

Основните постижения и резултати от дисертационния труд са представени в общо 6 авторски публикации, една от които е публикувана в международно научно списание в Q2. Пет от научните публикации са представени и публикувани в сборници от международни научни конференции и национални конференции с международно участие. Пет от публикациите са в съавторство с научните ръководители. Налични са общо 49 цитирания, всички от които са в индексирани издания.

Международните научни конференции са: 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), 2023 14th National Conference with International Participation (ELECTRONICA), 2023 XXXII International Scientific Conference Electronics (ET), 2023 31st National Conference with International Participation (TELECOM), 2025 33rd National Conference with International Participation (TELECOM).

Международното научно списание е: MDPI Symmetry, 2025.

Структура и обем на дисертационния труд

Дисертационният труд е написан на български език и има обем от 194 страници A4 и съдържа въведение, четири глави, заключение с основните приноси, насоки за бъдеща работа, списък на авторите публикации, списък на използваните съкращения, списък на фигури, списък на таблици, списък на използваната литература. Дисертационният труд съдържа 155 фигури и 5 таблици. Използвани са 240 литературни източника, всички на английски език, 95% от които са от последните десет години. Номерата на фигурите и таблиците в автореферата отговарят на тези в дисертацията.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

ГЛАВА 1. ТЕОРЕТИЧНА ОСНОВА И АРХИТЕКТУРНИ ПРИНЦИПИ НА ОБЛАЧНО-БАЗИРАНИ СИСТЕМИ ЗА НАБЛЮДЕНИЕ И АНАЛИЗ НА СЪБИТИЯ ПРИЛОЖИМИ В ИНДУСТРИАЛНИТЕ МРЕЖИ

1.2 Основни предизвикателства

Индустриалните среди се отличават с редица особености, които правят киберсигурността и мониторинга съществено по-различни от IT контекста. Основните предизвикателства са:

необходимост от непрекъсваемост на процеса и безопасност, наличие на устройства с дълъг жизнен цикъл, ограничени времеви прозорци за пачване и рестарт, както и риск от нарушаване на детерминизма при активни сканирания. Често е невъзможно да се инсталират агенти върху типичните устройства - PLC/RTU/HMI - или това е недопустимо поради производствени ограничения, което налага пасивни, мрежово-базирани методи за видимост. Допълнително, индустриалните инфраструктури страдат от системни слабости като ограничена видимост, недостатъчна сегментация, неконтролиран отдалечен достъп и споделени IT–OT креденшъни. Като резултат, дори при наличие на технологии за мониторинг, често се наблюдава разминаване между детекцията и реалните оперативни действия, което забавя реакцията и увеличава риска от ескалация.

1.4 Наблюдение и анализ на събития в Индустриални мрежи

Наблюдението и анализът на събития в OT/ICS мрежи е ключов механизъм за ранна детекция на аномалии и реагиране при инциденти, но е затруднен от хетерогенността на източниците на данни и ограниченията за активни методи. Ефективният подход изисква централизирано събиране на телеметрия, - пасивен анализ на трафика, аларми от OT мониторинг сензори, системни логове, IPS/IDS събития - нормализация и корелация в SIEM платформа, както и контекстно обогатяване - например локация, критичност на системата, CMDB справка, TI/OSINT проверка. Съществен проблем е надеждността на самата видимост - при отказ на сензор, загуба на облачна свързаност или проблем в интеграцията могат да възникнат “тихи откази”. Следователно, освен детекция, е необходим и оперативен мониторинг на мониторинг системата, който да сигнализира при деградация на телеметрията и да предотврати неустановена загуба на наблюдението.

1.5 Имплементация на механизъм за уведомления при откриване на зловреден софтуер в индустриални мрежи

1.5.3 Реализация на техническото решение

Предложеното инженерно техническо решение за уведомяване при откриване на зловреден софтуер в Индустриални мрежи е реализирано чрез три взаимосвързани компонента: (1) аналитично правило в Microsoft Sentinel за детекция и генериране на инцидент, (2) Azure Logic apps плейбук за автоматизирано изпращане на имейл уведомления и (3) Sentinel Automation rule, който свързва инцидента с плейбука и маркира обработените случаи. Решението използва като вход алармите от Industrial malware аналитичния модул на Microsoft Defender for IoT (D4IoT), които се внедряват в Sentinel посредством „Microsoft Defender for IoT“ data connector. Основната идея е техническата детекция да бъде трансформирана в кратко, контекстно и оперативно приложимо уведомление към SOC/CTAC, което да ускори реакцията, да подпомогне първоначалното разследване и да намали риска от ескалация в OT/ICS среда.

1.5.3.1 Аналитично правило, откриващо зловреден софтуер

Първият компонент е аналитично правило в Sentinel, което търси и корелира D4IoT аларми, свързани със зловредна активност. Правилото се създава от меню „Analytics“, като първоначално се задават име и описание в секция „General“. След това се дефинира Kusto Query Language (KQL) заявка, която използва таблицата SecurityAlert и ограничава резултатите до D4IoT аларми чрез условието ProviderName == "IoTSecurity". За да се селектира конкретно Industrial malware аналитичния модул на D4IoT, в заявката се изисква ProductComponentName == "MALWARE". По този начин правилото фокусира анализа върху аларми, отразяващи потенциална компрометация, като изключва останалите D4IoT модули (POLICY_VIOLATION, ANOMALY, OPERATIONAL, PROTOCOL_VIOLATION), които имат различен характер и цел.

След филтрацията се използва KQL операторът extend за инициализация на променливи, които извличат ключови атрибути от алармата – логически адреси на участващите системи, идентификатор/име на сензора, както и допълнителни свойства, необходими за последващо уведомяване. Извличането се реализира чрез tostring() и parse_json(), тъй като част от данните са вложени в JSON структура. Следва важна стъпка за контекстно обогатяване: чрез функция case() се въвежда мапинг между име на сензор и физическа локация. Така например на променлива Location се присвоява стойност „Sofia“ при сензори, чието име съдържа „sof“. Подобна логика позволява алармите да бъдат превърнати от чисто техническо събитие в сигнал с оперативен контекст, което е особено важно в разпределени индустриални среди с множество локации. Финално, чрез оператора project се избира подмножество от атрибути (например време, тип аларма, сорс система, сензор, локация), които са достатъчни за инцидент и за последващо имейл уведомление.

След дефиниране на KQL логиката се преминава към секция „Incident settings“, където се активира създаването на инциденти при съвпадение. Това е съществена настройка, тъй като именно инцидентът се използва като тригер за автоматизацията. След валидиране и „Save“ в „Review + create“ правилото е налично в Analytics и започва да генерира инциденти при откриване на malware аларми от D4IoT. По този начин първият компонент осигурява надежден механизъм за превръщане на D4IoT аларми в SIEM инциденти с подходящ приоритет, които могат да бъдат обработвани от SOC процесите.

1.5.3.2 Механизъм за уведомление при откриване на зловреден софтуер

Вторият компонент е Azure Logic apps плейбук, който изпраща автоматични имейл уведомления при създаден инцидент за зловредна активност. Ролята на този механизъм е да съкрати времето за известяване и да осигури бързо включване на по-опитни специалисти (Incident Responders) при наличие на подозрителни събития в индустриалната мрежа. Практическата логика е SOC анализаторът на смяна да получава структурирано уведомление с ключовата информация, без да се налага първо ръчно да търси и разкодира детайлите в SIEM конзолата.

Създаването на Logic apps започва с избор на хостинг план (Standard или Consumption) и задаване на Subscription, Resource group, име и регион. След успешното създаване структурата се изгражда през „Logic app designer“ като последователност от оператори. Тригерът на плейбука е операция „Microsoft Sentinel incident“, която стартира изпълнение при наличието на инцидент, породен от аналитичното правило „D4IoT – Malware Identified in Industrial Network“. Това осигурява директна интеграция между SIEM инцидента и автоматизацията.

След стартиране, плейбукът извършва разбор (parsing) на данните, които се съдържат в инцидента. Важно архитектурно решение е, че аналитичното правило групира множество malware аларми по сорс система (потенциално компрометираното устройство), така че при наличие на повече от една детекция за една и съща машина да се генерира един инцидент, вместо множество. Поради това Logic apps използва цикличен оператор „For each alert“, който преминава през алармите в инцидента. Данните в секцията „Custom Details“ се обработват чрез „Parse JSON“, като се използва предварително дефинирана схема за коректно извличане на полетата. Това позволява динамичните стойности (тип аларма, сорс адрес, сензор, локация и др.) да бъдат използвани в следващите стъпки.

Финалната част на Logic apps е оператор „Send an email“, който изпраща уведомление с висока важност. Тялото на имейла комбинира статичен текст и динамични променливи, така че получателят да получи минималния, но достатъчен контекст за започване на разследване: тип детекция (например „Malware Test File Detected“ или „Connection Attempt to Known Malicious IP“), засегната система, физическа локация и други релевантни

атрибути. По този начин уведомлението е кратко, ориентирано към реакция и подходящо за оперативна употреба.

1.5.3.3 Правило за автоматизиране в Sentinel

Третият компонент е Sentinel Automation rule, който представлява „свързващ елемент“ между генерираните инциденти и Logic apps плейбука. Правилото се създава в секция „Automation“ и се конфигурира да се задейства при „When incident is created“. За да се гарантира, че плейбукът ще се изпълнява само за релевантни инциденти, се задават условия: „Analytic rule name“ се ограничава до „D4IoT – Malware Identified in Industrial Network“. Това предотвратява ненужни уведомления при други типове инциденти и запазва фокуса върху Industrial malware детекции.

След задаване на условията се конфигурират действията на автоматизацията. Основното действие е „Run playbook“, което стартира Logic apps механизма за имейл уведомяване. След изпращането на имейла автоматизацията добавя таг към инцидента – „Email Sent“. Това има две практични ползи: (1) позволява бързо разграничаване на инциденти, по които вече е извършено уведомяване, и (2) подпомага отчетността и проследимостта на обработката.

В обобщение, реализацията на техническото решение изгражда цялостна верига „детекция-инцидент-автоматизация-уведомяване“, при която D4IoT осигурява пасивната OT-aware детекция, Sentinel предоставя SIEM корелация и инцидентен модел, а Logic apps реализира автоматизирана комуникация към оперативните екипи. Така се редуцира времето за уведомяване, осигурява се контекст за бързо triage начало и се създава практическа основа за по-ефективна реакция при зловредни активности в индустриални мрежи.

1.5.4 Тестване на работоспособността на техническото решение

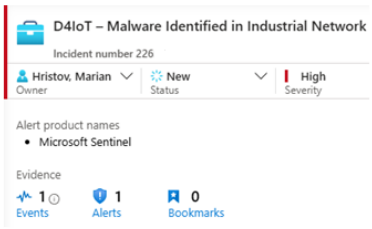
За да се провери и докаже работоспособността на предложеното техническо решение, са проведени тестове в два етапа, включително използване на EICAR тест файл. В първия етап, в непроизводствена среда на промишлено предприятие (инсталация за предварително въвеждане в експлоатация), EICAR файлът е изтеглен на работна станция, чийто мрежови трафик е наблюдаван чрез D4IoT. Действието е засечено и D4IoT генерира аларма „Malware Test File Detected – EICAR AV Success“. Алармата е изпратена към Azure D4IoT, постъпва в Microsoft Sentinel и задейства разработения механизъм за уведомяване. В Sentinel се визуализират ключови атрибути като време, име на алармата, участващи устройства, сензор и локация.

След внедряване на алармата посредством data connector и аналитично правило, в рамките на приблизително две минути се генерира инцидент „D4IoT – Malware Identified in Industrial Network“ с висок приоритет. Това автоматично активира последователно automation rule и Azure Logic apps, в резултат на което се изпраща имейл уведомление до предварително зададените получатели. Имейлът съдържа минимален, но оперативно достатъчен контекст: тип аларма, засегната система (192.168.10.1) и физическа локация (Sofia). Допълнително, инцидентът се тагва автоматично с „Email Sent“, което подпомага разграничаването на вече обработените случаи и улеснява работата на анализатора на смяна. Експерименталните резултати са представена на Фиг 1.30, 1.31 и 1.32.

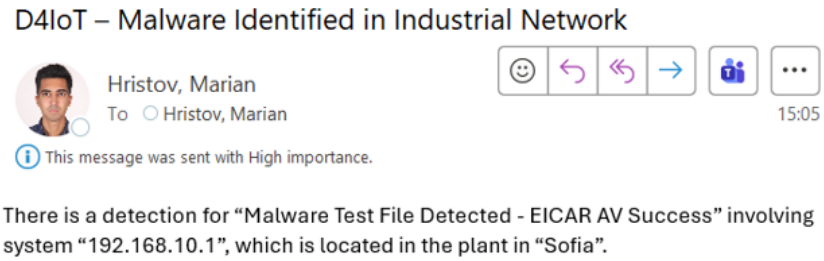
Results Chart

TimeGenerated [UTC]	AlertName	Source	Destination	Sensor	Location
1/9/2026, 12:54:22.827 PM	Malware Test File Detected - EICAR AV Success	192.168.10.1	192.168.10.2	mhh-sof-sens001	Sofia

Фиг 1.30 “Malware Test File Detected - EICAR AV Success” аларма в Sentinel

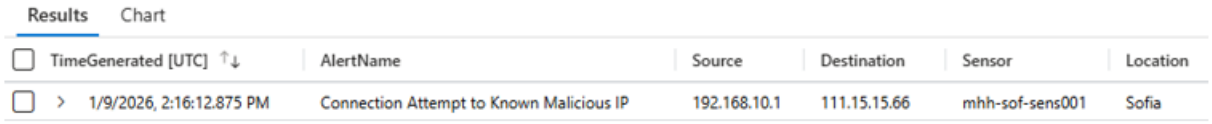


Фиг 1.31 "D4IoT – Malware Identified in Industrial Network" инцидент

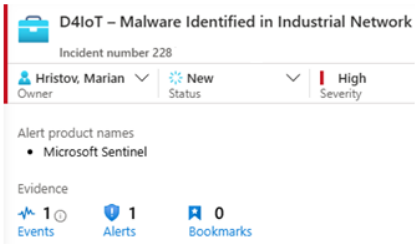


Фиг 1.32 "D4IoT – Malware Identified in Industrial Network" имейл уведомление

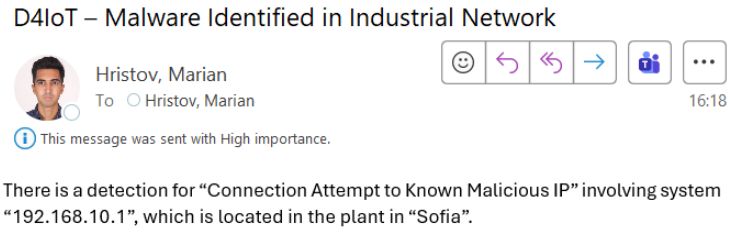
След доказване на работоспособността на решението, обхватът е разширен за покриване на цялата индустриална мрежа, като всички D4IoT сензори изпращат аларми към Azure D4IoT и Sentinel, а тези от Industrial malware аналитичния модул задействат механизма за уведомяване. Вторият експеримент цели да валидира ефективността при реалистичен индикатор за компрометиране – опит за комуникация със злонамерена система. Работната станция (192.168.10.1) прави опит за достъп до IP адрес 111.15.15.66. Комуникацията е блокирана от защитна стена, но изпратените пакети водят до генериране на аларма „Connection Attempt to Known Malicious IP“ от D4IoT сензора. Почти мигновено аналитичното правило в Sentinel генерира втори инцидент с висок приоритет, а автоматизацията изпраща имейл с детайли (тип аларма, източник и локация) и поставя таг „Email Sent“. Експерименталните резултати са представена на Фиг 1.34, 1.35 и 1.36.



Фиг 1.34 "Connection Attempt to Known Malicious IP" аларма



Фиг 1.35 Втори "D4IoT – Malware Identified in Industrial Network" инцидент



Фиг 1.36 Имейл уведомление относно засичане на “Connection Attempt to Known Malicious IP”

Полученото уведомление служи като начална точка за структурирано първоначално разследване (triage). Като първа стъпка засегнатата вътрешна система може да бъде проверена в CMDB с цел определяне на предназначение и критичност; при некритичност е препоръчително изолиране/карантина с цел ограничаване на разпространение или ексфилтрация, без да се създава допълнителен риск за индустриалния процес. Следващата стъпка е анализ на дестинацията чрез Threat Intelligence и OSINT проверки. Публични източници (например GreyNoise, Cisco Talos и AbuseIPDB) категоризират 111.15.15.66 като злонамерен адрес; според GreyNoise е наблюдавана активност като сканиране и опити за атаки по SSH (порт 22), което потвърждава валидността на детекцията и подпомага приоритизацията на реакцията. В обобщение, тестовете показват, че предложеното решение осигурява надеждна верига „детекция-инцидент-уведомяване“, предоставя минимален, но достатъчен контекст за SOC и подпомага бързо и аргументирано вземане на решения за реакция в индустриална среда.

1.5.5 Заключение

Представената проблематика в Глава 1 от дисертационния труд обосновава, че съвременните заплахи за OT/ICS организации не се ограничават до кражба на данни (интелектуална собственос) или финансови измами, а могат да доведат до реални оперативни прекъсвания и физически последствия застрашаващи човешкия живот и безопасност. Примери от близкото минало като Stuxnet, Industroyer и особено TRITON демонстрират еволюция на кибератаките насочени към критични процеси и системи за безопасност, като подчертават необходимостта от бърза детекция и координирана реакция цел максимално смекчаване на последствията за индустриалната среда. В този контекст, ключово предизвикателство за много компании в отрасъла не е липсата на механизми за детекция, а неналичието на надежден и достатъчно бърз способ за уведомяване, който да свързва техническия аспект на детекцията с реалните действия предприети от екипа за реагиране (SOC).

Предложеното техническо решение адресира именно този дефицит чрез изграждане на механизъм за нотификация в реално време относно засичането на зловреден софтуер в индустриалната мрежа, първоначално открит посредством D4IoT и надграден с интеграция на Microsoft Sentinel и Azure Logic apps автоматизация. Комбинацията между пасивното, без агент, мрежово наблюдение на D4IoT и SIEM функциите на Sentinel позволява превръщането на аларми в структурирани, оперативно полезни уведомления, които достигат до центъра за информационна сигурност почти мигновено. Съществена добавена стойност е обогатяването на контекста чрез KQL модификации, - извличане на SensorId и дефиниране на Location - което впоследствие прави имейл уведомлението едновременно кратко и с фокус върху реакцията, съдържайки: тип детекция, логически адрес на афектираното устройство/система и физическа локация.

Тестването - първоначално в непроизводствена, а в последствие и в реална производствена среда - демонстрира практическата ефективност на подхода, а именно при засичане на подозрителни действия - например опит за изграждане на изходяща връзка с известен злонамерен IP - автоматизацията генерира своевременен имейл до SOC, съкращавайки времето за реакция. Това е критично в OT/ICS контекст, където всяко потенциално забавяне

увеличава вероятността от разпространение на заразата, изнасяне на информация (Data Exfiltration) и/или ескалация към действия, които биха засегнали непрекъсваемостта на индустриалните процеси.

Важен аспект на авторското предложение е и описаният базов метод за начално разследване (Triage), което превръща уведомлението в начална точка на структуриран задълбочен анализ – първоначална CMDB проверка целяща определяне на критичността и предназначението на засегнатото устройство/система, последван от проверка на репутацията на дестинацията чрез OSINT. Тази установена последователност подпомага вземането на решение за изолиране/поставяне под карантина на устройството(а), когато това е допустимо - всяко действие трябва да се изпълни така, че да не създава допълнителен риск за индустриалната операция - и за приоритизация на последващи действия, без да се разчита на непълна контекстуална информация. Посредством това, автора успешно адресира следните предварително зададените цели и задачи към дисертационния труд, а именно:

1. Реализация на предложеното техническо решение в тестова и реална индустриална среда;
2. Оценка на ефективността на разработената система чрез практически експерименти, тестови подходи и реална работна функционалност;
3. Разработване на методология за оценка на риска в ОТ среди, базирана на разработеното техническо решение и установени практики и стандарти и областта.

В заключение, представеното техническо решение демонстрира, че установяването на надеждна взаимовръзка между детекция и реакция е ключов фактор за повишаване на киберустойчивостта на OT/ICS мрежи. Посредством прилагане на съществуващи функционалностти на D4IoT и Sentinel и създаване на целенасочени обогатявания и автоматизация, подходът осигурява бърза, контекстна и приложима нотификация за SOC, което от своя страна намалява времето за откриване и реакция и подпомага ограничаването на последствията от инциденти със зловреден софтуер ако не и пълното им елиминиране.

1.5.6 Приноси към глава 1

Основните приноси от направените проучвания и анализи в глава 1 могат да бъдат обобщени по следния начин:

- Извеждане на “real-time notification pipeline” метод за OT/ICS детекции, - D4IoT:SIEM:автоматизация:SOC - като се дефинира минимален набор от атрибути за оперативно валидно уведомление - тип аларма, физическа локация, система-източник;
- Формулиране на първичен модел за анализ на OT/ICS аларми, включващ комбинация от CMDB контекст - предназначение и критичност на системата - и Open Source Intelligence (OSINT) репутация на дестинации като минимален, възпроизводим набор от стъпки за своевренна първоначална оценка на потенциалния риск със сигурността;
- Създаване и успешно внедряване на автоматизация – базирана на Azure Logic apps - за незабавно уведомяване при наличие на malware детекции от D4IoT в индустриални среди с Sentinel интеграция. Механизма разполага с имейл известие, изпращано към SOC;
- Обогатяване на алармите и контекста на събитието на база на KQL подобриния, включващо извличане и представяне на SensorId (име на D4IoT сензор) и “case mapping” между SensorId и потенциалната физическа локация на потенциалнно компроментираната система-източник. Това превръща суровата аларма в контекстен и реакционно-ориентиран сигнал;

- Демонстриране на работещ тестови сценарий в производствена среда, включващ откриване на подозрителни действия и автоматично изпращане на имейл уведомление с ключови параметри, позволяващи незабавно стартиране на разследване и потенциално отреагиране. Техническия механизъм е допълнен от практически "playbook" за реакция, включващ проверка в CMDB за вътрешната система и OSINT проверка за външната, с препоръка за изолиране при допустимост спрямо производствения процес.

ГЛАВА 2. ИНТЕГРАЦИЯ И ЦЕНТРАЛИЗИРАНО СЪБИРАНЕ НА OT/IOT СЪБИТИЯ В ОБЛАЧНА АНАЛИТИЧНА ПЛАТФОРМА

2.1 Логове - същност и предназначение

Логовете представляват структурирани или неструктурирани записи за събития и действия в системи, приложения и мрежови устройства, които служат като основен източник на доказателствена информация за наблюдение, диагностика и разследване на инциденти. В OT/ICS контекст логовете имат допълнителна стойност, тъй като подпомагат проследимостта на технологични отклонения, промени по активи и аномалии в комуникациите, като едновременно с това трябва да се събират и използват без да се нарушава непрекъсваемостта на процеса. Поради хетерогенността на източниците и различните формати е необходимо логовете да бъдат нормализирани и обогатени с контекст, за да бъдат оперативно полезни за екипите по сигурност.

2.2 Системи за мониторинг и съхранение на събития

Системите за мониторинг и съхранение на събития (например SIEM) осигуряват централизирано събиране, съхранение, търсене, корелация и визуализация на логове и аларми от множество източници, създавайки "single pane of glass" за наблюдение и реакция. В индустриални мрежи ефективността на подобни системи зависи от възможността да интегрират OT-aware телеметрия (пасивен трафик анализ, аларми от OT сензори), да поддържат правила в реално време и автоматизации, и да предоставят контекст за приоритизация (локация, критичност на системи и/или репутация на дестинации). По този начин SIEM се превръща в ключов инструмент за намаляване на времето за откриване и реакция и за осигуряване на доказуемост при разследване.

2.3 Имплементация на оперативен мониторинг за Microsoft Defender for IoT

Оперативният мониторинг (health/operational monitoring) на имплементация на D4IoT е критичен компонент за киберустойчивостта на индустриалните среди. Причината е, че в OT/ICS контекст загубата на видимост не е просто технически проблем, а директен риск: липсата на телеметрия може да остави средата без защита точно при настъпваща атака, което потенциално води до прекъсване на технологични процеси, финансови щети и дори застрашаване на човешката безопасност. Настоящата секция представя разработеното инженерно решение за оперативен мониторинг на D4IoT, което адресира въпроса „кой и как наблюдава мониторинг системата?“ чрез многокомпонентен подход с независими механизми и различни канали за уведомяване.

2.3.2 Подготовка за внедряване на техническото решение

Разработеното инженерно решение изисква три предпоставки: (1) облачно свързан D4IoT сензор, (2) интеграция с Microsoft Sentinel и (3) наличен локален Zabbix сървър за SNMP мониторинг. При напълно air-gapped среди, където няма облачна свързаност, решението може да се редуцира до локалния компонент (Zabbix/SNMP), който остава приложим и независим от Azure.

За постигане на пълната функционалност е необходимо D4IoT сензорът да бъде конфигуриран като Cloud connected по време на onboarding процеса. Това осигурява

телеметрия към Azure портала, като допълнително е необходимо да бъдат разрешени изходящи връзки към определени Azure secure endpoints през порт 443 (TLS). На практика това се реализира чрез правила в защитната стена, позволяващи двупосочен трафик според изискванията на Microsoft документацията.

Следващата стъпка е интеграцията със Sentinel чрез активиране на "Microsoft Defender for IoT" data connector, който внедрява D4IoT алармите в таблицата SecurityAlert. Това е ключово условие за първия компонент на оперативния мониторинг (SIEM правила), тъй като те разчитат на наличието на D4IoT аларми в Sentinel работното пространство.

Последната подготвителна операция е интеграцията със Zabbix сървър чрез конфигуриране на SNMP MIB monitoring. Изисква се задаване на IP адрес на Zabbix и създаване на SNMPv3 потребител с парола и секретен ключ за автентикация. Тази конфигурация позволява Zabbix да следи управленския интерфейс на D4IoT сензора и да сигнализира при недостъпност.

2.3.3 Реализация на техническото решение

Решението е проектирано като три независими компонента, които работят паралелно и наблюдават различни аспекти от D4IoT имплементацията. Този многовалентен подход е избран целенасочено, за да се избегне зависимост от един индикатор и да се минимизира рискът от еднократна точка на отказ. В допълнение, компонентите използват различни канали за уведомяване (инцидент в Sentinel, Teams съобщение и имейл), което повишава надеждността и гарантира видимост за проблема дори при частична деградация на инфраструктурата.

Компонент 1: Sentinel аналитични правила за липса на D4IoT аларми

Първият компонент включва две аналитични правила в Sentinel, които следят за липса на D4IoT аларми в таблицата SecurityAlert. Идеята е проста, но практическа: ако D4IoT внезапно спре да подава аларми, това може да е симптом за проблем в интеграцията D4IoT–Azure D4IoT –Sentinel или за загуба на телеметрия. В OT/ICS контекст "липсата на събития" може да е по-опасна от наличието на аларми, защото скрива реални атаки.

Правило 1: "D4IoT – Operational (No alerts in the last 30 minutes)"

Целта му е да установи тотална липса на D4IoT аларми в последните 30 минути. Реализацията използва KQL заявка, която:

1. филтрира таблицата SecurityAlert по ProviderName == "IoTSecurity";
2. извлича времето на последната D4IoT аларма чрез агрегираща функция max() върху TimeGenerated;
3. сравнява резултата с ago(30m) и при условие, че последната аларма е по-стара, генерира инцидент.

Това правило е полезно като "бърз индикатор" за проблем в ingestion/връзката и е приложимо в среди с нормално наличие на D4IoT аларми.

Правило 2: "D4IoT – Operational (No alerts for a sensor in the last 12 hours)"

Второто правило проверява липса на аларми от конкретен сензор за последните 12 часа. Използва се:

1. таблица (datatable) със списък на всички очаквани D4IoT сензори (D4IoTSensors);
2. отделен набор (distinct) от сензори, които са изпратили поне една аларма за последните 12 часа (D4IoTAlerts);
3. операция rightanti join, за да се намерят сензори, присъстващи в D4IoTSensors, но липсващи в D4IoTAlerts.

Практически това правило открива сензори, които не са изпратили аларми в дадения период, което може да индикира проблем с конкретната производствена локация, SPAN конфигурация, ingestion или самия сензор. Важно е да се отбележи, че в определени среди и при високо ниво на tuning е възможно в рамките на 12 часа да няма нито една аларма,

което може да доведе до false positives. Поради това правилото е полезно като компонент на многовалентната система, но изисква внимателно интерпретиране в контекст.

Компонент 2: Azure Logic Apps + Azure Resource Graph мониторинг на облачната свързаност

Вторият компонент адресира различен тип проблем: сензорът е локално активен, но е загубил облачна свързаност към Azure. Този сценарий е критичен, защото при cloud-connected модел D4IoT алармите трябва да стигнат до Azure D4IoT/Sentinel, за да бъдат обработени от SOC. Ако свързаността отпадне, мониторингът може да продължи локално, но централизираната видимост се губи.

Компонентът е реализиран чрез Logic apps, който се изпълнява периодично (например на 10 минути) и изпраща HTTP POST заявка към Azure Resource Graph (ARG). ARG връща свойства за D4IoT сензорите, включително connectivityTime (AzureLastConnectivityTime). На база на тази стойност се маркира сензорът като "Disconnected", ако последната свързаност е по-стара от определен праг.

Решението включва и anti-flood механизъм: първо се изпраща Teams уведомление при преминаване на праг (например 30–40 мин), а след допълнително изчакване (Wait 15 minutes) се изпълнява втора проверка за по-тежък праг (например 45+ мин). Ако несвързаността е налице, но времето е под 56 минути, се създава Sentinel инцидент с висок приоритет чрез "Create Incident" операция. Тази двустепенна логика цели:

1. ранно предупреждение (Teams) за потенциален проблем;
2. ескалация (Sentinel incident) при продължителна несвързаност;
3. избягване на заливане с нотификации при дълготрайни повреди.

Изборът на Teams като първи канал е практичен – той е бърз, но не е SLA-базиран. Затова при продължителен проблем се преминава към SIEM инцидент, който е по-подходящ за проследяване и управляемост.

Компонент 3: Локален SNMP/Zabbix мониторинг на management интерфейса

Третият компонент е независим от Azure и Sentinel и е особено важен за air-gapped или Интернет-ограничени среди. Целта му е да открие проблеми на ниво операционна система или management интерфейс на D4IoT сензора – например когато сензорът спре да отговаря на ping към management IP. Практическите наблюдения показват, че това състояние често е свързано с проблем в OS/виртуален контейнер, което може да доведе до частична или пълна загуба на функционалност.

Реализацията е чрез Zabbix, конфигуриран да изпраща ICMP Echo Request към management интерфейса на всеки 2 минути. При три последователни пропуснати 3 отговора (общо 6 минути) Zabbix изпраща имейл уведомление до OT security екипа. Този компонент има две силни страни:

1. най-кратко време за детекция на "твърд" отказ (6 мин);
2. приложимост независимо от облачна свързаност и SIEM ingestion.

В обобщение, имплементацията на оперативния мониторинг за D4IoT реализира многовалентна система, която покрива: (1) ingestion/интеграционни проблеми (липса на аларми), (2) cloud connectivity проблеми (ARG/Logic apps), и (3) локални откази на management/OS (Zabbix). Комбинацията от независими индикатори и канали за уведомяване минимизира риска от неустановена загуба на видимост и създава практическа основа за своевременна реакция и troubleshooting в OT/ICS среда..

2.4 Тестване на работоспособността на техническото решение

За да се изпробва и докаже работоспособността на предложеното техническо решение за оперативен мониторинг на D4IoT, са проведени серия от тестове, насочени към наблюдение и отчитане на начина на работа на всеки от трите компонента в цялостната система. Симулациите са планирани така, че да тестват един от най-критичните практически сценарии – „повреден D4IoT сензор", тъй като това състояние може да доведе до загуба на телеметрия и "слепи петна" в индустриалната видимост. Времевите интервали на тестовете

са съобразени с логиката и праговете на отделните механизми за детекция (Zabbix/SNMP, Azure Logic apps/ARG и Sentinel аналитични правила), така че да се валидира както ранното известяване, така и последващата ескалация при продължителна неизправност.
За целта тестов D4IoT сензор е изключен контролирано чрез команда “system shutdown”, изпълнена в Command-line interface (CLI) от “admin” акаунт. Така се гарантира, че състоянието “off” е резултат от реална недостъпност на сензора, а не от междинни мрежови флуктуации. Проведените експерименти, очакваните резултати и отчетените резултати могат да бъдат обобщени в следната логика:
(1) Zabbix/SNMP имейл известяване (най-бърз компонент)

Първото очаквано действие е задействане на локалния SNMP мониторинг, реализиран чрез Zabbix, който проверява достъпността на management интерфейса на D4IoT. Тъй като Zabbix е конфигуриран да изпраща ICMP ping заявки на всеки две минути и да алармира при три последователно пропуснати отговора, очакваният резултат е имейл уведомление след приблизително 6 минути от окончателното спиране на сензора. Отчетеният резултат потвърждава това поведение – имейлът е получен шест минути след shutdown. Практическата стойност на този компонент е, че осигурява най-ранна сигнализация при “твърд” отказ на сензора, независимо от облачната свързаност и SIEM интеграцията.

При реална повреда, засечена по този начин, отстраняването често изисква физически достъп до сензора или отдалечен достъп до хардуерния интерфейс за управление (например iLO/iDRAC). Това показва, че Zabbix компонентът е подходящ за ранно откриване на проблеми на ниво операционна система/хардуер и служи като първичен индикатор, че мониторинг инфраструктурата е деградирала.

(2) Teams съобщение при деградация на облачната свързаност (ранно предупреждение)

Следващият очакван етап е активиране на Logic apps механизма, който следи последната облачна свързаност на D4IoT сензора към Azure чрез ARG. В предложената логика първото действие е Teams уведомление при достигане на праг (30 минути) от последната успешна комуникация. Отчетеният резултат потвърждава очакването – 30 минути след последната връзка между D4IoT и Azure е получено Teams съобщение.

Това известяване има предупредителен характер: на този етап е възможно проблемът да е временен (например кратка загуба на Интернет/маршрут, рестарт на защитна стена или мрежово устройство). По тази причина Logic apps включва допълнителна стъпка “wait” (изчакване 15 минути), преди да премине към ескалация. Teams каналът е избран като ранно уведомление, защото е бърз и оперативен, но не е SLA-ориентиран механизъм, което логично води до следващата стъпка – инцидент в Sentinel при по-продължителна несвързаност.

(3) Sentinel инцидент за цялостна липса на D4IoT аларми (интеграционен индикатор)

Паралелно с Teams предупреждението е отчетен и инцидент в Sentinel, генериран от аналитичното правило “D4IoT – Operational (No alerts in the last 30 minutes)”, което следи за тотална липса на D4IoT аларми в таблицата SecurityAlert за последните 30 минути. В тестовия сценарий, където има само един сензор, липсата на телеметрия логично води до задействане на инцидента. В реална производствена среда с множество сензори това правило би било по-информативно за глобални проблеми – например срив в Azure D4IoT, проблем с data connector-а или обща деградация в Sentinel.

Този резултат потвърждава, че първият компонент на решението (Sentinel правила) работи коректно като индикатор за прекъсване на телеметрията на ниво SIEM.

(4) Sentinel инцидент при 45-минутна несвързаност към Azure (ескалация)

След изчакването от 15 минути Logic apps извършва втора проверка чрез ARG и при достигане на по-висок праг (45 минути от последната комуникация) генерира Sentinel

инцидент. Отчетеният резултат съответства на очакваното поведение – 15 минути след Teams съобщението е получен Sentinel инцидент, указващ, че сензорът не е комуникирал с Azure вече 45 минути.

Тази стъпка се интерпретира като необходимост от реално отреагиране, тъй като вероятността проблемът да е временен намалява, а рискът от загуба на централизирана видимост става значим. Добра отправна точка за troubleshooting в такъв случай е проверка на Интернет/маршрутната свързаност на сензора и изпълнение на диагностични CLI команди (например за connectivity checks), които да потвърдят дали проблемът е в локалната мрежа, защитната стена или в конфигурацията на сензора.

(5) Sentinel инцидент за липса на аларми от конкретен сензор за 12 часа (дългосрочен индикатор)

Последният валидиран елемент е аналитичното правило "D4IoT – Operational (No alerts for a sensor in the last 12 hours)", което генерира инцидент при липса на аларми от конкретен сензор в продължение на 12 часа. Отчетеният резултат потвърждава очакваното – поне 12 часа след окончателното спиране на сензора се появява съответният Sentinel инцидент.

Важно е да се подчертае, че в реални среди е възможно правилото да генерира false positives, ако за определен период реално няма аларми (например малка среда с високо ниво на tuning). Поради това правилото има стойност като компонент на многовалентната система, но резултатите му следва да се интерпретират в контекст и да се потвърждават с допълнителни проверки.

Практически насоки за верификация и начално troubleshooting

Като отправна точка за установяване дали телеметрията е възстановена, подходяща практика е генериране на тестова D4IoT аларма (например чрез вградените проверки за системно състояние в интерфейса на сензора). Успешното генериране и последващо появяване на алармата в Azure/Sentinel е индикация, че веригата D4IoT-Azure D4IoT-Sentinel отново функционира коректно.

Обобщение на резултатите от тестовете

Проведените експерименти потвърждават, че предложеното техническо решение функционира според дизайна и осигурява поетапно, надеждно и многоканално известяване при сценарий „неналичен D4IoT сензор". Най-ранният индикатор е Zabbix (6 мин), следван от Teams предупреждение (30 мин) и последваща ескалация чрез Sentinel инциденти (30 мин за липса на аларми, 45 мин за несвързаност към Azure, 12 часа за липса на аларми от конкретен сензор). Тази последователност демонстрира основната стойност на многовалентния подход: дори при частична деградация на инфраструктурата системата запазва способност да уведомява чрез независими канали и индикатори, като едновременно подпомага диагностицирането на вероятната причина (OS/management отказ, загуба на cloud свързаност, интеграционен проблем). Това валидира практическата приложимост на оперативния мониторинг като необходим слой за гарантиране на видимост и устойчивост на OT/ICS мониторинг внедрявания. Експерименталните резултати са представена на Фиг 2.24, 2.25, 2.26, 2.27 и 2.28.

-----Original Message-----
From:
Sent:
To:
Subject: Problem: Node is unavailable by ping

Problem started at 14:56:36 on 2023.10.26 Problem name: Node is unavailable by ping
Host: mhh-sof-sens001
Severity: High
Operational data: Down (0)

Фиг 2.24 Имейл от Zabbix сървъра уведомяващ за неполучаване на 3 последователни пинг отговора

Фиг 2.25 Съобщение в Teams уведомяващо, че сензорът D4IoT е несвързан към Azure

Title	Created time	Owner
D4IoT - Operational (No alerts in the last 30 minutes)	10/26/23, 3:10 PM	Hristov, Marian

Фиг 2.26 Sentinel инцидент задействан поради липса на D4IoT аларми през последните 30 минути

Title	Created time	Owner	Description
mhh-sof-sens001 is Disconnected	10/26/23, 3:25 PM	Hristov, Marian	The last connectivity of mhh-sof-sens001 to Azure was 10/26/23, 2:40 PM

Фиг 2.27 Sentinel инцидент задействан 45 след последната комуникация между D4IoT и Azure

Title	Created time	Owner
D4IoT - Operational (No alerts for a sensor in the last 12 hours)	10/27/23, 12:00 PM	Hristov, Marian

Фиг 2.28 Sentinel инцидент задействан поради липса на D4IoT аларми от конкретен сензор през последните 12 часа

2.5 Заключение

Представената информация в Глава 2 от дисертационния труд обосновава, че с нарастващата дигитализация и размиване на границите между ИТ и ОТ средите, внедряването на способи за видимост и детекция в индустриалните мрежи вече е необходимост, но само по себе си не е достатъчно. Когато OT/ICS организация разчита на техническо решение за пасивно, без агент, мрежово наблюдение като D4IoT, възниква критичният въпрос “кой и как наблюдава мониторинг системата?”. Ограничената или изцяло загубена видимост поради проблем в сензор, интеграция или инфраструктура може

да остави средата незащитена точно в момент на настъпваща кибератака, което в OT/ICS контекст означава риск от оперативни прекъсвания, финансови щети и потенциално застрашаване на човешката безопасност.

В този контекст, предложеното техническо решение за оперативен мониторинг (health/operational monitoring) адресира ключовия риск от трудно уловими откази на D4IoT имплементация, като разполага с разработени независими един от друг механизми за откриване на влошаване на видимостта и за ранно известяване. Практическата стойност на предложения подход е, че той не се основополага на един-единствен индикатор, а използва три компонента, които работят паралелно и независимо, като наблюдават различни аспекти от D4IoT имплементацията и използва различни способи за уведомяване. Първият компонент – съставен от две Sentinel правила – следи за липса на D4IoT аларми на глобално ниво и относно конкретен сензор. По този начин своевременно се установяват потенциални проблеми D4IoT-Azure D4IoT-Sentinel интеграцията и се минимизира вероятността за неустановена загуба на телеметрия. Вторият компонент – реализиран посредством Azure Logic apps и ARG заявка - добавя способ за мониториране на облачната свързаност на D4IoT сензор(ите), разполагайки с механизъм за предотвратяване на потенциално заливане от известия в случай на дълготрайни повреди с D4IoT инфраструктурата. Третият и последен компонент - SNMP мониторинг, базиран на Zabbix сървър – осигурява способ за откриване на потенциални проблеми на ниво операционна система и/или интерфейс за управление в индустриални среди, които са изцяло откъснати от Интернет – тоест не разполагат с облачна свързаност.

Тестването на предложеното техническо решение се базира на контролирано изключване на D4IoT сензор и демонстрира ефективността на многовалентния инженерен подход - първоначално се задейства най-бързият компонент - Zabbix - след едва шест минути от “повредата” на D4IoT сензора, което води до автоматично имейл уведомление. В последствие - при преминаване на прага за облачна свързаност – Azure Logic app изпраща Teams съобщение, индикиращо несвързаността на сензора. Финално, два Sentinel инцидента биват генерирани поради липсата на D4IoT аларми в съответните времеви интервали. Тази последователност показва, че реализирания способ за оперативен мониторинг не само открива потенциални проблеми своевременно, но и предоставя насоки за допълнителни проверки относно изправността на D4IoT имплементацията. По този начин, автора успешно адресира две от предварително зададените цели и задачи към дисертационния труд, а именно: Реализация на предложеното техническо решение в тестова и реална индустриална среда / Оценка на ефективността на разработената система чрез практически експерименти, тестови подходи и реална работна функционалност.

В обобщение, предложеното техническо решение представя практичен и реализуем подход за повишаване на устойчивостта на OT/ICS мониторинга, комбинирайки SIEM правила, способ за наблюдение на облачната свързаност и независим изцяло локален мониторинг. Това води до основният извод - в индустриална среда не е достатъчно да се имплементира способ за детекция, а е необходимо е да се изгради и надежден оперативен мониторинг, който постоянно да валидира наличието на видимост и да уведомява при отпадане на ключови компоненти.

2.6 Приноси към глава 2

Приносите към глава 2 могат да бъдат обобщени, както следва:

- Формализиране на концепцията “наблюдение на системата за наблюдаване” в OT/ICS контекст на база дефиниране на индикатори на оперативен мониторинг, - липса на аларми, прекъсване на облачната свързаност или не наличност на интерфейса за управление - служещи за измерима симптоматика на потенциалната деградация на системата;
- Прилагане на многослоен модел целящ надеждност на оперативния мониторинг, който се базира на независими уведомителни механизми, - SIEM инцидент, Teams съобщение и имейл – намаляващи вариантността от “single point of failure”;
- Създаване и успешно внедряване на две Sentinel аналитични правила, които откриват и уведомяват при “липса на D4IoT аларми - глобално (30 минути)” и “липса на D4IoT аларми - конкретен сензор (12 часа)”, използвайки KQL оператори за време и корелация;
- Създаване и успешно внедряване на Azure Logic apps с ARG заявка за наблюдение на “LastConnectivityTime” и автоматично Teams уведомяване при прекъсната облачна свързаност. Механизма разполага с “anti-flood” логика;
- Създаване и успешно внедряване на независим изцяло локален мониторинг слой - Zabbix сървър, SNMP и Ping - за наблюдаване на интерфейса за управление на D4IoT сензор(а/ите) и имейл уведомяване при не наличност. Механизма е приложим основно за “air-gapped” среди;
- Извеждане на метод за валидирана работоспособност чрез контролирано тестване - целенасочено изключване - и детайлно описание на необходимите стъпки за “troubleshooting”, включващи директен/локален достъп, проверка на Интернет свързаността и/или проверка на Azure D4IoT-Sentinel интеграцията.

ГЛАВА 3. АВТОМАТИЗИРАНИ МЕХАНИЗМИ ЗА ИЗВЕСТЯВАНЕ И ОПЕРАТИВЕН МОНИТОРИНГ В ИНДУСТРИАЛНИ МРЕЖИ

3.1 Откриване на атаки срещу индустриални мрежи, целящи понижаване на версията на фърмуера

Firmware downgrade атаките се отличават с това, че злоупотребяват с легитимни процеси по обновяване и по този начин прикриват злонамерените действия като нормална поддръжка. В OT/ICS контекст това е критично, защото регресия към по-стара версия може умишлено да върне отново известни уязвимости и да създаде предпоставка за последваща компрометация на контролни системи, загуба на управляемост и риск за безопасността. Следователно ефективната детекция изисква OT-aware мониторинг и аналитика, които да различават не просто “Firmware Change”, а посоката на промяната и нейната рискова стойност.

3.3 Реализация на техническото решение

Проектираното инженерно техническо решение за откриване на Firmware Downgrade атаки е реализирано като комбинация от (1) аналитично правило в Microsoft Sentinel, което генерира инцидент само при регресивна промяна на фърмуера, и (2) автоматизиран седмичен отчет за всички останали „Firmware Change Detected“ аларми, който осигурява governance и контрол върху легитимните обновявания. Източник на данни за решението е алармата “Firmware Change Detected” генерирана от D4IoT и внедрена в Sentinel чрез “Microsoft Defender for IoT” data connector. Основната цел е да се преодолее ограничението

на стандартните платформи, които сигнализират за настъпила фърмуерна промяна, но не различават дали тя е ъпгрейд или даунгрейд, и по този начин да се постигне селективна ескалация на събития с най-висок риск.

3.3.1 Аналитично правило, откриващо Firmware Downgrades

Първият компонент е аналитично правило в Sentinel, което изпълнява version-aware логика: извлича предишната и текущата версия на фърмуера от алармата и ги сравнява по компоненти, за да определи посоката на промяната. Правилото се създава в меню “Analytics”, като в секция “General” се задават име и описание.

В следващата стъпка се дефинира KQL заявка върху таблица SecurityAlert, като се ограничава до D4IoT аларми чрез ProviderName == "IoTSecurity" и се филтрира конкретно по алармата “Firmware Change Detected”. Чрез оператора extend се инициализират променливи и се извличат необходимите параметри от JSON структурите посредством tostring() и parse_json(). Версиите “UpdatedFrom” и “UpdatedTo” се парсват и се екстрахират числовите стойности чрез extract() и регекс: @"Software Revision:\s+([^;]+);"

След това версиите се разделят на компоненти (Major, Minor, Patch, Build) чрез split(), като всеки компонент се преобразува към числов тип с toint(). По този начин се създават отделни променливи за старата и новата версия, което позволява коректна “побитова” съпоставка.

Самото сравнение се реализира с case(), като резултатът класифицира промяната като “OldVersion is newer” (downgrade), “NewVersion is newer” (upgrade) или “Versions are equal”. Инцидент се генерира само когато старата версия е по-висока от новата, т.е. при регресия. Избраната информация (устройство, версии, време, сензор и др.) се извежда чрез project. След дефинирането на KQL логиката, в “Incident settings” се активира създаването на инциденти при съвпадение и правилото се записва (Save) в “Review + create”.

Изборът на KQL + Regex подход е целенасочен: логиката се изпълнява в Sentinel като единична атомна операция, което осигурява бързина (детекция и инцидент в рамките на около минута) и елиминира външни зависимости. При анализ на алтернативи (например Azure Functions или Python скриптове) се установява, че те въвеждат допълнителна латентност и оперативна сложност (поддръжка на код, версии, мащабиране), без да осигуряват съществена функционална полза спрямо вградените възможности на Sentinel (parse_json(), extract(), split(), case()). Регекса е съобразен с начина, по който D4IoT анотира фърмуерните версии – най-често като 3- или 4-компонентен низ (например “2.6.6.6”).

Допълнително, при проектирането е извършена съпоставка с конкурентни решения (Nozomi Guardian и Claroty CTD), която показва сходна базова функционалност: те отчитат фърмуерна промяна, но по подразбиране не определят дали промяната е прогресивна или регресивна. Това потвърждава практическата добавена стойност на version-aware логиката като механизъм за разрушаване на “симетрията” между легитимни и злонамерени фърмуерни потоци.

3.3.2 Седмичен отчет за всички Firmware Changes

Вторият компонент е спомагателна автоматизация, която управлява останалите “Firmware Change Detected” аларми – т.е. легитимните ъпгрейди – без да се създава постоянен оперативен шум в SOC. Целта е тези събития да не се игнорират напълно, а да се наблюдават с подходяща честота и контекст. Седмичният отчет има два основни сценария на приложение:

1. откриване на аномалии (например 5–10× увеличение на броя обновени устройства, което може да индикира координирана промяна към конкретна версия с известни уязвимости);
2. подпомагане на compliance/регулации, изискващи регистър на активи и промени по тях (change log).

Автоматизацията е реализирана чрез Azure Logic apps, който веднъж седмично изпълнява KQL заявка в Sentinel за "Firmware Change Detected" аларми за последните 7 дни, структурира резултатите в CSV файл и изпраща имейл до предварително дефинирани получатели (например екип по поддръжка на индустриалното оборудване). Logic apps се стартира по график (например всеки петък в 12:00), изпълнява "Run query and list results", проверява дали има резултати (IF условие), създава CSV (Create CSV table) и изпраща доклада като прикачен файл чрез "Send an email". По този начин легитимните промени се управляват отчетно, а SOC остава фокусиран върху високорискови downgrade инциденти.

3.4 Тестване на работоспособността на техническото решение

За да се изпробва и докаже работоспособността на предложеното техническо решение за откриване на Firmware Downgrade атаки са проведени тестове в два етапа, базирани на целенасочено понижаване на фърмуера на Atlas Copco Power Focus 6000 контролер. Първоначално механизмът е внедрен и валидиран в специална непроизводствена среда на промишлено предприятие (инсталация за предварително въвеждане в експлоатация). След доказване на работоспособността и практическата му стойност, обхватът на инсталацията е разширен с цел покриване на цялата индустриална мрежа на производственото предприятие, включително активиране на седмичен доклад за всички фърмуерни промени.

Първият етап на тестовете е проведен в контролирана тестова постановка, съставена от следните компоненти:

- Atlas Copco Power Focus 6000 – централен контролер за затягане, типично свързан към една или повече машини за завинтване;
- Moxa AWK-3131A – индустриална безжична точка за достъп (IEEE 802.11 a/b/g/n), използвана за свързване на контролера и панелния компютър към безжичната мрежа;
- Beckhoff CP2916-0000 – индустриален панелен компютър (процесор + HMI) с Windows 10 IoT, използван за наблюдение и управление на процеси.

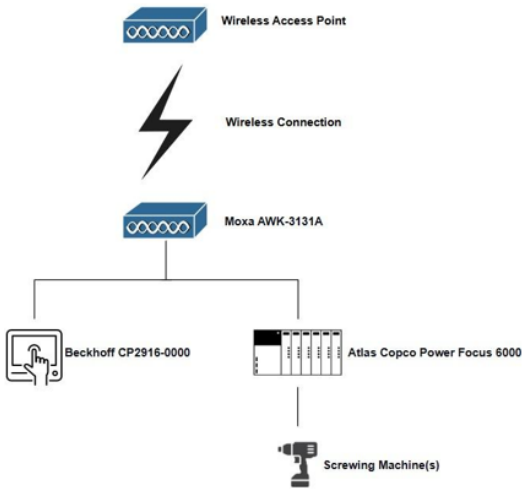
Мрежовата топология на тестовата среда е проектирана така, че да отговаря максимално на реалната индустриална среда на предприятието, с цел по-обективни резултати и улеснено последващо внедряване. Логическата адресация е конфигурирана както следва: Moxa – 10.10.10.10/28, Beckhoff – 10.10.10.1/28, Atlas Copco – 10.10.10.2/28. Следва да се отбележи, че в реални производствени среди фърмуерни промени обичайно се разгръщат от централизирана система (сървър), а не от локално устройство, както е реализирано в тестовата постановка, но това не влияе на валидността на проверяваната детекционна логика.

Таблично тестовете могат да бъдат обобщени в пет основни сценария: (1) откриване на устройството и идентифициране на текущата фърмуер версия; (2) понижаване на фърмуера и очаквана "Firmware Change Detected" аларма; (3) генериране на Sentinel инцидент "D4IoT – Firmware Downgrade Detected"; (4) повишаване (възстановяване) на фърмуера и генериране на втора "Firmware Change Detected" аларма; (5) потискане на инцидент при upgrade, т.е. втората аларма да не води до ескалация. Отчетените резултати потвърждават

очакванията: D4IoT открива първоначална версия 2.6.6.6, след което при понижаване до 2.5.7.2 генерира “Firmware Change Detected” аларма. Приблизително една минута след алармата, аналитичното правило в Sentinel създава инцидент “D4IoT – Firmware Downgrade Detected”, базиран на резултата от version-aware сравнението.
Втората фаза на първия етап включва “научаване” (Learn) на алармата, така че D4IoT да обнови базата си с текущата версия и да може да засече последваща промяна. След възстановяване на контролера до оригиналната версия 2.6.6.6 D4IoT генерира втора “Firmware Change Detected” аларма, но за разлика от downgrade събитието, тя не води до нов Sentinel инцидент, тъй като аналитичното правило коректно класифицира промяната като upgrade и потиска ескалацията. Това се потвърждава чрез търсене на инциденти от правилото, при което се открива само един резултат – първоначалният downgrade.
С приключване на първия етап е изпълнена основната цел на тестването: доказано е, че решението генерира инцидент само при регресивна промяна и по този начин намалява алармената умора, като фокусира вниманието на SOC върху най-високорисковите сценарии. Вторият етап разширява приложимостта на решението в производствена среда, като обхватът е увеличен до цялата индустриална мрежа, а допълнителната автоматизация (Logic apps) изпраща седмичен доклад за всички “Firmware Change Detected” аларми в CSV формат по имейл до предварително дефинирани получатели. Така легитимните фърмуерни промени се управляват отчетно и контекстно, докато потенциалните downgrade атаки се ескалират незабавно като инциденти. Експерименталните резултати са представена на Таб 3.1 и Фиг 3.21, 3.22, 3.23, 3.24, 3.25, 3.26.

Устройство	Логически адрес	Мрежова маска
Moxa AWK-3131A	10.10.10.10	255.255.255.240
Beckhoff CP2916-0000	10.10.10.1	255.255.255.240
Atlas Copco Power Focus 6000	10.10.10.2	255.255.255.240

Табл 3.1 Логическа адресация на опитната постановка



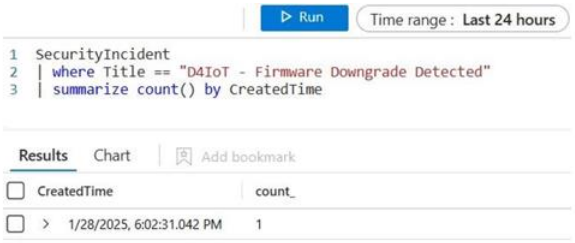
Фиг 3.21 Мрежова топология на тестовата среда

TimeGenerated [UTC]	AlertName	Source	Destination	NewVersion	OldVersion	CompareResult
1/28/2025, 6:01:26.958 PM	Firmware Change Detected	10.10.10.1	10.10.10.2	2.5.7.2	2.6.6.6	OldVersion is newer

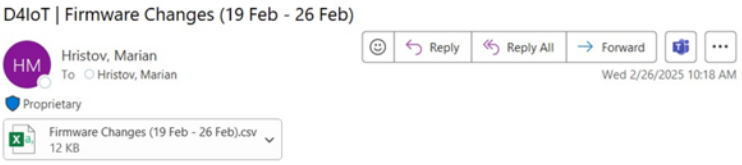
Фиг 3.22 “Firmware Change Detected” аларма вследствие на понижаване на версията на фърмуера

Фиг 3.23 "D4IoT - Firmware Downgrade Detected" инцидент в Sentinel

Фиг 3.24 "Firmware Change Detected" аларма вследствие на повишаване на версията на фърмуера



Фиг 3.25 Налични "D4IoT - Firmware Downgrade Detected" инциденти



Фиг 3.26 Имейл съдържащ седмичен отчет "Firmware Change Detected"

3.5 Заключение

Представената информация в Глава 3 от дисертационния труд аргументира, че при съвременните кибератаки срещу OT/ICS организации нараства значението на заплахите, които не разчитат на класическите техники за атакуване, а вместо това на злоупотреба с легитимни процеси и функционалности в съществуващите устройства, системи и автоматизирани индустриални потоци. Понижаване на версията на фърмуера (Firmware Downgrade) атаките са типичен пример за това – злонамерените действия биват прикрити посредством имитацияна на нормалния просец по обновяване (Firmware Upgrade), което създава симетрия между легитимни и нелегитимни операции, усложнявайки детекцията. В OT/ICS този тип атаки са особено критични, тъй като възстановяването на по-стара версия на фърмуера може умишлено да въвежда уязвимости, които впоследствие да бъдат използвани за компрометиране на системите, загуба на видимост и управляемост на процеса и потенциални рискове за човешката безопасност.

Основният принос на разработеното техническо решение за откриване на потенциални Firmware Downgrade атаки постига разрушаване на функционалната симетрия между процесите на повишаване и понижаване, като се основополага на поведенческо профилиране на OT/ICS устройства и протоколен анализ, реализирани с пасивен, без агент, мрежов мониторинг на D4IoT и корелационните възможности на Sentinel. Вместо да разчита единствено на "Firmware Change" алармата като бинарно събитие, предложеното решение въвежда логика съобразена с версиите (Version-aware), която сравнява предишна

и настоящата версия на фърмуера и генерира инцидент само при регресия (Downgrade). Тази функционалност на системата адресира системен недостатък, наблюдаван и при конкурентните на Microsoft, - Nozomi и Claroty – които също като D4IoT по подразбиране уведомяват за настъпили промени, но не индикират дали са прогресивно обновяване или регресивно връщане.

Резултатите от проведените експерименти демонстрират, че подходът работи надеждно както в тестова, така и в реална производствена среда - D4IoT открива промяната, а Sentinel генерира един единствен инцидент за Downgrade събитието в рамките на приблизително една минута. Същевременно, при последващо възстановяване на по-новата версия (Upgrade), проектираното техническо решението коректно потиска и предотвратява създаването на инцидент, което намалява алармена умора (Alert Fatigue) и фокусира вниманието на центъра за информационна сигурност върху събития с най-висок риск. Това от своя страна води до изпълнението на две от предварително зададените цели и задачи към дисертационния труд, а именно: Реализация на предложеното техническо решение в тестова и реална индустриална среда / Оценка на ефективността на разработената система чрез практически експерименти, тестови подходи и реална работна функционалност.

В допълнение, автора представя спомагателна – незадължителна – функционалност включваща автоматизирани седмични доклади за останалите “Firmware Change Detected” аларми, тоест обновените устройства и системи. По този начин ъпгрейдите във версиите на фърмуера не се игнорират, а се управляват с подходяща честота и контекст.

В заключение, научно-изследователския труд показва, че пасивният, без агент, мрежов мониторинг, комбиниран със SIEM-ориентирана, аналитична логика съобразена с версиите, може ефективно да различава легитимни фърмуерни промени от злонамерени регресивни опити, въпреки високата прилика между двата процеса. Практическият ефект е намаляване на пропуските в OT/ICS видимостта, ускоряване на детекцията и ограничаване на риска от използването на въведени уязвимости в по-старите фърмуерни версии.

3.6 Принос към глава 3

Приносът на глава 3 може да се обобщи, както следва:

- Създаване на концептуален модел за симетрия на сигурността при “firmware lifecycle” базиран на “downgrade” злонамерено действие, маскирано като легитимен “upgrade” и представящ аргументи за необходимостта симетрията да бъде дисектирана посредтвом поведенческо профилиране;
- Извеждане на метод за “version-aware” детекция на понижаване на фърмуера в пасивен OT/ICS мониторинг контекст, който трансформира стандартната “Firmware change” аларма от общ сигнал в конкретна индикация за регресия и потенциално ре-въвеждане на уязвимости в по-старите фърмуерни версии; Интерпретиране през призмата на MITRE ATT&CK for ICS - T0857 - и избиране на протоколен анализ - DS0029 - за практическа основа за детекция в среди без агенти и без активно сканиране;
- Създаване и успешно внедряване на Sentinel аналитично правило, което прави разбор на ExtendedProperties от съответната аларма, - “Firmware Change Detected” - извлича версии посредством Regex, разпределя отделните компоненти и извършва “digit-by-digit” сравнение за определяне на потенциалното фърмуерно понижение;

- Валидиране на “test bed” в непроизводствена среда с реални индустриални устройства и съответинте сценарии, - засичане на фърмуерно понижаване -> инцидент и засичане на фърмуерно повишаване -> подтискане на инцидент - водещо до намаляване на SOC алармена умора и фокусиране върху високорискови събития;
- Създаване и успешно внедряване на автоматизация за седмични доклади относно фърмуерни промени - Azure Logic apps -> CSV -> имейл - с цел създаване на управленчески механизъм за откриване на потенциални аномални модели - масова миграция към определена фърмуерна версия - и за регулаторно съответствие.

ГЛАВА 4 . Аналитични модели за детекция и разследване: malware индикации и контекстно обогатяване на инциденти

4.1 Система за наблюдение и анализ на събитията – Splunk Enterprise

Splunk Enterprise представлява платформа за централизирано събиране, съхранение, търсене и анализ на събития (логове) от множество хетерогенни източници, като осигурява единна видимост - “single pane of glass” - за екипите по сигурност. Чрез индексиране на данните, разнообразни SPL заявки и механизми за алармиране, системата позволява наблюдение в реално време и корелация на събития, както и изграждане на отчети и визуализации, подпомагащи разследване и реакция. В контекста на дисертационния труд Splunk се разглежда като приложим SIEM компонент за мониторинг на Индустриални и IoT среди, включително като алтернатива в случаи, при които се използват локални инфраструктури и/или ограничена облачна свързаност.

4.3 Конфигуриране на правила за генериране на аларми

Конфигурирането на правила за генериране на аларми е ключов елемент от предложеното инженерно решение, тъй като позволява превръщането на логове и IDS/IPS събития в оперативно приложими сигнали към екипа по сигурност. Подходът е проектиран така, че да осигурява уведомяване в реално време при високорискови събития и периодична отчетност при по-нискорискови индикатори, което подпомага баланса между чувствителност и контрол върху алармения шум.

4.3.1 Генериране на аларма за всеки успешен опит за влизане на SIEM инфраструктурата

Добра практика при управлението на мониторинга е SIEM инфраструктурата да бъде инсталирана на отделна машина, предназначена единствено за наблюдение и съхранение на събития. В такъв сценарий успешните влизания в системата не са често очаквани и обикновено са резултат от планирани промени или отстраняване на неизправности. Поради това всяка успешна автентикация – особено атипична спрямо време, източник или акаунт – следва да се разглежда като потенциален индикатор за злонамерена активност.

В рамките на решението е създадена аларма в реално време, която се задейства при всеки успешен опит за влизане в SIEM системата. Алармата използва два основни критерия: (1) логовете да са налични в специализирания индекс “siem_local”, и (2) EventCode = 4624, който съответства на Windows събитие “An account was successfully logged on”. Настройката е реализирана като real-time, Per-Result аларма, така че всеки отделен резултат, отговарящ на условията, да задейства уведомяване.

Към алармата са добавени две действия: запис в SIEM конзолата и автоматично имейл уведомление до екипа по сигурност. Имейлът съдържа основните детайли, включително CSV с атрибутите на събитието и линк към алармата в Splunk. Практическата стойност е,

че екипът може да реагира незабавно и да верифицира: кой е акаунтът, оторизиран ли е достъпът и каква е причината за влизането.

4.3.2 Автоматичен доклад за неуспешните опити за влизане на SIEM инфраструктурата
Неуспешните опити за влизане също са важен индикатор, но за разлика от успешните логини, те не винаги изискват мигновено отреагиране. Често причините са тривиални (грешна парола, грешно потребителско име или изтекъл акаунт), поради което подходяща стратегия е периодичен анализ, а не аларма в реално време.
Затова е конфигуриран автоматичен дневен доклад, който обхваща последните 24 часа и извлича всички неуспешни опити за влизане на база: (1) индекс “siem_local” и (2) EventCode = 4625, съответстващ на Windows събитие “An account failed to log on”. Докладът се изпълнява по график (например в 12:00) и предоставя обобщена картина за тенденции и аномалии (например множество неуспешни опити за даден акаунт), без да генерира постоянни аларми и оперативен шум.

4.3.3 Генериране на аларма за CnC комуникация от вътрешна машина
Наблюдението на IDS/IPS събития е необходимо, тъй като съвременните атаки често включват компрометиране на вътрешни системи и опити за комуникация с Command-and-Control (CnC) инфраструктура. Макар по-голямата част от IPS трафика да е входящ и да се блокира на периметъра, изходящият трафик от локалната мрежа към Интернет е особено критичен, защото може да индикира компрометирана система и активна зловредна дейност.

В решението е конфигурирана аларма в реално време, която се задейства при наличие на зловреден изходящ IDS/IPS трафик (например детекция на Mirai.Botnet). Алармата записва събитието в SIEM конзолата с висок приоритет и изпраща автоматичен имейл с линк към алармата и CSV с детайлите. По този начин SOC получава незабавна видимост за потенциална компрометация дори когато защитната стена е блокирала комуникацията, тъй като самият опит е индикатор за инфекция и риск от участие в DDoS мрежа.

4.3.4 Генериране на аларма за умишлено изтриване на лог с цел прикриване на следи
При успешни атаки често се наблюдава опит за прикриване на следи чрез изтриване на логове. Поради това инженерното решение включва аларма за изчистване на Windows Security log, базирана на EventCode = 1102 (“The audit log was cleared”). Алармата е конфигурирана като real-time, Per-Result и използва индекс “siem_local” като източник.

При задействане се извършват две действия: запис в SIEM конзолата с висок приоритет и автоматичен имейл до екипа по сигурност с детайли и линк към алармата. Това осигурява незабавно внимание към потенциален опит за прикриване на злонамерена активност и подпомага бързо предприемане на мерки.

4.4 Тестване на работоспособността на предложеното инженерно решение

За да се провери работоспособността на предложеното инженерно решение, са проведени контролирани тестове, които валидират коректното задействане на всички конфигурирани аларми и отчети, както и изпращането на автоматични имейл уведомления. Тестването е организирано по отделните сценарии, съответстващи на правилата от предходната секция, като целта е да се потвърди, че SIEM инфраструктурата осигурява своевременно алармиране и предоставя достатъчно контекст за започване на разследване.

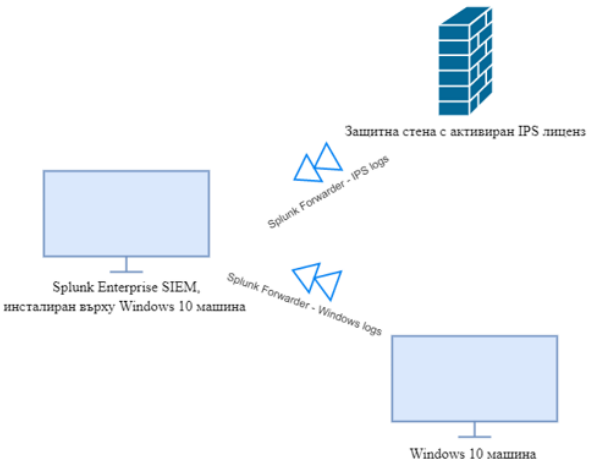
4.4.1 Получаване на аларма за всеки успешен опит за влизане на SIEM инфраструктурата
С цел тестване на имейл уведомленията при успешен логин е извършен успешен опит за влизане в машината, на която е инсталиран Splunk Enterprise. В резултат е задействана алармата, базирана на EventCode 4624 в индекс “siem_local”, и е изпратен автоматичен имейл до предварително зададен получател. Уведомлението съдържа кратко описание на събитието, прикачен CSV файл с детайлите и линк към конкретната аларма в SIEM. Практическата стойност на този тест е, че потвърждава възможността екипът по сигурност да получава сигнал в реално време за потенциално неоторизиран достъп и да извърши незабавна проверка кой акаунт е използван, от къде и в какъв контекст.

4.4.2 Генериране на доклад съдържащ неуспешните опити за влизане на SIEM инфраструктурата
За валидиране на автоматичния доклад за неуспешни логини е извършена промяна на графика на изпълнение с тестова цел: вместо веднъж дневно, докладът е задействан на всеки час и обхваща последните 30 минути. В рамките на тестовия прозорец (06:30–07:30) докладът открива 13 неуспешни опита за влизане, базирани на EventCode 4625. Допълнителен преглед на детайлен лог показва конкретните параметри на опитите, включително потребителски акаунт, тип логин (например “2” – интерактивен вход на самата машина) и причина за неуспех (“Unknown user name or bad password”). Тестът потвърждава, че докладът е подходящ механизъм за периодично проследяване на тенденции и аномалии, без да създава излишен оперативен шум.

4.4.3 Получаване на аларма за CnC комуникация от вътрешна машина
За тестване на IDS/IPS алармата, предназначена да засича зловредна изходяща комуникация (например Mirai.Botnet), е използван контролиран подход за репродукция на събитието без риск за тестовата среда. Изпратена е специално подготвена заявка към Splunk, която симулира необходимото събитие и задейства конфигурираното действие (sendemail). В резултат е получен имейл, съдържащ ключови детайли за алармата – източник, дестинация, портове, предприето действие (например blocked/dropped) и друга релевантна информация. Прегледът на алармата в SIEM потвърждава наличието на достатъчен контекст за първоначална оценка и реакция. Тестът демонстрира и практическата интерпретация на подобни случаи: дори при успешно блокиране на изходящия трафик, вътрешната система остава потенциално компрометирана и е необходимо ограничаване (карантина) и последваща ремедиация, за да се предотврати разпространение.

4.4.4 Получаване на аларма за умишлено изтриване на лог с цел прикриване на следи
С цел валидиране на алармата за изчистване на Security log е извършено контролирано изтриване на “Security” лога чрез Event Viewer. Това действие генерира Windows събитие 1102 (“The audit log was cleared”), което съответно задейства конфигурираната real-time аларма в Splunk (index “siem_local”, EventCode 1102). В резултат е изпратен автоматичен имейл с CSV детайли и линк към алармата в SIEM. Прегледът на алармата предоставя необходимата информация за разследване, като тестът потвърждава, че решението успешно открива действия, характерни за опит за прикриване на следи – типичен елемент от финалните фази на кибератака.
В обобщение, проведените тестове доказват коректната работа на внедрените аларми и отчети, както и надеждността на автоматичните уведомления. Решението осигурява

своевременно алармиране при високорискови събития (успешни логини, IDS/IPS индикации, изчистване на логове) и контролирана отчетност при по-нискорискови индикатори (неуспешни логини), като предоставя оперативно приложим контекст за започване на разследване и предприемане на адекватни мерки. Фиг 4.15, 4.55, 4.56 и Таб 4.1.

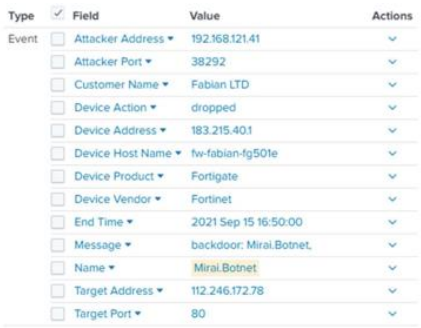


Фиг 4.15 Топология на представеното техническо решение

Име	Логически адрес	Предназначение
DESKTOP-3GD6VLQ	192.168.0.106	Splunk SIEM
DESKTOP-5TG45DH	192.168.0.107	Splunk Forwarder

Табл 4.1 Логическа адресация на представеното техническо решение

Фиг 4.55 Получения имейл за IDS/IPS алармата



Фиг 4.56 Налични детайли в IDS/IPS алармата

4.6 Заключение

Представеният обзор на проблема в Глава 4 от дисертационния труд показва, че OT/ICS киберсигурността не може вече да се разглежда като допълнение на класическата ИТ сигурност, а като самостоятелна и критична област, в която водещи приоритети са непрекъсваемостта на процеса и безопасността. С развитието на четвъртата индустриална революция, - представена в 1.1 - ИТ-ОТ интеграцията и нарастващата отдалечена свързаност повишават ефективността на Индустриалните системи, но едновременно с това разширяват експозицията за потенциални атаки и увеличават вероятността от инциденти с оперативен и физически ефект.

Обзорът на заплахите за периода 2022-2025 г. потвърждава устойчивото нарастване както в обем, така и в зрелостта на злонамерените извършители, целящи по негативен начин да въздействат на индустриални организации. Наблюдава се експоненциално нарастване на рансъмуер инцидентите (605 през 2022, 905 през 2023, и над 3300 засегнати индустриални организации през 2025), което заключава, че киберпрестъпността остава доминиращ риск за индустрията. Паралелно с това се наблюдава и друга осезаема промяна, а именно появата и усъвършенстването на ICS-специфични зловредни инструменти и тактики, насочени към детайлно разбиране и целящи въздействие върху самия индустриален процес, а не само таргетиране на ИТ инфраструктурата.

Представените особености, а оттам и произлизащите слабости в индустриалните среди са до голяма степен системни и повтаряеми: ограничена видимост, недостатъчна сегментация, неконтролиран отдалечен достъп и споделени ИТ-ОТ креденшъли. Особено критичен е проблемът с видимостта, тъй като без OT-aware мониторинг организациите не могат нито да засичат навреме аномалии, нито да провеждат надеждно разследване по време на инцидент. Данните на Dragos потвърждават това - ограничена или изцяло липсваща видимост в ОТ е сред най-честите находки в реални среди, а глобално едва малка част от ОТ мрежите имат внедрен адекватен мониторинг.

В обобщение, Глава 4 изгражда необходимата теоретична и аналитична основа за практическата част на дисертационния труд, като същевременно изпълнява част от поставените цели и задачи на научно-изследователския труд. А именно: Идентифициране на заплахи и основни рискове в OT/ICS, Анализ на съществуващите решения за справяне с тях, Проучване на технологии за мониторинг и анализ на събитията в OT/ICS и Разработване на система за наблюдение на OT/ICS базирана на резултатите от направеното задълбочено проучване. За финал, Глава 1 представя първия компонент от разработеното инженерно решение - Splunk Enterprise имплементация с цел откриване на DDoS атаки в IoT.

4.7 Приноси към глава 4

Приносът на глава 4 може да се обобщи, както следва:

- Създаване на SIEM-ориентиран модел за детекция на различини фази на традиционна кибератака, Първоначален достъп, Изтичане на данни, Прикриване на следи - рализиран посредством корелация на Операционни (Windows) и мрежови (IPS) логове в единна система;
- Дефиниране на измерим подход за реално-времева детекция посредством класификация на алармите - real-time или scheduled reports - и обосноваване на избора спрямо оперативната стойност на индикаторите;

- Интеграция на OSINT-обогатяване в процеса на разследване целяща да повиши доказателствената стойност на детекцията базирайки се на аналитичната методология (потвърждаване на злонамерения контекст);
- Реализирана работеща референтна Splunk Enterprise архитектура, осъществена посредством отделни индекси за локални, отдалечени и мрежови логове, същевременно демонстрирайки типичен производствен подход за внедряване на данни;
- Разработен набор от SPL-базирани правила за аларми в реално време и такива, които биват преглеждани на седмична база:
    - Windows събитие 4624 - подозрителни успешни опити за влизане на SIEM инфраструктурата;
    - IPS Mirai.Botnet лог - потенциално изтичане на данни и/или комуникация с CnC;
    - Windows събитие 1102 - опит за прикриване на следи от злонамерени действия;
    - Windows събитие 4625 - доклад за аномалии при неуспешни опити за влизане на SIEM инфраструктурата.
- Валидиране чрез контролирани тестове, - възпроизвеждане на 4624, 1102, 4625 и IPS Mirai.Botnet - доказващи повторяемост и приложимост, както и извеждане на практическа рамка за намаляване на “alert fatigue” посредством разграничаване на високо-приоритетни реално-времеви аларми от ниско-спешни индикатори подходящи за доклад. Това цели оптимизация на SOC натоварването.

ПРИНОСИ В ДИСЕРТАЦИОННИЯ ТРУД

Темата на настоящия дисертационен труд е свързана с изследване и разработване на облачно-базирана система за наблюдение и анализ на събития в Индустриални (OT/ICS) мрежи. Индустриалните среди имат специфични ограничения и приоритети – непрекъсваемост на процеса, безопасност, дълъг жизнен цикъл на оборудването и ограничени възможности за внедряване на агентни решения или активни сканирания. Това налага използване на пасивни подходи за видимост, централизирана корелация на събитията и надеждни механизми за автоматизирана реакция. В дисертационния труд са анализирани актуални заплахи и системни слабости в OT/ICS, както и са реализирани и валидирани конкретни инженерни механизми, които подобряват връзката между детекция и реакция, надеждността на мониторинга и разграничаването на легитимни от злонамерени активности.

Приносите в дисертационния труд могат да се обобщят, както следва:

1. Представен е цялостен анализ на OT/ICS заплахите и системните слабости (видимост, сегментация, отдалечен достъп и идентичности), като на тази основа са формулирани архитектурните изисквания и целите към облачно-базирана система за наблюдение и анализ на събития в индустриални среди.
2. Разработен и реализиран е механизъм за нотификация в реално време при откриване на зловреден софтуер в индустриална мрежа чрез интеграция на пасивен OT мониторинг (Microsoft Defender for IoT), SIEM корелация (Microsoft Sentinel) и

автоматизация (Azure Logic apps), с контекстно обогатяване (SensorId/Location), което превръща алармите в кратки и оперативно приложими уведомления към SOC.

3. Разработен и валидиран е многокомпонентен оперативен на D4IoT внедряване, включващ независими механизми за откриване на деградация/загуба на видимост (Sentinel правила за липса на аларми, Logic Apps/ARG мониторинг на cloud свързаност с anti-flood логика и локален SNMP/Zabbix мониторинг), приложим и в среди без Интернет свързаност.
4. Разработен е version-aware механизъм за детекция на firmware downgrade атаки, който разрушава симетрията между легитимни и злонамерени фърмуерни промени чрез пасивен мониторинг (D4IoT) и SIEM аналитика (Sentinel), генерирайки инцидент само при регресия и потискайки ескалация при легитимни upgrade-и, като допълнително е реализирана автоматизация за седмични отчети на всички firmware промени за governance и контрол.

СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИОННИЯ ТРУД

[A1] V. Dimitrova, **M. Hristov**, K. Nikolova and M. Nenova, "Securing the Edge: A Comparative Analysis of Microsoft Defender for IoT and Nozomi Guardian," 2025 33rd National Conference with International Participation (TELECOM), Sofia, Bulgaria, 2025, pp. 1-4, doi: 10.1109/TELECOM66943.2025.11304075.

[A2] **Hristov, M**.; Nenova, M.; Dimitrova, V. Security Symmetry in Embedded Systems: Using Microsoft Defender for IoT to Detect Firmware Downgrade Attacks. Symmetry 2025, 17, 1061. https://doi.org/10.3390/sym17071061 . **Citations: 1**

[A3] **M. Hristov**, M. Nenova, K. Nikolova and V. Dimitrova, "Health Monitoring of Microsoft Defender for IoT Implementation," 2023 31st National Conference with International Participation (TELECOM), Sofia, Bulgaria, 2023, pp. 1-4, doi: 10.1109/TELECOM59629.2023.10409719. **Citations: 1**

[A4] B. Stoytcheva, M. Nenova, **M. Hristov**, Z. V. Djarvis and G. Balabanov, "A specifics overview of Systems for monitoring security events in different areas," 2023 XXXII International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 2023, pp. 1-4, doi: 10.1109/ET59121.2023.10279205.

[A5] **M. Hristov**, M. Nenova, Z. Valkova-Jarvis and V. Dimitrova, "Notification mechanism for malware detected by Microsoft Defender for IoT in industrial networks," 2023 14th National Conference with International Participation (ELECTRONICA), Sofia, Bulgaria, 2023, pp. 1-4, doi: 10.1109/ELECTRONICA58875.2023.11173900.

[A6] **M. Hristov**, M. Nenova, G. Iliev and D. Avresky, "Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT," 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 2021, pp. 1-5, doi: 10.1109/NCA53618.2021.9685977. **Citations: 47**

TECHNICAL UNIVERSITY OF SOFIA

FACULTY OF TELECOMMUNICATIONS

DEPARTMENT OF "COMMUNICATION NETWORKS"

Marian Hristov

Investigation of a Cloud-based Security Information and Event Management Sistem

ABSTRACT of Ph.D THESIS

The title of this thesis is related to the research and implementation of a cloud-based system for monitoring and analyzing security events in Industrial (OT/ICS) environments. Modern OT/ICS infrastructures face unique constraints and priorities - process continuity, safety, long asset lifecycles, and frequent limitations for agent-based protection or active scanning. At the same time, Industry 4.0 and IT–OT convergence increases connectivity and operational efficiency but expand the attack surface and the probability of incidents with operational and even physical impact. Therefore, OT environments require passive, OT-aware visibility, centralized correlation, and automation that reliably connects detection to response and preserves observability even when parts of the monitoring stack degrade.

The thesis consists of 4 chapters and is supported by extensive figures and tables illustrating the proposed architectures, rule logic, automation workflows, and validation results. The analysis, methods, and results are presented through 6 author publications, covering (1) SIEM-based detection in IoT environments, (2) real-time notification mechanisms for malware detections in industrial networks, (3) health/operational monitoring of an OT monitoring deployment, and (4) version-aware detection of firmware downgrade activity using passive OT monitoring and SIEM analytics. Chapter 1 motivates the problem by demonstrating that OT cyber threats can extend beyond data theft into operational disruption and safety-relevant consequences, and it introduces a practical detection-to-response approach through real-time notification. Chapter 2 addresses the reliability challenge by proposing a multi-layer operational monitoring solution for the monitoring infrastructure itself. Chapter 3 focuses on advanced OT threat scenarios that abuse legitimate processes - specifically firmware downgrades - and presents a version-aware analytic mechanism that distinguishes benign upgrades from potentially malicious regressions. Chapter 4 complements the cloud-centric architecture with SIEM-based detection workflows in IoT, demonstrating unified monitoring, alerting, and investigation patterns.

The major contributions of the thesis are listed - 1. A structured OT/ICS security framework is developed that links Industry 4.0 exposure, OT-specific constraints, and systemic weaknesses (visibility, segmentation, remote access, identities) to concrete monitoring and response requirements for a cloud-based event analytics system. 2.A real-time notification mechanism is designed and validated for malware-related detections in industrial networks by integrating passive, agentless OT monitoring with SIEM correlation and automation, producing context-rich, operationally actionable alerts for rapid SOC response. 3. A multi-component health/operational monitoring approach is implemented and tested, combining independent indicators and notification channels (SIEM rules, cloud connectivity checks with automation, and local SNMP-based monitoring) to detect silent failures and prevent loss of OT visibility. 4.A version-aware firmware downgrade detection method is proposed and validated, leveraging passive OT monitoring and SIEM analytics to break the symmetry between upgrades and downgrades by generating incidents only for regressions, while supporting governance through automated periodic reporting of firmware changes.

TECHNICAL UNIVERSITY OF SOFIA

FACULTY OF TELECOMMUNICATIONS

DEPARTMENT “COMMUNICATION NETWORKS”

MEng. Marian Hristov

RESEARCH OF A CLOUD-BASED SYSTEM FOR EVENT MONITORING AND ANALYSIS

THESIS SUMMARY

for awarding the educational and scientific degree
"PhD"

Field: 5. Engineering Sciences

Professional field: 5.3 “Communication and Computer Engineering”

Scientific specialty: “Communication Networks and Systems”

Supervisors: Assoc. Prof. Maria Valcheva Nenova, PhD

Assoc. Prof. Elica Emilova Gieva, PhD

Sofia, 2026

The dissertation was discussed and approved for defense by the Department Council of the "Communication Networks" Department at the Faculty of Telecommunications, TU–Sofia, at a regular meeting held on 06.04.2026.

The public defense will be held on 24 July 2026 (Friday) at 13:00 in the conference hall of the BIC at the Technical University of Sofia, at an open session of the Scientific Jury appointed by Order No. ОЖ-5.3-35 of 23 April 2026 of the Rector of TU-Sofia, composed as follows:

1. Assoc. Prof. Eng. Kiril Marinov Kasev, PhD – Chairperson
2. Assoc. Prof. Eng. Georgi Rumenov Tsochev, PhD – Scientific Secretary
3. Prof. Eng. Stanimir Mihaylov Sadinov, PhD
4. Prof. Eng. Gabriela Lachezarova Atanasova, PhD
5. Assoc. Prof. Eng. Ivelina Stefanova Balabanova, PhD

Substitute members:

1. Prof. Eng. Ivaylo Ivanov Atanasov, DSc
2. Prof. Eng. Valentina Ilieva Markova, PhD

Reviewers:

1. Assoc. Prof. Kiril Marinov Kasev, PhD
2. Assoc. Prof. Gabriela Lachezarova Atanasova, PhD

The defense materials are available to interested parties at the office of the Faculty of Telecommunications of TU-Sofia, Block No. 1, Room No. 1306.

The doctoral candidate is an external (part-time) PhD student at the "Communication Networks" Department, Faculty of Telecommunications. The research was carried out by the author, with some parts supported by research projects.

Author: MEng. Marian Hristov

Title: RESEARCH OF A CLOUD-BASED SYSTEM FOR EVENT MONITORING AND ANALYSIS

Number of copies: 10 pcs

Printed at the Publishing House of the Technical University of Sofia

III. GENERAL CHARACTERIZATION OF THE DISSERTATION

Relevance of the problem

The contemporary digital transformation, accelerated by Industry 4.0, leads to increasingly tighter integration between Information Technology (IT) and Operational Technology (OT), as well as widespread adoption of IoT/IIoT devices and remote connectivity in industrial networks. This improves efficiency, scalability, and process automation, but simultaneously expands potential attack vectors and increases the likelihood of cyber incidents with operational and physical impact. Unlike traditional IT environments, a compromise of OT/ICS infrastructure may cause not only data leakage or financial losses, but also disruption of technological processes, safety risks, and potentially endanger human life.

In this context, monitoring, collection, and analysis of events become a key factor for cyber resilience. In practice, however, industrial organizations often face several systemic challenges. First, a large portion of OT devices do not allow agent installation, and active scanning can disrupt process determinism, which necessitates passive, network-based visibility approaches. Second, events and telemetry originate from heterogeneous sources (OT sensors, network devices, security systems, platform and application logs), which complicates correlation and the transformation of detections into clear operational actions. Third, even when detection mechanisms exist, there is often no sufficiently reliable and fast connection between detection and response - i.e., a timely notification channel and a structured response/containment process for security teams.

An additional challenge is the reliability of monitoring itself: when deploying passive, agentless monitoring solutions, the critical question arises – "who monitors the monitoring system?" Loss of visibility due to sensor, integration, or infrastructure issues can leave the environment unprotected precisely when an attack occurs. Moreover, modern threat actors increasingly abuse legitimate functionalities and processes to conceal malicious actions - for example, a firmware downgrade disguised as a normal update process - creating "symmetry" between legitimate and illegitimate operations and increasing detection complexity without context-oriented analytic logic.

For these reasons, the research and development of a cloud-based system for event monitoring and analysis is a relevant applied research problem. A cloud approach provides centralization, scalability, integration of multiple telemetry sources, real-time analytics, and response automation. At the same time, it requires OT-specific solutions: passive operation, high monitoring reliability, context enrichment of alerts, and mechanisms for distinguishing legitimate from malicious activity. Therefore, developing and validating architecture and mechanisms for cloud-based monitoring and event analysis applicable to industrial networks is a significant and timely contribution to improving the cyber resilience of critical environments.

Objective, main tasks, and research methods

The main objective of this dissertation is to **develop and build an end-to-end system for monitoring and analyzing events in industrial (OT/ICS) networks, aiming to improve organizational security by increasing network visibility and enabling timely detection of threats**. To achieve this objective, the following tasks are addressed:

8. Identification of threats and key risks in OT/ICS;
9. Analysis of existing solutions for addressing them;
10. Study of technologies for monitoring and event analysis in OT/ICS;

11. Development of an OT/ICS monitoring system based on the results of the conducted research;
12. Implementation of the proposed technical solution in a test and real industrial environment;
13. Evaluation of the system's effectiveness through practical experiments, test approaches, and real operational functionality;
14. Development of a methodology for OT risk assessment based on the proposed solution and established practices and standards in the field.

Methodological framework

The research methodology combines analytical and experimental/applied approaches. The analytical approach is used for reviewing the state of the problem, defining requirements for a cloud-based system for event monitoring and analysis, and establishing criteria for evaluating its effectiveness in an OT/ICS context. The experimental/applied approach is used to design, implement, and validate the proposed technical mechanisms through controlled test scenarios and practical execution in a test and/or real industrial manufacturing environment. Based on experiments, the dissertation evaluates the integration between passive OT monitoring (D4IoT), the SIEM component (Microsoft Sentinel), and automations (Azure Logic Apps), as well as the effectiveness of real-time notifications, health/operational monitoring, and version-aware detection of firmware downgrade activity.

Scientific novelty

The scientific novelty of this dissertation is related to researching and developing a cloud-based system for monitoring and analyzing events in industrial (OT/ICS) networks, aligned with OT-specific constraints and requirements (passive operation, lack of agents, and stringent continuity and safety requirements). The work analyzes modern threats and systemic weaknesses in OT/ICS, serving as the basis for defining architectural principles and functional requirements for the monitoring and event analysis system. A novel contribution is the development of a multi-component approach for improving monitoring reliability through independent mechanisms for detecting degradation or loss of visibility. In addition, version-aware analytic logic is presented to differentiate legitimate firmware changes from potentially malicious firmware downgrades despite the strong similarity between the two processes. The approaches demonstrate practical effectiveness through validation in test and/or real industrial environments, leading to faster detection and response and a reduction of irrelevant alerts, thereby improving monitoring resilience and OT/ICS risk manageability.

Practical applicability

All developed mechanisms and components of the proposed cloud-based event monitoring and analysis system are implemented and validated through practical experiments in test and/or real industrial manufacturing environments. The effectiveness of the integration between passive, agentless OT monitoring (D4IoT), the SIEM component (Microsoft Sentinel), and the automations (Azure Logic Apps) is evaluated, along with additional mechanisms for operational monitoring and diagnostics. The proposed solutions are applicable in real industrial networks where active scanning and endpoint agents are constrained, and where reliable notification and rapid incident response are required. The mechanisms can be used both for operational alerting and initial triage, and for improving monitoring resilience by detecting degradation or loss of visibility and distinguishing high-risk events (e.g., firmware downgrade) from routine changes .

Publications

The key achievements and results are presented in a total of six publications, including one published in an international scientific journal. Five publications are presented in proceedings of international scientific conferences and national conferences with international participation. All six publications are co-authored with the supervisors. A total of 51 citations are reported, all in indexed publications.

International scientific conferences: 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), 2023 14th National Conference with International Participation (ELECTRONICA), 2023 XXXII International Scientific Conference Electronics (ET), 2023 31st National Conference with International Participation (TELECOM), 2025 33rd National Conference with International Participation (TELECOM).

International scientific journal: MDPI Symmetry, 2025.

Structure and volume

The dissertation is written in Bulgarian and has a volume of 194 A4 pages. It includes an introduction, four chapters, a conclusion with the main contributions, directions for future work, a list of author publications, a list of abbreviations, lists of figures and tables, and a reference list. The dissertation contains 155 figures and 5 tables. A total of 240 references are used, all in English, with 95% from the last ten years. Figure and table numbering in the thesis summary corresponds to the dissertation.

IV. CONTENT OF THE DISSERTATION

CHAPTER 1. THEORETICAL BASIS AND ARCHITECTURAL PRINCIPLES OF CLOUD-BASED SYSTEMS FOR EVENT MONITORING AND ANALYSIS IN INDUSTRIAL NETWORKS

1.2 Main Challenges

Industrial environments have a number of characteristics that make cybersecurity and monitoring substantially different from the IT context. The main challenges include: the need for process continuity and safety; the presence of long-lifecycle devices; limited maintenance windows for patching and restarts; and the risk of disturbing determinism through active scanning. In many cases, installing agents on typical OT devices - PLC/RTU/HMI - is impossible or unacceptable due to production constraints, which necessitates passive, network-based visibility methods. In addition, industrial infrastructures often suffer from systemic weaknesses such as limited visibility, insufficient segmentation, uncontrolled remote access, and shared IT-OT credentials. As a result, even when monitoring technologies are present, a gap is frequently observed between detection and real operational actions, which slows response and increases the risk of escalation.

1.4 Event Monitoring and Analysis in Industrial Networks

Monitoring and analyzing events in OT/ICS networks is a key mechanism for early anomaly detection and incident response, but it is complicated by the heterogeneity of data sources and the limitations on active methods. An effective approach requires centralized telemetry collection - passive traffic analysis, alerts from OT monitoring sensors, system logs, IPS/IDS events - followed by normalization and correlation in a SIEM platform, as well as context enrichment (e.g., location, system criticality, CMDB lookup, TI/OSINT checks). A significant problem is the reliability of visibility itself: sensor failures, loss of cloud connectivity, or integration issues can result in “silent

failures". Therefore, in addition to detection, operational monitoring of the monitoring system is required to signal telemetry degradation and prevent unnoticed loss of visibility.

1.5 Implementation of a Notification Mechanism for Malware Detection in Industrial Networks

1.5.3 Implementation of the Technical Solution

The proposed engineering solution for notifying on malware detection in industrial networks is implemented through three interconnected components: (1) a Microsoft Sentinel analytic rule for detection and incident creation, (2) an Azure Logic Apps playbook for automated email notifications, and (3) a Sentinel Automation rule that links the incident to the playbook and labels processed cases. The solution uses as input the alerts from the Industrial malware analytics module of Microsoft Defender for IoT (D4IoT), ingested into Sentinel via the "Microsoft Defender for IoT" data connector. The core idea is to transform technical detections into a short, context-rich, operationally actionable notification to SOC/CTAC, accelerating response, supporting initial investigation, and reducing escalation risk in OT/ICS environments.

1.5.3.1 Analytic Rule for Malware Detection

The first component is a Sentinel analytic rule that searches for and correlates D4IoT alerts related to malicious activity. The rule is created from the "Analytics" menu, initially assigning a name and description in the "General" section. Then, a Kusto Query Language (KQL) query is defined using the SecurityAlert table and limiting results to D4IoT alerts via ProviderName == "IoTSecurity". To specifically select the Industrial malware analytics module, the query requires ProductComponentName == "MALWARE". This focuses analysis on alerts indicating potential compromise, while excluding the other D4IoT modules (POLICY_VIOLATION, ANOMALY, OPERATIONAL, PROTOCOL_VIOLATION), which have different purpose and semantics.

After filtering, the KQL extend operator is used to initialize variables that extract key alert attributes - logical addresses of the involved systems, the sensor identifier/name, and additional properties needed for downstream notification. Extraction is performed using tostring() and parse_json(), since part of the data is embedded in JSON structures. A critical context-enrichment step follows: the case() function introduces a mapping between sensor name and physical location. For example, the variable Location is assigned the value "Sofia" for sensors whose name contains "sof." This logic transforms a raw technical alert into a signal with operational context, which is especially important in distributed industrial environments with multiple sites. Finally, the project operator selects a subset of attributes (e.g., time, alert type, source system, sensor, location) sufficient for incident creation and for the subsequent email notification.

After defining the KQL logic, the workflow proceeds to "Incident settings," where incident creation on match is enabled. This is essential because the incident serves as the trigger for automation. After validation and "Save" in "Review + create", the rule becomes available in Analytics and begins generating incidents when malware alerts are detected. Thus, the first component provides a reliable mechanism for converting D4IoT alerts into SIEM incidents with appropriate priority, suitable for SOC workflows.

1.5.3.2 Notification Mechanism for Malware Detection

The second component is an Azure Logic Apps playbook that sends automated email notifications when an incident for malicious activity is created. Its role is to reduce notification latency and

ensure rapid engagement of more experienced specialists (Incident Responders) when suspicious events occur in the industrial network. The practical goal is that the shift SOC analyst receives a structured notification with key information, without first needing to manually locate and interpret details in the SIEM console.

Logic Apps creation begins by selecting a hosting plan (Standard or Consumption) and specifying Subscription, Resource group, name, and region. After creation, the structure is built in "Logic app designer" as a sequence of operators. The playbook trigger is the "Microsoft Sentinel incident" operation, which starts execution when an incident generated by the analytic rule "D4IoT - Malware Identified in Industrial Network" appears. This provides direct integration between the SIEM incident and the automation.

After triggering, the playbook parses the incident data. An important architectural decision is that the analytic rule groups multiple malware alerts by source system (the potentially compromised device), so that if multiple detections occur on the same machine, only one incident is created rather than many. Therefore, Logic Apps uses a loop operator "For each alert," iterating through the alerts within the incident. Data in "Custom Details" is processed using "Parse JSON" with a predefined schema to extract fields correctly, enabling dynamic values (alert type, source address, sensor, location, etc.) to be used in subsequent steps.

The final part of Logic Apps is "Send an email," which sends a high-importance message. The email body combines static text and dynamic variables, so the recipient gets the minimal but sufficient context to start an investigation: detection type (e.g., "Malware Test File Detected" or "Connection Attempt to Known Malicious IP"), the affected system, physical location, and other relevant attributes. This keeps notifications short, response-oriented, and operationally usable.

1.5.3.3 Sentinel Automation Rule

The third component is a Sentinel Automation rule acting as the "link" between incident generation and the Logic Apps playbook. The rule is created in the "Automation" section and configured to trigger "When incident is created". To ensure the playbook runs only for relevant incidents, conditions are applied: "Analytic rule name" is restricted to "D4IoT - Malware Identified in Industrial Network". This prevents unnecessary notifications for other incident types and keeps focus on Industrial malware detections.

After defining the conditions, actions are configured. The primary action is "Run playbook", which starts the Logic Apps email notification workflow. After the email is sent, the automation adds a tag to the incident - "Email Sent". This has two practical benefits: (1) it allows quick separation of incidents already notified from those that are not, and (2) it supports traceability and operational reporting.

In summary, the implementation builds an end-to-end chain "detection-incident-automation-notification", where D4IoT provides passive OT-aware detection, Sentinel provides SIEM correlation and the incident model, and Logic Apps provides automated communication to operational teams. This reduces notification time, provides context for rapid triage, and creates a practical basis for more effective response to malware activity in industrial networks.

1.5.4 Testing the Technical Solution

To verify and demonstrate the functionality of the proposed solution, testing was performed in two phases, including use of the EICAR test file. In the first phase, in a non-production environment

of an industrial enterprise (pre-commissioning installation), the EICAR file was downloaded to a workstation whose network traffic was monitored by D4IoT. The activity was detected and D4IoT generated the alert “Malware Test File Detected - EICAR AV Success”. The alert was forwarded to Azure D4IoT, ingested into Microsoft Sentinel, and triggered the notification mechanism. In Sentinel, key attributes are visualized such as time, alert name, involved devices, sensor, and location.

After ingestion via the data connector and the analytic rule, within approximately two minutes an incident “D4IoT - Malware Identified in Industrial Network” with high priority was generated. This automatically triggered the automation rule and Azure Logic Apps, resulting in an email notification to predefined recipients. The email contained minimal but operationally sufficient context: alert type, affected system (192.168.10.1), and physical location (Sofia). In addition, the incident was automatically tagged “Email Sent”, which helps distinguish processed cases and supports shift analyst operations. The experimental results are presented in Figures 1.30, 1.31, and 1.32.

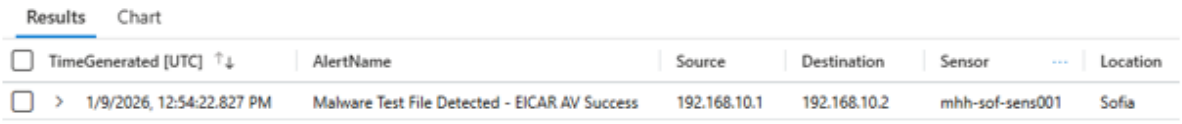


Fig 1.30 “Malware Test File Detected - EICAR AV Success” Sentinel alert

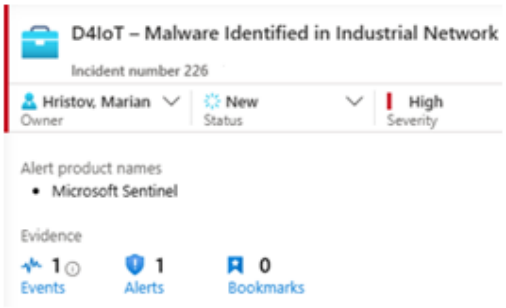


Fig 1.31 “D4IoT – Malware Identified in Industrial Network” incident

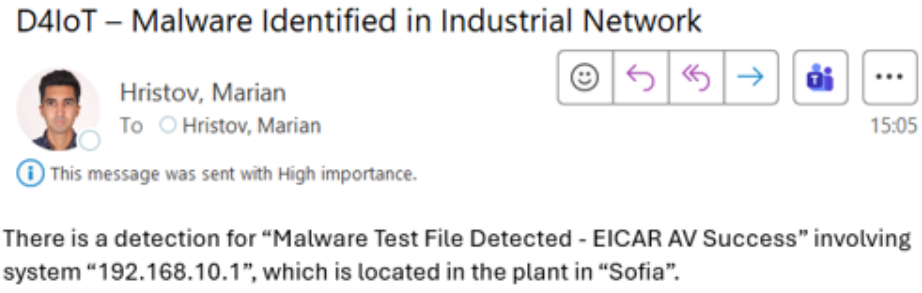


Fig 1.32 “D4IoT – Malware Identified in Industrial Network” email

After confirming the solution’s functionality, its scope was expanded to cover the entire industrial network, with all D4IoT sensors sending alerts to Azure D4IoT and Sentinel, and Industrial malware analytics alerts triggering the notification mechanism. The second experiment validated effectiveness using a realistic compromise indicator - an attempted communication with a malicious system. The workstation (192.168.10.1) attempted to reach IP 111.15.15.66. The

communication was blocked by a firewall, but the emitted packets caused the D4IoT sensor to generate the alert "Connection Attempt to Known Malicious IP". Almost immediately, the Sentinel analytic rule generated a second high-priority incident, and automation sent an email (alert type, source, location) and applied the "Email Sent" tag. The experimental results are presented in Figures 1.34, 1.35, and 1.36.

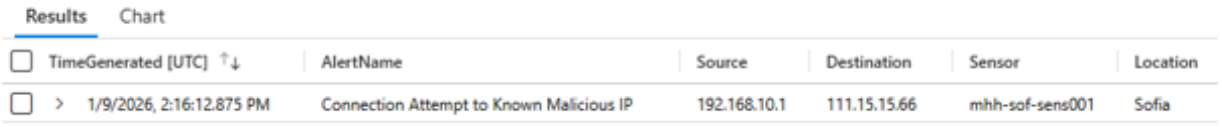


Fig 1.34 "Connection Attempt to Known Malicious IP" alert

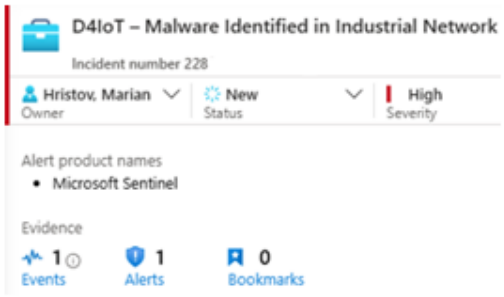


Fig 1.35 Second "D4IoT – Malware Identified in Industrial Network" incident

Fig 1.36 "Connection Attempt to Known Malicious IP" email

The received notification serves as a starting point for structured initial triage. First, the affected internal system can be checked in the CMDB to determine purpose and criticality; if non-critical, isolation/quarantine is recommended to limit spread or exfiltration without creating additional operational risk. Next, the destination should be analyzed through Threat Intelligence and OSINT. Public sources (e.g., GreyNoise, Cisco Talos, AbuseIPDB) classify 111.15.15.66 as malicious; according to GreyNoise, activity includes scanning and SSH attack attempts (port 22), which confirms detection validity and supports prioritization. Overall, testing shows the solution provides a reliable chain "detection-incident-notification", delivers minimal but sufficient SOC context, and enables fast, evidence-based response decisions in an industrial environment.

1.5.5 Conclusion

The issues discussed in Chapter 1 show that modern threats against OT/ICS organizations are not limited to data theft (intellectual property) or financial fraud but may lead to real operational interruptions and physical consequences that endanger human life and safety. Recent examples

such as Stuxnet, Industroyer, and especially TRITON demonstrate the evolution of cyberattacks targeting critical processes and safety systems, highlighting the need for rapid detection and coordinated response to minimize impact. In this context, a key challenge for many organizations is not the absence of detection mechanisms, but the lack of a reliable and sufficiently fast notification method that links technical detection to real actions by the response team (SOC).

The proposed solution addresses this gap by building a real-time notification mechanism for malware detections in industrial networks, initially detected by D4IoT and enhanced through integration with Microsoft Sentinel and Azure Logic Apps automation. The combination of D4IoT's passive, agentless network monitoring and Sentinel's SIEM capabilities enables transformation of alerts into structured, operationally useful notifications that reach the security center almost immediately. A key added value is context enrichment via KQL modifications - extracting SensorId and defining Location - making the email both concise and response-focused, including detection type, logical address of the affected system, and physical location.

Testing - first in a non-production and later in a real production environment - demonstrated practical effectiveness: when suspicious actions are detected (e.g., an outbound connection attempt to a known malicious IP), automation generates a timely email to SOC, reducing response time. This is critical in OT/ICS, where delays increase the likelihood of malware propagation, data exfiltration, and/or escalation into actions affecting industrial process continuity.

An important aspect is the described basic triage method, which turns the notification into the starting point for structured analysis: an initial CMDB check to determine system purpose and criticality, followed by OSINT reputation checks of the destination. This sequence supports decisions such as isolation/quarantine when operationally permissible - while ensuring actions do not introduce additional process risk - and supports prioritization without relying on incomplete context.

Through this, the author addresses the following predefined objectives and tasks:

1. Implementation of the proposed technical solution in test and real industrial environments;
2. Evaluation of effectiveness through practical experiments, test approaches, and real operational functionality;
3. Development of an OT risk assessment methodology based on the solution and established practices and standards.

In conclusion, the proposed solution demonstrates that establishing a reliable link between detection and response is a key factor for improving OT/ICS cyber resilience. By leveraging existing capabilities of D4IoT and Sentinel and adding targeted enrichment and automation, the approach provides fast, contextual, and actionable notifications for SOC, reducing time to detect and respond and supporting impact limitation for malware incidents - if not their full prevention.

1.5.6 Contributions to Chapter 1

The main contributions from the research and analysis in Chapter 1 can be summarized as follows:

- Derivation of a "real-time notification pipeline" method for OT/ICS detections - D4IoT-SIEM-automation-SOC - defining a minimal attribute set for an operationally valid notification: alert type, physical location, and source system;

- Formulation of an initial OT/ICS alert analysis model combining CMDB context (system purpose and criticality) and OSINT destination reputation as a minimal, reproducible set of steps for timely initial security risk assessment;
- Creation and successful implementation of an Azure Logic Apps–based automation for immediate notification on D4IoT malware detections in industrial environments integrated with Sentinel, including email delivery to SOC;
- Alert/context enrichment via KQL improvements, including extraction and presentation of SensorId (D4IoT sensor name) and "case mapping" between SensorId and the likely physical location of the potentially compromised source system, turning raw alerts into context-rich, response-oriented signals;
- Demonstration of a working production test scenario involving detection of suspicious activity and automatic email notification with key parameters enabling immediate investigation and potential response, complemented by a practical response "playbook" (CMDB check for the internal system and OSINT check for the external destination), with an isolation recommendation when permissible for the production process.

CHAPTER 2. INTEGRATION AND CENTRALIZED COLLECTION OF OT/IOT EVENTS IN A CLOUD ANALYTICS PLATFORM

2.1 Logs - Nature and Purpose

Logs are structured or unstructured records of events and actions in systems, applications, and network devices, and they serve as a primary source of evidence for monitoring, diagnostics, and incident investigation. In an OT/ICS context, logs have additional value because they support traceability of process deviations, asset changes, and communication anomalies, while at the same time they must be collected and used without compromising process continuity. Due to the heterogeneity of sources and formats, logs must be normalized and enriched with context in order to be operationally useful for security teams.

2.2 Systems for Monitoring and Storing Events

Event monitoring and storage systems (e.g., SIEM) provide centralized collection, storage, search, correlation, and visualization of logs and alerts from multiple sources, creating a "single pane of glass" for monitoring and response. In industrial networks, the effectiveness of such systems depends on their ability to integrate OT-aware telemetry (passive traffic analysis, alerts from OT sensors), support real-time rules and automations, and provide prioritization context (location, system criticality, and/or destination reputation). In this way, SIEM becomes a key tool for reducing time to detect and respond, and for ensuring evidentiary soundness during investigations.

2.3 Implementation of Health/Operational Monitoring for Microsoft Defender for IoT

Health/operational monitoring of a D4IoT implementation is a critical component of industrial cyber resilience. In OT/ICS, loss of visibility is not merely a technical issue but a direct risk: missing telemetry can leave the environment unprotected precisely during an attack, potentially leading to process disruption, financial losses, and even endangering human safety. This section presents an engineering solution for operational monitoring of D4IoT that addresses the question, "Who monitors the monitoring system?" through a multi-component approach with independent mechanisms and different notification channels.

2.3.2 Preparation for Deploying the Technical Solution

The engineering solution requires three prerequisites: (1) a cloud-connected D4IoT sensor, (2) integration with Microsoft Sentinel, and (3) an on-premises Zabbix server for SNMP monitoring.

In fully air-gapped environments without cloud connectivity, the solution can be reduced to the local component (Zabbix/SNMP), which remains applicable and independent of Azure. To achieve full functionality, the D4IoT sensor must be configured as Cloud connected during onboarding. This enables telemetry to the Azure portal and requires outbound access to specific Azure secure endpoints over port 443 (TLS), typically implemented through firewall rules allowing bidirectional traffic as required by Microsoft documentation. Next, Sentinel integration is achieved by enabling the Microsoft Defender for IoT data connector, which ingests D4IoT alerts into the SecurityAlert table. This is essential for the SIEM-based monitoring component, because the rules rely on the presence of D4IoT alerts in the Sentinel workspace.

Finally, integrating a Zabbix server requires SNMP MIB monitoring configuration - specifying the Zabbix IP and creating an SNMPv3 user with a password and secret key. This allows Zabbix to monitor the D4IoT sensor's management interface and signal unavailability.

2.3.3 Implementation of the Technical Solution

The solution is designed as three independent components operating in parallel and monitoring different aspects of the D4IoT deployment. This multi-layer approach is deliberately chosen to avoid reliance on a single indicator and minimize the risk of a single point of failure. In addition, the components use different notification channels (Sentinel incident, Teams message, and email), improving reliability and ensuring visibility even under partial infrastructure degradation.

Component 1: Sentinel analytics rules for missing D4IoT alerts

Two Sentinel rules monitor for lack of D4IoT alerts in SecurityAlert. The logic is practical: if D4IoT suddenly stops producing alerts, it may indicate a problem in the D4IoT-Azure D4IoT-Sentinel integration or a loss of telemetry. In OT/ICS, "absence of events" can be more dangerous than the presence of alerts because it may conceal real attacks.

- Rule 1: "D4IoT – Operational (No alerts in the last 30 minutes)" detects a complete absence of D4IoT alerts in the last 30 minutes by: (1) filtering SecurityAlert by ProviderName == "IoTSecurity"; (2) using max() over TimeGenerated to find the most recent alert; (3) comparing it to ago(30m) and generating an incident if it is older.
- Rule 2: "D4IoT – Operational (No alerts for a sensor in the last 12 hours)" detects sensors that have not produced any alerts in the last 12 hours by: (1) defining a list of expected sensors (datatable); (2) building a distinct list of sensors that produced at least one alert in the last 12 hours; (3) using rightanti join to identify sensors present in the expected list but missing from the observed list. This can indicate issues with a specific site, SPAN configuration, ingestion, or the sensor itself. It should be noted that in some environments (especially with strong tuning or low activity) it is possible to have no alerts for 12 hours, which can lead to false positives - therefore, the rule must be interpreted in context.

Component 2: Azure Logic Apps + Azure Resource Graph monitoring of cloud connectivity

This component addresses a different failure mode: the sensor is locally active but has lost cloud connectivity to Azure. In the cloud-connected model, alerts must reach Azure D4IoT/Sentinel to be processed by SOC. If connectivity drops, local monitoring may continue but centralized visibility is lost.

The mechanism is implemented via Logic Apps that runs periodically (e.g., every 10 minutes) and sends an HTTP POST query to Azure Resource Graph (ARG). ARG returns sensor properties, including connectivityTime (AzureLastConnectivityTime). A sensor is marked "Disconnected" if the last connectivity is older than a defined threshold.

An anti-flood logic is implemented: first a Teams notification is sent when a lower threshold is exceeded (e.g., 30–40 minutes), then after a "Wait 15 minutes" step a second, stricter check is

performed (e.g., 45+ minutes). If disconnection persists (but is still below a defined cap such as 56 minutes), a high-priority Sentinel incident is created. This two-stage logic aims to:

1. provide early warning (Teams);
2. escalate (Sentinel incident) for prolonged disconnection;
3. avoid notification flooding for long outages.

Teams is intentionally used first as a fast but non-SLA channel; Sentinel is used for tracked escalation and management.

Component 3: Local SNMP/Zabbix monitoring of the management interface

The third component is independent of Azure and Sentinel and is especially important for air-gapped or Internet-restricted environments. Its goal is to detect OS-level or management-interface issues - for example, when the sensor stops responding to ping on its management IP. Practical observations show that this is often associated with OS/container issues and may lead to partial or complete loss of functionality.

Zabbix is configured to send ICMP Echo Requests to the management interface every 2 minutes. If three consecutive replies are missed (6 minutes total), Zabbix sends an email notification to the OT security team. The strengths of this component are:

1. the shortest detection time for a “hard” failure (6 minutes);
2. independence from cloud connectivity and SIEM ingestion.

Overall, the operational monitoring implementation provides a multi-layer system covering:

1. ingestion/integration problems (missing alerts);
2. cloud connectivity problems (ARG/Logic Apps);
3. local management/OS failures (Zabbix).

The combination of independent indicators and notification channels minimizes the risk of unnoticed loss of visibility and provides a practical foundation for timely response and troubleshooting in OT/ICS.

2.4 Testing the Technical Solution

To test and prove the functionality of the proposed technical solution for operational monitoring of D4IoT, a series of tests was conducted, aimed at observing and recording the behavior of each of the three components of the overall system. The simulations were planned to test one of the most critical practical scenarios - an “unavailable/failed D4IoT sensor”, since this condition can lead to loss of telemetry and “blind spots” in industrial visibility. The time intervals in the tests were aligned with the logic and thresholds of the individual detection mechanisms (Zabbix/SNMP, Azure Logic Apps/ARG, and Sentinel analytic rules), so that both early notification and subsequent escalation in case of a prolonged fault could be validated.

For this purpose, a test D4IoT sensor was shut down in a controlled manner using the “system shutdown” command, executed in the Command-line interface (CLI) from an “admin” account. This guarantees that the “off” state is the result of real sensor unavailability rather than intermediate network fluctuations. The performed experiments, expected results, and observed results can be summarized in the following logic:

(1) Zabbix/SNMP email notification (fastest component)

The first expected action is the triggering of local SNMP monitoring implemented through Zabbix, which checks the availability of the D4IoT management interface. Since Zabbix is configured to

send ICMP ping requests every two minutes and to alert after three consecutively missed replies, the expected outcome is an email notification approximately 6 minutes after the sensor is fully stopped. The observed result confirms this behavior - the email was received six minutes after shutdown. The practical value of this component is that it provides the earliest signaling for a "hard" sensor failure, regardless of cloud connectivity and SIEM integration.

In a real failure detected in this way, remediation often requires physical access to the sensor or remote access to the hardware management interface (e.g., iLO/iDRAC). This shows that the Zabbix component is suitable for early detection of operating system/hardware-level issues and serves as a primary indicator that the monitoring infrastructure has degraded

(2) Teams message upon cloud connectivity degradation (early warning)

The next expected stage is activation of the Logic Apps mechanism, which monitors the last cloud connectivity of the D4IoT sensor to Azure via ARG. In the proposed logic, the first action is a Teams notification when a threshold is reached (30 minutes) since the last successful communication. The observed result confirms the expectation - 30 minutes after the last connection between D4IoT and Azure, a Teams message was received.

This notification has a warning character: at this stage, the issue may still be temporary (e.g., short Internet/route loss, restart of a firewall or network device). For this reason, Logic Apps includes an additional "wait" step (15-minute delay) before escalation. The Teams channel was selected for early notification because it is fast and operational, but it is not an SLA-oriented mechanism, which logically leads to the next step - creation of a Sentinel incident in case of more prolonged disconnection.

(3) Sentinel incident for complete absence of D4IoT alerts (integration indicator)

In parallel with the Teams warning, a Sentinel incident was also observed, generated by the analytic rule "D4IoT – Operational (No alerts in the last 30 minutes)," which monitors for a complete absence of D4IoT alerts in the SecurityAlert table during the last 30 minutes. In the test scenario, where there is only one sensor, the lack of telemetry naturally leads to triggering the incident. In a real production environment with multiple sensors, this rule would be more informative for global issues - such as an outage in Azure D4IoT, a problem with the data connector, or general degradation in Sentinel.

This result confirms that the first component of the solution (Sentinel rules) works correctly as an indicator of telemetry interruption at the SIEM level.

(4) Sentinel incident at 45 minutes without Azure connectivity (escalation)

After the 15-minute wait, Logic Apps performs a second check via ARG and, upon reaching a higher threshold (45 minutes since the last communication), generates a Sentinel incident. The observed result matches the expected behavior - 15 minutes after the Teams message, a Sentinel incident was received indicating that the sensor had not communicated with Azure for 45 minutes.

This step is interpreted as a need for real response, since the likelihood that the issue is temporary decreases and the risk of losing centralized visibility becomes significant. A good starting point for troubleshooting in such a case is verifying the sensor's Internet/route connectivity and executing diagnostic CLI commands (e.g., connectivity checks) to confirm whether the problem lies in the local network, the firewall, or the sensor configuration.

(5) Sentinel incident for no alerts from a specific sensor for 12 hours (long-term indicator)

The last validated element is the analytic rule "D4IoT – Operational (No alerts for a sensor in the last 12 hours)", which generates an incident when no alerts from a specific sensor are observed for 12 hours. The observed result confirms the expected behavior - at least 12 hours after the sensor was fully stopped, the corresponding Sentinel incident appeared.

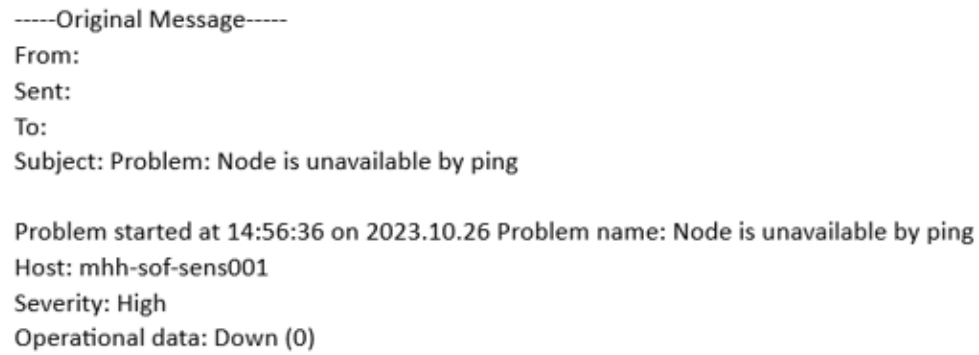
It is important to emphasize that in real environments it is possible for the rule to generate false positives if there are genuinely no alerts during a given period (e.g., a small industrial environment with excellent tuning). Therefore, the rule has value as part of the multi-layer system, but its results should be interpreted in context and confirmed with additional checks.

Practical guidance for verification and initial troubleshooting

As a starting point to determine whether telemetry has been restored, a good practice is to generate a test D4IoT alert (for example, using the built-in system health checks in the sensor interface). Successful generation and subsequent appearance of the alert in Azure/Sentinel is an indication that the D4IoT-Azure D4IoT-Sentinel chain is functioning properly again.

Summary of the test results

The conducted experiments confirm that the proposed technical solution operates according to design and provides staged, reliable, multi-channel notification for the scenario "unavailable D4IoT sensor." The earliest indicator is Zabbix (6 min), followed by a Teams warning (30 min), and subsequent escalation via Sentinel incidents (30 min for lack of alerts, 45 min for loss of Azure connectivity, 12 hours for lack of alerts from a specific sensor). This sequence demonstrates the key value of the multi-layer approach: even under partial infrastructure degradation, the system retains the ability to notify through independent channels and indicators, while simultaneously supporting diagnosis of the likely cause (OS/management failure, cloud connectivity loss, or integration issue). This validates the practical applicability of operational monitoring as a necessary layer for ensuring visibility and resilience in OT/ICS monitoring deployments. The experimental results are presented in Figures 2.24, 2.25, 2.26, 2.27, and 2.28.


-----Original Message-----
From:
Sent:
To:
Subject: Problem: Node is unavailable by ping

Problem started at 14:56:36 on 2023.10.26 Problem name: Node is unavailable by ping
Host: mhh-sof-sens001
Severity: High
Operational data: Down (0)

Fig 2.24 Zabbix email after 3 unsuccessful pings

Fig 2.25 Teams notification that the sensor is disconnected

Title	Created time	Owner
D4IoT - Operational (No alerts in the last 30 minutes)	10/26/23, 3:10 PM	Hristov, Marian

Fig 2.26 Sentinel incident for the lack of alerts in the last 30 minutes

Title	Created time	Owner	Description
mhh-sof-sens001 is Disconnected	10/26/23, 3:25 PM	Hristov, Marian	The last connectivity of mhh-sof-sens001 to Azure was 10/26/23, 2:40 PM

Fig 2.27 Sentinel incidents for reaching 45 minutes of the last connection

Title	Created time	Owner
D4IoT - Operational (No alerts for a sensor in the last 12 hours)	10/27/23, 12:00 PM	Hristov, Marian

Fig 2.28 Sentinel incident for the lack of alerts in the last 12 hours

2.5 Conclusion

Chapter 2 demonstrates that as digitization increases and the IT–OT boundary continues to blur, deploying visibility and detection mechanisms in industrial networks is necessary but not sufficient on its own. When an OT/ICS organization relies on passive, agentless monitoring such as D4IoT, the critical question arises: “Who and how monitors the monitoring system?”. Limited or lost visibility due to sensor, integration, or infrastructure problems can leave the environment unprotected precisely during a cyberattack, creating risks of operational disruption, financial damage, and potential safety impact.

The proposed health/operational monitoring solution addresses this risk through independent mechanisms for detecting visibility degradation and providing early notification. Its key practical value is that it does not rely on a single indicator; instead, it uses three parallel components monitoring different aspects of the D4IoT deployment and different notification channels. Sentinel rules detect missing alerts globally and per sensor, reducing the chance of unnoticed telemetry loss; Logic Apps/ARG monitors cloud connectivity with anti-flood logic; and SNMP/Zabbix provides a fully local layer for Internet-disconnected environments. Testing via controlled sensor shutdown confirms the effectiveness of the multi-layer design: Zabbix triggers fastest (6 minutes), followed by Teams notification at the cloud threshold, and finally Sentinel incidents for the defined conditions.

In summary, the key takeaway is that in an industrial environment it is not enough to deploy

detection; it is also necessary to build reliable operational monitoring that continuously validates visibility and alerts when key components fail.

2.6 Contributions to Chapter 2

The contributions of Chapter 2 can be summarized as follows:

- Formalization of the "monitoring the monitoring system" concept in an OT/ICS context by defining measurable operational indicators (missing alerts, cloud connectivity loss, management-interface unavailability) as symptoms of potential monitoring degradation;
- Application of a multi-layer reliability model based on independent notification mechanisms (SIEM incident, Teams message, email), reducing the likelihood of a single point of failure;
- Design and successful deployment of two Sentinel analytics rules detecting "no D4IoT alerts globally (30 minutes)" and "no D4IoT alerts from a specific sensor (12 hours)" using KQL time and correlation operators;
- Design and successful deployment of Azure Logic Apps with ARG queries for monitoring "LastConnectivityTime" and automatic Teams notification upon cloud connectivity loss, including anti-flood logic;
- Design and successful deployment of a fully local monitoring layer (Zabbix server, SNMP and ping) for management-interface monitoring and email notification on unavailability, primarily applicable to air-gapped environments;
- Derivation of a validated testing method (controlled shutdown) and a detailed troubleshooting approach, including direct/local access, Internet connectivity checks, and/or verification of the Azure D4IoT–Sentinel integration.

CHAPTER 3. AUTOMATED NOTIFICATION MECHANISMS AND OPERATIONAL MONITORING IN INDUSTRIAL NETWORKS

3.1 Detecting Attacks Against Industrial Networks Aimed at Downgrading Firmware Versions

Firmware downgrade attacks are characterized by abusing legitimate update processes and thus concealing malicious actions as routine maintenance. In an OT/ICS context, this is critical because a regression to an older version can intentionally reintroduce known vulnerabilities and create conditions for subsequent compromise of control systems, loss of manageability, and safety risk. Therefore, effective detection requires OT-aware monitoring and analytics that distinguish not merely a "Firmware Change," but the direction of the change and its associated risk.

3.3 Implementation of the Technical Solution

The designed engineering solution for detecting Firmware Downgrade attacks is implemented as a combination of: (1) a Microsoft Sentinel analytic rule that generates an incident only for regressive firmware changes, and (2) an automated weekly report for all remaining "Firmware Change Detected" alerts, providing governance and control over legitimate updates. The data source for the solution is the "Firmware Change Detected" alert generated by D4IoT and ingested into Sentinel via the "Microsoft Defender for IoT" data connector. The main goal is to overcome the limitation of standard platforms that signal a firmware change but do not distinguish whether it is an upgrade or a downgrade, thereby enabling selective escalation of the highest-risk events.

3.3.1 Analytic Rule for Detecting Firmware Downgrades

The first component is a Sentinel analytic rule that implements version-aware logic: it extracts the previous and current firmware versions from the alert and compares them by components to determine the direction of change. The rule is created in the “Analytics” menu, where a name and description are provided in the “General” section.

In the next step, a KQL query is defined over the SecurityAlert table, limited to D4IoT alerts through ProviderName == "IoTSecurity", and filtered specifically for the “Firmware Change Detected” alert. Using the extend operator, variables are initialized and the required parameters are extracted from JSON structures via tostring() and parse_json(). The “UpdatedFrom” and “UpdatedTo” versions are parsed and the numeric values are extracted using extract() and the regex: @"Software Revision:\s+([^;]+);".

The versions are then split into components (Major, Minor, Patch, Build) using split(), and each component is converted into a numeric type using toint(). In this way, separate variables are created for the old and new versions, enabling correct “digit-by-digit” comparison.

The comparison itself is implemented using case(), where the result classifies the change as “OldVersion is newer” (downgrade), “NewVersion is newer” (upgrade), or “Versions are equal”. An incident is generated only when the old version is higher than the new one, i.e., in the case of regression. The selected information (device, versions, time, sensor, etc.) is output using project. After defining the KQL logic, incident creation on match is enabled in “Incident settings”, and the rule is saved (“Save”) in “Review + create”.

The choice of a KQL + Regex approach is intentional: the logic executes in Sentinel as a single atomic operation, ensuring speed (detection and incident creation within approximately one minute) and eliminating external dependencies. When analyzing alternatives (e.g., Azure Functions or Python scripts), it was determined that they introduce additional latency and operational complexity (code maintenance, versioning, scaling) without providing meaningful functional benefit compared to Sentinel’s built-in capabilities (parse_json(), extract(), split(), case()). The regex aligns with the way D4IoT annotates firmware versions - most commonly as a 3- or 4-component string (e.g., “2.6.6.6”).

Additionally, during design a comparison with competing solutions (Nozomi Guardian and Claroty CTD) was performed, showing similar baseline capability: they report a firmware change, but by default do not determine whether the change is progressive or regressive. This confirms the practical added value of version-aware logic as a mechanism for breaking the “symmetry” between legitimate and malicious firmware flows.

3.3.2 Weekly Report for All Firmware Changes

The second component is a supporting automation that manages the remaining “Firmware Change Detected” alerts - i.e., legitimate upgrades - without creating operational noise in the SOC. The goal is not to ignore these events entirely, but to observe them at an appropriate frequency and with context. The weekly report has two main use cases:

1. Detecting anomalies (e.g., a 5-10 times increase in the number of updated devices, which may indicate a coordinated shift to a specific version with known vulnerabilities);
2. Supporting compliance/regulatory requirements that demand an asset register and a change log.

The automation is implemented using Azure Logic Apps, which once per week executes a KQL query in Sentinel for "Firmware Change Detected" alerts from the last 7 days, structures the results into a CSV file, and sends an email to predefined recipients (e.g., a team responsible for maintaining industrial equipment). Logic Apps is scheduled (e.g., every Friday at 12:00), runs "Run query and list results," checks whether there are results (IF condition), creates a CSV (Create CSV table), and sends the report as an attachment via "Send an email." In this way, legitimate changes are handled in a reporting workflow, while the SOC remains focused on high-risk downgrade incidents.

3.4 Testing the Functionality of the Technical Solution

To test and prove the functionality of the proposed technical solution for detecting Firmware Downgrade attacks, testing was conducted in two phases based on intentionally downgrading the firmware of an Atlas Copco Power Focus 6000 controller. Initially, the mechanism was deployed and validated in a dedicated non-production environment of an industrial enterprise (a pre-commissioning installation). After proving functionality and practical value, the deployment scope was expanded to cover the entire industrial network of the manufacturing enterprise, including enabling a weekly report for all firmware changes.

The first phase of the tests was conducted in a controlled test setup consisting of the following components:

- Atlas Copco Power Focus 6000 - a central tightening controller typically connected to one or more screwing machines;
- Moxa AWK-3131A - an industrial wireless access point (IEEE 802.11 a/b/g/n) used to connect the controller and the panel PC to the wireless network;
- Beckhoff CP2916-0000 - an industrial panel PC (processor + HMI) with Windows 10 IoT used for monitoring and controlling processes.

The network topology of the test environment was designed to match the enterprise's real industrial network as closely as possible, to obtain objective results and facilitate subsequent deployment. Logical addressing was configured as follows: Moxa - 10.10.10.10/28, Beckhoff - 10.10.10.1/28, Atlas Copco - 10.10.10.2/28. It should be noted that in real production environments firmware changes are typically deployed from a centralized system (server) rather than from a local device as in the test setup; however, this does not affect the validity of the detection logic being evaluated.

The tests can be summarized in five key scenarios:

1. device discovery and identification of the current firmware version;
2. firmware downgrade and an expected "Firmware Change Detected" alert;
3. generation of a Sentinel incident "D4IoT – Firmware Downgrade Detected";
4. firmware upgrade (restoration) and generation of a second "Firmware Change Detected" alert;
5. incident suppression for the upgrade, i.e., the second alert should not lead to escalation.

The observed results confirm the expectations: D4IoT identified the initial version 2.6.6.6, then generated a "Firmware Change Detected" alert when downgraded to 2.5.7.2. Approximately one minute after the alert, the Sentinel analytic rule created an incident "D4IoT – Firmware Downgrade Detected", based on the result of the version-aware comparison.

The second phase of the first testing stage included “learning” (Learn) the alert so that D4IoT updates its database with the current version and can detect subsequent changes. After restoring the controller to the original version 2.6.6.6, D4IoT generated a second “Firmware Change Detected” alert, however, unlike the downgrade event, it did not lead to a new Sentinel incident because the analytic rule correctly classified the change as an upgrade and suppressed escalation. This was confirmed by searching for incidents generated by the rule, which returned only one result - the initial downgrade.

With the completion of the first stage, the primary testing goal was achieved: it was proven that the solution generates an incident only for regressive changes, thereby reducing alert fatigue and focusing SOC attention on the highest-risk scenarios. The second stage expanded applicability in a production environment: the scope was extended to the entire industrial network, and the additional automation (Logic Apps) sends a weekly report containing all “Firmware Change Detected” alerts in CSV format by email to predefined recipients. Thus, legitimate firmware changes are managed in a reporting and contextual manner, while potential downgrade attacks are immediately escalated as incidents. The experimental results are presented in Table 3.1 and Figures 3.21, 3.22, 3.23, 3.24, 3.25, and 3.26.

Device	IP address	Subnet mask
Moxa AWK-3131A	10.10.10.10	255.255.255.240
Beckhoff CP2916-0000	10.10.10.1	255.255.255.240
Atlas Copco Power Focus 6000	10.10.10.2	255.255.255.240

Tab 3.1 IP addressing of the test bed

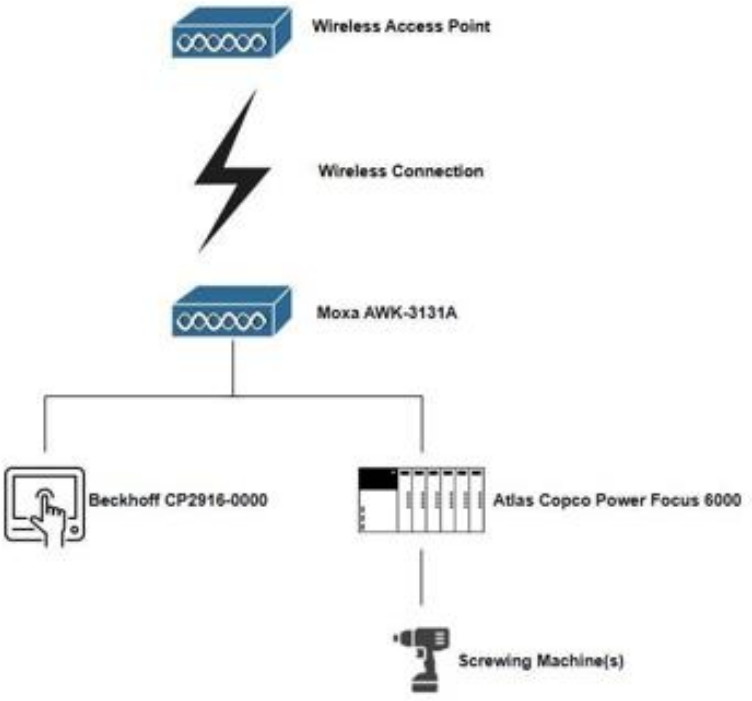


Fig 3.21 Test bed topology

TimeGenerated [UTC]	AlertName	Source	Destination	NewVersion	OldVersion	CompareResult
1/28/2025, 6:01:26.958 PM	Firmware Change Detected	10.10.10.1	10.10.10.2	2.5.7.2	2.6.6.6	OldVersion is newer

Fig 3.22 "Firmware Change Detected" alert

Title	CreatedTime	Owner
D4IoT - Firmware Downgrade Detected	1/28/2025, 6:02:31.042 PM	Hristov, Marian

Fig 3.23 "D4IoT - Firmware Downgrade Detected" incident

TimeGenerated [UTC]	AlertName	Source	Destination	NewVersion	OldVersion	CompareResult
1/28/2025, 7:16:08.119 PM	Firmware Change Detected	10.10.10.1	10.10.10.2	2.6.6.6	2.5.7.2	NewVersion is newer

Fig 3.24 "Firmware Change Detected"alert

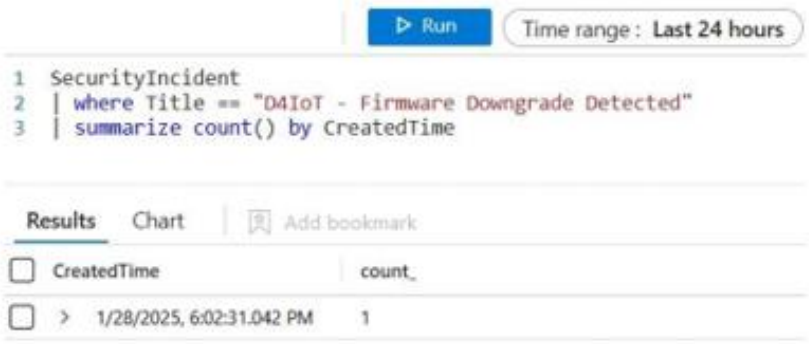


Fig 3.25 Exisiting "D4IoT - Firmware Downgrade Detected" incidents in the last 24 hours

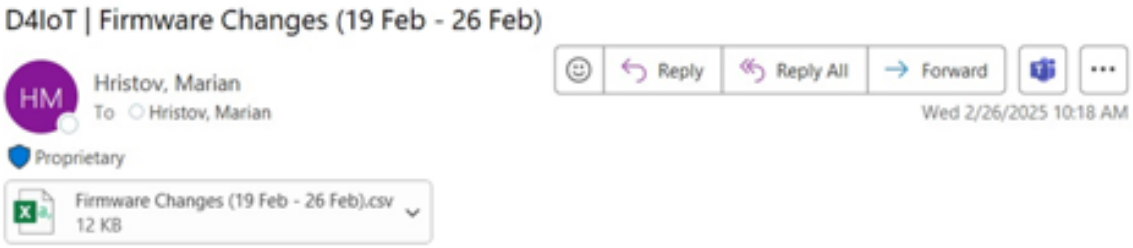


Fig 3.26 "Firmware Change Detected" weekly report

3.5 Conclusion

The information presented in Chapter 3 argues that in modern cyberattacks against OT/ICS organizations, the importance is increasing of threats that do not rely on classic attack techniques but instead abuse legitimate processes and functionalities in existing devices, systems, and automated industrial workflows. Firmware Downgrade attacks are a typical example: malicious actions are concealed by imitating the normal firmware update process (Firmware Upgrade),

creating symmetry between legitimate and illegitimate operations and complicating detection. In OT/ICS, this type of attack is particularly critical because restoring an older firmware version can intentionally introduce vulnerabilities that may later be used to compromise systems, reduce visibility and process controllability, and create potential safety risks.

The main contribution of the developed technical solution for detecting potential Firmware Downgrade attacks is that it breaks the functional symmetry between upgrade and downgrade processes by relying on behavioral profiling of OT/ICS devices and protocol analysis implemented through passive, agentless network monitoring (D4IoT) and Sentinel's correlation capabilities. Instead of relying only on a binary "Firmware Change" alert, the proposed solution introduces version-aware logic that compares the previous and current firmware versions and generates an incident only for regressions (Downgrade). This functionality addresses a systemic limitation observed in competing platforms - Nozomi and Claroty - which, like D4IoT, notify about changes by default but do not indicate whether the change is progressive or regressive.

Experimental results demonstrate that the approach works reliably both in a test and in a real production environment - D4IoT detects the change, and Sentinel generates a single incident for the Downgrade event within approximately one minute. At the same time, when restoring to a newer version (Upgrade), the designed solution correctly suppresses and prevents incident creation, reducing alert fatigue and focusing the security center's attention on the highest-risk events. This supports the achievement of two predefined objectives of the dissertation: implementation of the proposed solution in test and real industrial environments, and evaluation of effectiveness through practical experiments, test approaches, and real operational functionality.

In addition, the author presents an auxiliary (optional) feature that includes automated weekly reports for the remaining "Firmware Change Detected" alerts, i.e., upgraded devices and systems. In this way, upgrades are not ignored but are managed with appropriate frequency and context.

In conclusion, the research shows that passive, agentless network monitoring combined with SIEM-oriented, version-aware analytic logic can effectively distinguish legitimate firmware changes from malicious regressive attempts despite the high similarity between the two processes. The practical effect is reduced visibility gaps in OT/ICS, accelerated detection, and reduced risk of exploiting vulnerabilities reintroduced through older firmware versions.

3.6 Contributions to Chapter 3

The contributions of Chapter 3 can be summarized as follows:

- Introducing a conceptual model of "security symmetry" in the firmware lifecycle based on downgrade malicious activity disguised as a legitimate upgrade, and arguing that this symmetry must be dissected through behavioral profiling;
- Deriving a version-aware method for detecting firmware downgrades in a passive OT/ICS monitoring context, transforming the standard "Firmware change" alert from a generic signal into a specific indicator of regression and potential reintroduction of vulnerabilities in older firmware versions;
- Interpreting the problem through the MITRE ATT&CK for ICS lens (T0857) and selecting protocol analysis (DS0029) as a practical detection basis for environments without agents and without active scanning;
- Creating and successfully deploying a Sentinel analytic rule that parses the ExtendedProperties of the corresponding alert ("Firmware Change Detected"), extracts

versions via Regex, splits components, and performs digit-by-digit comparison to determine a potential firmware downgrade;

- Validating a test bed in a non-production environment with real industrial devices and corresponding scenarios - detect downgrade-incident and detect upgrade-incident suppression - leading to reduced SOC alert fatigue and better focus on high-risk events;
- Creating and successfully deploying an automation for weekly firmware change reports (Azure Logic Apps-CSV-email) to establish a management mechanism for detecting anomalous patterns (mass migration to a specific firmware version) and supporting regulatory compliance.

CHAPTER 4. Analytic Models for Detection and Investigation: Malware Indicators and Context Enrichment of Incidents

4.1 Event Monitoring and Analysis System – Splunk Enterprise

Splunk Enterprise is a platform for centralized collection, storage, search, and analysis of events (logs) from multiple heterogeneous sources, providing unified visibility - "single pane of glass" - for security teams. By indexing data, using diverse SPL queries, and leveraging alerting mechanisms, the system enables real-time monitoring and event correlation, as well as the creation of reports and visualizations that support investigation and response. In the context of this dissertation, Splunk is considered a practical SIEM component for monitoring Industrial and IoT environments, including as an alternative in cases where local infrastructures and/or limited cloud connectivity are used.

4.3 Configuration of Rules for Generating Alerts

Configuring rules for generating alerts is a key element of the proposed engineering solution, as it enables the transformation of logs and IDS/IPS events into operationally actionable signals for the security team. The approach is designed to provide real-time notification for high-risk events and periodic reporting for lower-risk indicators, supporting a balance between sensitivity and control over alert noise.

4.3.1 Generating an Alert for Every Successful Login to the SIEM Infrastructure

A best practice in monitoring management is to install the SIEM infrastructure on a separate machine dedicated solely to monitoring and event storage. In such a scenario, successful logins to the system are not frequently expected and are typically the result of planned configuration changes or troubleshooting activities. Therefore, every successful authentication - especially one that is atypical in terms of time, source, or account - should be treated as a potential indicator of malicious activity.

Within the solution, a real-time alert is created that triggers on every successful login attempt to the SIEM system. The alert uses two main criteria: (1) the logs must be present in the dedicated index "siem_local", and (2) EventCode = 4624, which corresponds to the Windows event "An account was successfully logged on". The configuration is implemented as a real-time, Per-Result alert so that each individual result matching the conditions triggers notification.

Two actions are added to the alert: recording the alert in the SIEM console and automatically sending an email notification to the security team. The email contains the key details, including a

CSV file with the event attributes and a link to the alert in Splunk. The practical value is that the team can respond immediately and verify who the account is, whether access is authorized, and what the reason for the login is.

4.3.2 Automated Report for Unsuccessful Login Attempts to the SIEM Infrastructure

Unsuccessful login attempts are also an important indicator, but unlike successful logins, they do not always require immediate response. The reasons are often trivial (wrong password, wrong username, or an expired account), so an appropriate strategy is periodic analysis rather than real-time alerting.

Therefore, an automated daily report is configured, covering the last 24 hours and extracting all unsuccessful login attempts based on: (1) index "siem_local" and (2) EventCode = 4625, corresponding to the Windows event "An account failed to log on". The report runs on a schedule (e.g., at 12:00) and provides an aggregated view of trends and anomalies (e.g., many failed attempts for a given account) without generating constant alerts and operational noise.

4.3.3 Generating an Alert for C2 Communication from an Internal Machine

Monitoring IDS/IPS events is necessary because modern attacks often involve compromising internal systems and attempting communication with Command-and-Control (C2) infrastructure. Although most IPS traffic is inbound and blocked at the perimeter, outbound traffic from the local network to the Internet is particularly critical, because it may indicate a compromised system and active malicious behavior.

In the solution, a real-time alert is configured to trigger when malicious outbound IDS/IPS traffic is detected (e.g., a Mirai.Botnet detection). The alert records the event in the SIEM console with high priority and sends an automatic email with a link to the alert and a CSV file containing the details. This provides SOC with immediate visibility into a potential compromise even when the firewall has blocked the communication, since the attempt itself is an indicator of infection and risk of participation in a DDoS network.

4.3.4 Generating an Alert for Intentional Log Deletion to Conceal Traces

In successful attacks, attempts to conceal traces by deleting logs are frequently observed. For this reason, the engineering solution includes an alert for clearing the Windows Security log, based on EventCode = 1102 ("The audit log was cleared"). The alert is configured as real-time, Per-Result, and uses index "siem_local" as the source.

When triggered, two actions are performed: recording the alert in the SIEM console with high priority and sending an automatic email to the security team with details and a link to the alert. This ensures immediate attention to a potential attempt to conceal malicious activity and supports rapid mitigation actions.

4.4 Testing the Functionality of the Proposed Engineering Solution

To verify the functionality of the proposed engineering solution, controlled tests were conducted to validate correct triggering of all configured alerts and reports, as well as the sending of automatic email notifications. Testing was organized by individual scenarios corresponding to the rules

described in the previous section, with the goal of confirming that the SIEM infrastructure provides timely alerting and sufficient context to begin an investigation.

4.4.1 Receiving an Alert for Every Successful Login to the SIEM Infrastructure

To test email notifications for successful logins, a successful login attempt was performed on the machine where Splunk Enterprise is installed. As a result, the alert based on EventCode 4624 in index "siem_local" was triggered, and an automatic email was sent to a predefined recipient. The notification included a brief description of the event, an attached CSV file with details, and a link to the specific alert in the SIEM. The practical value of this test is that it confirms the security team can receive a real-time signal for potentially unauthorized access and immediately verify which account was used, from where, and in what context.

4.4.2 Generating a Report Containing Unsuccessful Login Attempts to the SIEM Infrastructure

To validate the automated report for failed logins, the execution schedule was modified for testing purposes: instead of once per day, the report was triggered every hour and covered the last 30 minutes. Within the test window (06:30–07:30), the report found 13 unsuccessful login attempts based on EventCode 4625. A deeper review of a detailed log showed the specific parameters of the attempts, including the user account, the login type (e.g., "2" – interactive login on the machine itself), and the reason for failure ("Unknown user name or bad password"). The test confirms that the report is suitable for periodic tracking of trends and anomalies without creating unnecessary operational noise.

4.4.3 Receiving an Alert for C2 Communication from an Internal Machine

To test the IDS/IPS alert intended to detect malicious outbound communication (e.g., Mirai.Botnet), a controlled approach was used to reproduce the event without risk to the test environment. A specially prepared query was sent to Splunk that simulated the required event and triggered the configured action (sendemail). As a result, an email was received containing key alert details - source, destination, ports, the action taken (e.g., blocked/dropped), and other relevant information. Reviewing the alert in the SIEM confirmed that sufficient context was available for an initial assessment and response. The test also demonstrates the practical interpretation of such cases: even when outbound traffic is successfully blocked, the internal system remains potentially compromised and must be contained (quarantined) and remediated to prevent spread.

4.4.4 Receiving an Alert for Intentional Log Deletion to Conceal Traces

To validate the alert for clearing the Security log, the "Security" log was deliberately cleared via Event Viewer. This action generated Windows event 1102 ("The audit log was cleared"), which triggered the configured real-time alert in Splunk (index "siem_local", EventCode 1102). As a result, an automatic email was sent with CSV details and a link to the alert in the SIEM. Reviewing the alert provided the information needed for investigation, and the test confirms that the solution successfully detects actions characteristic of trace removal - typically part of the final phases of a cyberattack.

In summary, the tests confirm the correct operation of the implemented alerts and reports, as well as the reliability of automated notifications. The solution provides timely alerting for high-risk

events (successful logins, IDS/IPS indicators, log clearing) and controlled reporting for lower-risk indicators (failed logins), while delivering operationally actionable context for starting investigations and taking appropriate measures. Figures 4.15, 4.55, 4.56 and Table 4.1.

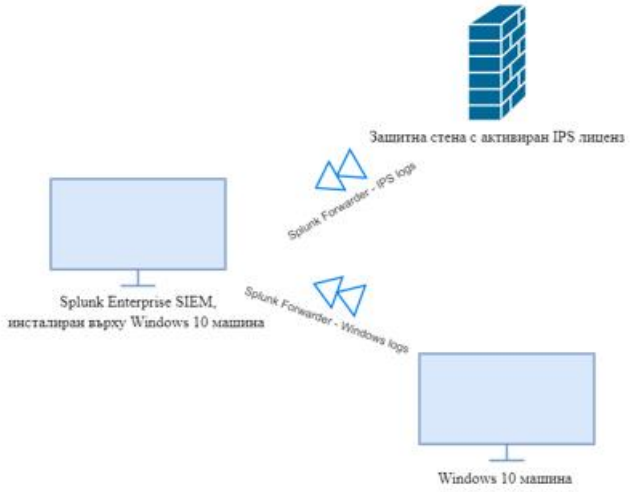
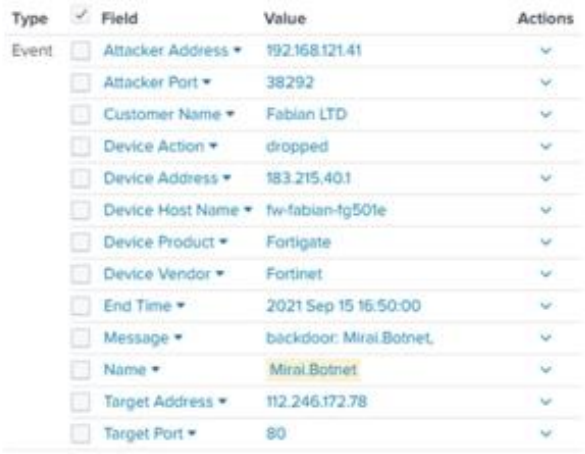


Fig 4.15 Test bed topology

Name	IP address	Description
DESKTOP-3GD6VLQ	192.168.0.106	Splunk SIEM
DESKTOP-5TG45DH	192.168.0.107	Splunk Forwarder

Tab 4.1 IP addressing of the test bed

Fig 4.55 IDS/IPS alert notification



Type	Field	Value	Actions
Event	Attacker Address	192.168.121.41	
	Attacker Port	38292	
	Customer Name	Fabian LTD	
	Device Action	dropped	
	Device Address	183.215.40.1	
	Device Host Name	fw-fabian-fg501e	
	Device Product	Fortigate	
	Device Vendor	Fortinet	
	End Time	2021 Sep 15 16:50:00	
	Message	backdoor: Mirai.Botnet,	
	Name	Mirai.Botnet	
	Target Address	112.246.172.78	
	Target Port	80	

Fig 4.56 IDS/IPS alert details

4.6 Conclusion

The review presented in Chapter 4 shows that OT/ICS cybersecurity can no longer be treated as an add-on to traditional IT security, but as an independent and critical field in which the leading priorities are process continuity and safety. With the development of the Fourth Industrial Revolution - presented in 1.1 – IT-OT integration and the growing level of remote connectivity increase the efficiency of industrial systems, but at the same time expand exposure to potential attacks and increase the likelihood of incidents with operational and physical effect.

The threat overview for the period 2022–2025 confirms sustained growth both in volume and in the maturity of malicious actors aiming to negatively impact industrial organizations. An exponential increase in ransomware incidents is observed (605 in 2022, 905 in 2023, and more than 3300 affected industrial organizations in 2025), leading to the conclusion that cybercrime remains the dominant risk to industry. At the same time, another noticeable change is observed: the emergence and advancement of ICS-specific malicious tools and tactics aimed at detailed understanding of - and impact on - the industrial process itself, not only targeting IT infrastructure.

The presented characteristics, and thus the resulting weaknesses in industrial environments, are to a large extent systemic and repeatable: limited visibility, insufficient segmentation, uncontrolled remote access, and shared IT–OT credentials. The visibility problem is especially critical because without OT-aware monitoring organizations can neither detect anomalies in time nor conduct reliable investigation during an incident. Dragos data confirms this - limited or entirely missing OT visibility is among the most common findings in real environments, and globally only a small portion of OT networks have adequate monitoring deployed.

In summary, Chapter 4 builds the necessary theoretical and analytical foundation for the practical part of the dissertation, while also fulfilling part of the research objectives and tasks, namely: identifying threats and main risks in OT/ICS; analyzing existing solutions to address them; studying technologies for monitoring and analyzing events in OT/ICS; and developing an OT/ICS monitoring system based on the results of the conducted in-depth research. Finally, Chapter 4 presents the first component of the developed engineering solution - Splunk Enterprise implementation for detecting DDoS attacks in IoT.

4.7 Contributions to Chapter 4

The contribution of Chapter 4 can be summarized as follows:

- Creation of a SIEM-oriented model for detecting different phases of a traditional cyberattack - Initial Access, Data Exfiltration, Trace Covering - implemented through correlation of operating system (Windows) and network (IPS) logs in a unified system;
- Definition of a measurable approach for real-time detection by classifying detections as real-time alerts or scheduled reports, and justifying the selection based on the operational value of the indicators;
- Conceptual integration of OSINT enrichment into the investigation process to increase the evidentiary value of detections based on the analytical methodology (confirmation of malicious context);
- Implementation of a working reference Splunk Enterprise architecture through separate indexes for local, remote, and network logs, demonstrating a typical production-grade onboarding approach;
- Development of a set of SPL-based rules for real-time alerts and periodic review reports:
    - Windows event 4624 – suspicious successful logins to the SIEM infrastructure;
    - IPS Mirai.Botnet log – potential data exfiltration and/or C2 communication;
    - Windows event 1102 – attempt to conceal traces of malicious activity;
    - Windows event 4625 – report for anomalies in failed login attempts to the SIEM infrastructure;
- Demonstration and validation through controlled tests - reproducing 4624, 1102, 4625, and IPS Mirai.Botnet - proving repeatability and applicability, and deriving a practical framework for reducing alert fatigue by distinguishing high-priority real-time alerts from low-urgency indicators suitable for reporting. This aims to optimize SOC workload.

CONTRIBUTIONS OF THE DISSERTATION

The topic of this dissertation is related to the research and development of a cloud-based system for monitoring and analyzing events in Industrial (OT/ICS) networks. Industrial environments have specific constraints and priorities - process continuity, safety, long equipment lifecycles, and limited ability to deploy agent-based solutions or perform active scanning. This necessitates the use of passive visibility approaches, centralized event correlation, and reliable mechanisms for automated response. The dissertation analyzes current threats and systemic weaknesses in OT/ICS and implements and validates concrete engineering mechanisms that improve the link between detection and response, enhance the reliability of monitoring, and support distinguishing legitimate from malicious activity.

The contributions of the dissertation can be summarized as follows:

5. A comprehensive analysis of OT/ICS threats and systemic weaknesses (visibility, segmentation, remote access, and identities) is presented, and based on it the architectural requirements and objectives for a cloud-based event monitoring and analysis system in industrial environments are formulated.
6. A real-time notification mechanism for malware detection in an industrial network is designed and implemented by integrating passive OT monitoring (Microsoft Defender for IoT), SIEM correlation (Microsoft Sentinel), and automation (Azure Logic Apps), with context enrichment (SensorId/Location) that transforms alerts into concise, operationally actionable SOC notifications.
7. A multi-component operational monitoring solution for a D4IoT deployment is developed and validated, including independent mechanisms for detecting degradation/loss of

visibility (Sentinel rules for missing alerts, Logic Apps/Azure Resource Graph monitoring of cloud connectivity with anti-flood logic, and local SNMP/Zabbix monitoring), applicable also in environments without Internet connectivity.

8. A version-aware firmware downgrade detection mechanism is developed that breaks the symmetry between legitimate and malicious firmware changes through passive monitoring (D4IoT) and SIEM analytics (Sentinel), generating incidents only for regressions while suppressing escalation for legitimate upgrades; additionally, an automation for weekly reports of all firmware changes is implemented to support governance and control.

LIST OF PUBLICATIONS RELATED TO THE DISSERTATION

[A1] V. Dimitrova, **M. Hristov**, K. Nikolova and M. Nenova, "Securing the Edge: A Comparative Analysis of Microsoft Defender for IoT and Nozomi Guardian," 2025 33rd National Conference with International Participation (TELECOM), Sofia, Bulgaria, 2025, pp. 1-4, doi: 10.1109/TELECOM66943.2025.11304075.

[A2] **Hristov, M**.; Nenova, M.; Dimitrova, V. Security Symmetry in Embedded Systems: Using Microsoft Defender for IoT to Detect Firmware Downgrade Attacks. Symmetry 2025, 17, 1061. https://doi.org/10.3390/sym17071061 . **Citations: 1**

[A3] **M. Hristov**, M. Nenova, K. Nikolova and V. Dimitrova, "Health Monitoring of Microsoft Defender for IoT Implementation," 2023 31st National Conference with International Participation (TELECOM), Sofia, Bulgaria, 2023, pp. 1-4, doi: 10.1109/TELECOM59629.2023.10409719. **Citations: 1**

[A4] B. Stoytcheva, M. Nenova, **M. Hristov**, Z. V. Djarvis and G. Balabanov, "A specifics overview of Systems for monitoring security events in different areas," 2023 XXXII International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 2023, pp. 1-4, doi: 10.1109/ET59121.2023.10279205.

[A5] **M. Hristov**, M. Nenova, Z. Valkova-Jarvis and V. Dimitrova, "Notification mechanism for malware detected by Microsoft Defender for IoT in industrial networks," 2023 14th National Conference with International Participation (ELECTRONICA), Sofia, Bulgaria, 2023, pp. 1-4, doi: 10.1109/ELECTRONICA58875.2023.11173900.

[A6] **M. Hristov**, M. Nenova, G. Iliev and D. Avresky, "Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT," 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 2021, pp. 1-5, doi: 10.1109/NCA53618.2021.9685977. **Citations:49**