Софийски университет "Св. Климент Охридски"
Факултет по математика и информатика
Катедра „Изчислителни системи”

РАЗРАБОТВАНЕ НА ЙЕРАРХИЧНА ТАКСОНОМИЯ ОТ МОДЕЛИ ЗА ПОДОБРЯВАНЕ НА СИГУРНОСТТА В ИНФОРМАЦИОННИТЕ СИСТЕМИ БАЗИРАНИ НА МИКРОСЪРВИСНА АРХИТЕКТУРА

Автореферат

на дисертационен труд за придобиване
на образователна и научна степен “Доктор”
в професионално направление
4.6 „Информатика и компютърни науки“
докторска програма „Компютърни науки“

Докторант: маг. инж. Тихомир Димитров Тенев
Научни ръководители: моц. д-р инж. Симеон Емилов Цветанов
доц. д-р Димитър Биров

София 2020

СЪДЪРЖАНИЕ

Списък с фигури

УВОД

Основната цел на дисертацията е разработването на йерархична таксономия от модели за подобряване на сигурността в информационните системи.

Предложената таксономия е посветена изцяло на софтуерна архитектура от ново поколение „Микросървисна архитектура“, която дава възможност на разработчици да създават гъвкави приложения покриващи големи натоварвания в кратки срокове без това да въздейства върху нормалната им работа.

Направен е анализ на заплахите за всяка една от областите дефинирани за архитектура от този вид, като това дава възможност за по-прецизното позициониране на избраните модели за сигурност.

При описанието на йерархичната таксономия е използван обектно ориентиран език за моделиране, който използва език за дефиниране на интерфейси прилагащ „Управляем формат на обектите“.

При изграждането на графичен интерфейс е използван способ за представяне на идеи върху хартия, така че да могат да бъдат графично изобразени. Той онагледява връзките между дефинираните области и избраните модели.

Разгледани са съвременни инструменти за управление на микросървиси, като към тях са предложени решения за употреба следвайки добрите практики на избраните модели за сигурност от йерархичната таксономия от модели за сигурност.

Дисертационния труд е оформен в осем глави, всяка от които завършва с изводи.

Първа глава разглежда микросървисна архитектура придружена с концептуално приложение. Прави се обосновка за необходимостта от повишаване на сигурността и се разглеждат предимствата от използването на модели за сигурност.

Втора глава разглежда заплахите в област Акаунт и Идентичност, предлага се списък от модели за сигурност и се дават препоръки за всеки един от моделите.

Трета глава разглежда заплахите в област Комуникация, предлага се списък от модели за сигурност и се дават препоръки за всеки един от моделите.

Четвърта глава разглежда заплахите в област Съхраняване на данни, предлага се списък от модели за сигурност и се дават препоръки за всеки един от моделите.

Пета глава разглежда заплахите в област Микросървисна среда, предлага се списък от модели за сигурност и се дават препоръки за всеки един от моделите.

Шеста глава разглежда заплахите в област Микросървиси разпределени върху платформи на различни доставчици, предлага се списък от модели за сигурност и се дават препоръки за всеки един от моделите.

Седма глава е посветена на изграждането на йерархична таксономия от модели за сигурност за информационни системи базирани на микросървисна архитектура, прави се избор на методология за описание на йерархична таксономия, дефинират се релациите между отделните области, създава се графичен интерфейс и се прави анализ на постигнатите резултати.

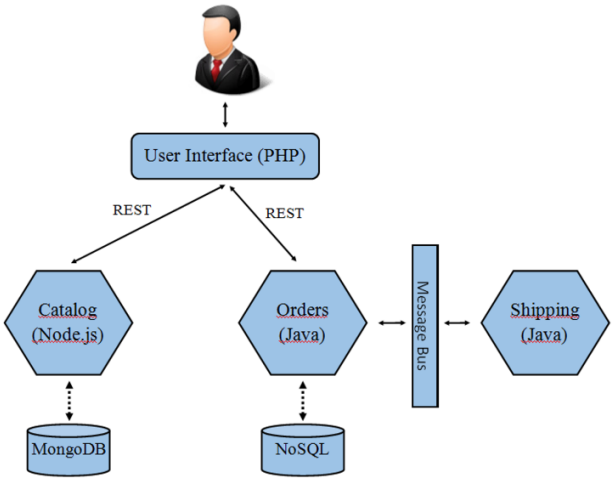
Осма глава описва имплементация с използването на съвременни инструменти за управление на микросървиси, които могат да бъдат конфигурирани следвайки добрите практики на избраните модели за сигурност от Йерархичната таксономия от модели за сигурност. Предложени са решения спрямо изискванията поставени във всеки един от моделите.

1 I ГЛАВА. ПОВИШАВАНЕ СИГУРНОСТТА НА СОФТУЕРНИ СИСТЕМИ БАЗИРАНИ НА МИКРОСЪРВИСНА АРХИТЕКТУРА

1.1 Микорсървисна архитектура.

Съвременно решение на проблемът с управлението на големи софтуерни приложения е използването на архитектура за разработване на софтуерни продукти наречена „Микросървисна архитектура“ [1]. Тя дава множество предимства пред редица други софтуерни архитектури. Пример за това е намаляване на сложността при големи приложения чрез модулиране на бизнес логиката на по-малки функционални единици Тези единици са озаглавявени под името микросървиси. Всяка една единица извършва точно определена дейност и не препокрива нечия друга функционалност.

На Фиг. 1 е показан концептуален архитектурен модел на онлайн магазин използващ Микросървисна архитектура. При него са показани само някои от основните функционалности необходими при преглед и поръчка на артикул. Отделните модули са проектирани да бъдат изготвени с помощта на различни програмни езици, с което се цели да се покаже гъвкавостта на архитектура от такъв тип.



Фиг. 1 Концептуален архитектурен модел

1.2 Необхоодимост от повишаване на сигурността чрез използване на модели за сигурност.

В днешни дни основно предизвикателство за индустриите е повишаването на сигурността в използваните от тях хостове и софтуерни продукти. Добър подход е сигурността да бъде осигурена още в самото начало на проектиране и изграждане на конкретна система и прилежащия към нея софтуер [2]. В противен случай, заплахи в сигурността могат да бъдат очаквани във всеки един момент по време на експлоатация. В този контекст, намирането на способи за противодействие на заплахи и прилагането им по правилния начин и на правилното място ще улесни многократно работата на

програмисти и инженери. Пример за такъв способ за противодействие е използването на модели за сигурност (Security Patterns).

Терминът модели за сигурност (Security Patterns) идва от така известните модели за дизайн на софтуер (Design Patterns) [3]. Както при моделите за дизайн, всеки модел за сигурност има своя структура. Тя служи за по-доброто му прочитане и прилагане. В голяма част от моделите, които са избрани и показани в следващите глави, съдържат следните точки – Цел (Intent), Контекст (Context), Проблем (Problem), Решение (Solution), Структура (Structure), Имплементация (Implementation), Следствия (Consequences) и Места на прилагане (Known uses). За по-интуитивен подбор на модели за сигурност, е решено да се вземат под внимание две основни точки – Контекст (Context) и Решение (Solution). Контекстът описва същината на ситуация, като включва в себе си предположения за обсег на действие, както и очаквания за среда, в която се предполага, че ще се прилагат. Решението дава яснота как даден проблем може да бъде разрешен. Това включва в себе си подробно разяснение на мерки, които трябва да бъдат взети под внимание.

Идеята за използване на модели за сигурност лежи върху проучването, което е направено за модели за дизайн [4] [5]. Резултатът на това проучване показва, че прилагането на модели за дизайн осезаемо допринася за подобряването на функционалностите на едно софтуерно приложение .

Основно затруднение, което се срещна по време на работа с модели за сигурност е, че те дават решение на по-общи проблеми. След подробно разследване на всеки един от тях се постигна удовлетворяваща трансформация, като решенията бяха отнесени към проблемите свързани със сигурността в една система изградена на принципа на Микросървисната архитектура. Тази трансформация е представена под форма на препоръки и е описана във всяка една от главите 2, 3, 4, 5 и 6.

1.3 Анализ на заплахите и препоръки при избор на модели за сигурност.

След направата на задълбочени проучвания се взе решение да се използва подход за анализ на заплахи базиран на STRIDE [6]. Този подход съветва читателите да разделят дадено приложение на няколко компонента, като с това се постига по-прецизно идентифициране на видовете атаки, които биха могли да му въздействат.Акронимът STRIDE означава: Измама (**S**poofing), Подправяне (**T**ampering), Отхвърляне (**R**epudiation), Разкриване на информация (**I**nformation Disclosure), Отказ от обслужване (**D**enial of Service) и Повишаване на привилегии (**E**levation of Privilege).

В следващите няколко точки е направен анализ на заплахи в различните области от микросървисната архитектура. След което се прави категоризиране на всички избрани модели за сигурност, за да може да бъдат причислени към различните категории на STRIDE. Някои от моделите покриват повече от една от категориите на STRIDE.

В глави 2, 3, 4, 5 и 6 са представени няколко списъка от модели за сигурност за всяка една от дефинираните областите. Взети са под внимание предимствата и недостатъците при използване на микросървисна архитектура за разработване на софтуерно приложение, нуждата от подобрения на сигурността и резултатът от анализа на заплахите. Всеки от представените моделите е придружен от препоръка за място на употреба, което би подпомогнало разработчици в процесът на изграждане на устойчиво приложение, базирано на микросървисна архитектура. Препоръките за място на използване на всеки модел за сигурност се извлича от неговите структурни точки – Контекст (Context) и Решение (Solution) и след това се привежда в съответствие с конкретна ситуация.

1.4 Изготвяне на йерархична таксономия от модели за сигурност

Йерархичната таксономия от модели за сигурност, представена в глава 7, е постигната благодарение на анализът направен в следващите няколко глави - 2, 3, 4, 5 и 6. Във всяка една от тези глави е представена различна категория от микросървисната архитектура, както и списък с модели за сигурност: „Акаунт и Идентичност (Account and Identity)“, “Комуникация (Communication)”, „Съхраняване на данни (Data Persistance)“, “Микросървисна среда (Environment)“ и „Различни доставчици (Third Party Suppliers)“. За по-голяма прецизност областта “Комуникация (Communication)” е разделена на две подобласти – “Взаимовръзки (inter-connection)“ и „Съобщения (messages)”. Описанието на самата таксономия е направена с помощта на обектно ориентирано моделиране CIM (Common Information Model) [7]. CIM е концептуален информационен модел за описание на различни субекти, който допринася за структурирането на събраните до момента модели за сигурност. Цялата онтология е трансформирана в графичен вид с помощта на безплатна онлайн платформа за UI Wireframing [8].

1.5 Избор на съвременни технологии за управление на микросървиси

Всяка една от разгледаните пет области на микросървисно приложение може да бъде управлявано посредством различни технологии. Изборът на съвременни технологии за управление на микросървиси е описано в глава 8. Областта „Акаунт и Идентичност (Account and Identity)” може да бъде управлявана с помощта на продукта Kong [9]. Областта „Комуникация“ е разделена на две части „Междусървисна комуникация“ и „Съобщения“ . Инструментът, който може да се използва при осъществяване на комуникация между микросървиси е RabbitMQ [10]. Инструментите, които се разглеждат в областта Съхраняване на данни са: база от данни MongoDB [11] и файлова система ext4 [12] предоставена от операционна система CentOS [13]. Инструментите предлагащи функционалности, които могат да бъдат приложени в област „Микросървисна среда“ са Docker container [14], CentOS [13] и IaaS [15]. Областта „Микросървиси разположени върху платформи на различни доставчици“ може да бъде управлявана с помощта на инструменти предлагащи клъстеризация като Kubernetes (K8s) [16].

ЦЕЛ И ЗАДАЧИ НА ДИСЕРТАЦИОННИЯ ТРУД

Целта на дисертационния труд е разработването на йерархична таксономия от модели за подобряване на сигурността в софтуерни системи базирани на микросървисна архитектура. Таксономията лежи върху отделните категории дефинирани за микросървисна архитектура, както и анализ на заплахите направен за всяка една от тези категории. Това дава възможност на по-прецизно подбиране на откритите модели за сигурност (Security Patterns).

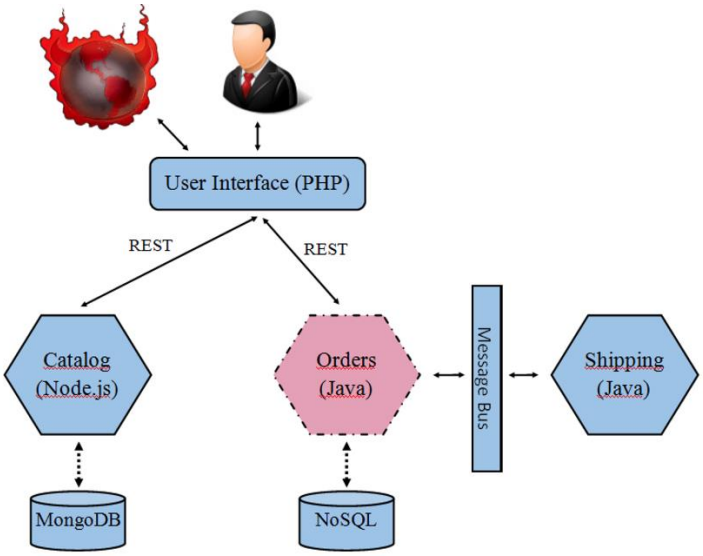
Във връзка с основната цел са формулирани следните задачи:

1. Да се категоризира и гранулира софтуерна архитектура от ново поколение, наричана още Микросървисна архитектура, с цел представянето и под различни форми.
2. Да се направи анализ на заплахите за всяка една от представените категории на микросървисната архитектура.
3. Да се открият подходящи модели за сигурносткоито да влизат в обхвата на отделните категории, уповавайки се на анализът на заплахи.
4. Да се трансформират обосновките и решенията предоставени от различните модели за сигурност към контекста на мокросървисна архитектура.
5. Да се направи йерархичен модел на всички категории от микросървисната архитектура, който да служи като скелет при изграждането на подробна йерархична таксономия.
6. Да се открие и използва подходящ обектно ориентиран език за моделиране с цел привеждане на избраните модели за сигурност в четим вид, като се спазва йерархичността на отделните категории.
7. Да се трансформира езикът за моделиране, така че да може да бъде представен в графичен вид.
8. Да се намерят съвременни продукти с помощта на които може да се създаде устойчива среда за микросървисно приложение.
9. Да се направи изследване, кои от представените модели за сигурност могат да бъдат прилагани посредством избраните съвременни продукти за управление на микросървисни приложения, както и да се даде примерно решение.

2 II ГЛАВА. ПРЕДЛОЖЕНИЯ ЗА СМЕКЧАВАНЕ НА ЗАПЛАХИ В ЧАСТ АКАУНТ И ИДЕНТИЧНОСТ (ACCOUNT AND IDENTITY)

2.1 Концептуален модел

Една от целите в този раздел е да се покажат уязвимостите, които биха могли да се проявят при досег на софтуерни приложения, базирани на микросървисна архитектура, с външни потребители –Фиг. 2 . Разгледан е и случай, в който по някаква причина микросървис е поставен с уязвимост в него (Orders), като това не бива да пречи на правилната работа на цялото приложение.



Фиг. 2 Опит за пробив на миркосървисно приложение

2.2 Анализ на заплахите

В текущата точка се прави анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия домейн. Взимайки в предвид условията описани за всяка една от точките от модела за откриване на заплахи STRIDE и тълкувайки ги спрямо контекста на избраната среда “Акаунт и Идентичност (Account and Identity)” в API Gateway, може да се направи следният анализ:

Измама (Spoofing) е вид злоупотреба, при която нарушителят се опитва да получи достъп до система или информация на потребител, като се преструва, че е самият потребител.

Подправяне (Tampering) означава извършване на незаконни промени с цел смяна на ключ за достъп до микросървис или промяна на информация за работни сесии.

„***Отхвърлянето***“ (Repudiation) се интерпретира като отрицание да се приеме дадено твърдение за вярно.

Разкриване на информация (Information Disclosure) може да се случи, когато неоторизиран потребител достъпи информация, която е забранено за разкриване.

Отказът от услуга (Denial of Service) се отнася най-вече до изчерпване на ресурсите на един или няколко елемента от сървърите, върху които се помещават микросървисите.

Повишаване на привилегии (Elevation of Privilege) означава да се позволи на потребител да изпълнява команда без той да разполага с необходимите права.

2.3 Препоръки при избор на модели за сигурност

Предложени са модели за сигурност, които имат връзка с категория „Акаунт и Идентичност (Account and Identity)“. Към всеки един модел са добавени и препоръки за места, където биха могли да бъдат използвани. Списъкът с всички модели, както и STRIDE категорията, към която спадат, е показан на Фиг. 3. Публикуван е научен труд в тази насока [17].

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
A Pattern for WS-Trust	▯		▯			▯
Access Control List	▯				▯	
Account Lockout	▯					
Actor and Role Lifecycle Pattern			▯	▯		
Administrator Objects	▯					▯
Authenticated Session	▯					
Authenticator	▯					
Authorization						▯
Biometrics Design Alternatives	▯					
Capability	▯				▯	
Client Input Filters	▯					▯
Credential Delegation				▯		
Directed Session		▯				
Grant-Based Access Control Pattern				▯		
Password Design and Use						▯
Privilege-Limited Role				▯		
Role-Based Access Control (RBAC)	▯					
Session-Based Attribute-Based Authorization	▯					▯

Фиг. 3 Модели за сигурност за Акаунт и Идентичност

Заключение

Разгледано са особеностите на микросървисно приложение в част “Акаунт и Идентичност (Account and Identity)”. Разгледан е и случай, в който по някаква причина микросървис е поставен с уязвимост в него.

Предложен е концептуален модел на микросървисно приложение разположено върху облачна платформа. Специфицирани са и сървис моделите, които отговарят на приложение от такъв тип.

Направен е избор на връзка между микросървисно приложение и външни потребители. Първият пример е с директна връзка „Потребител-Микросървис“, а вторият чрез внедряване на допълнителен инструмент – „ API Gateway”.

Направен е анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия домейн. Използваният подход за анализ на заплахите е STRIDE.

Предложени са модели за сигурност, които имат връзка с частта „Акаунт и Идентичност (Account and Identity)“.

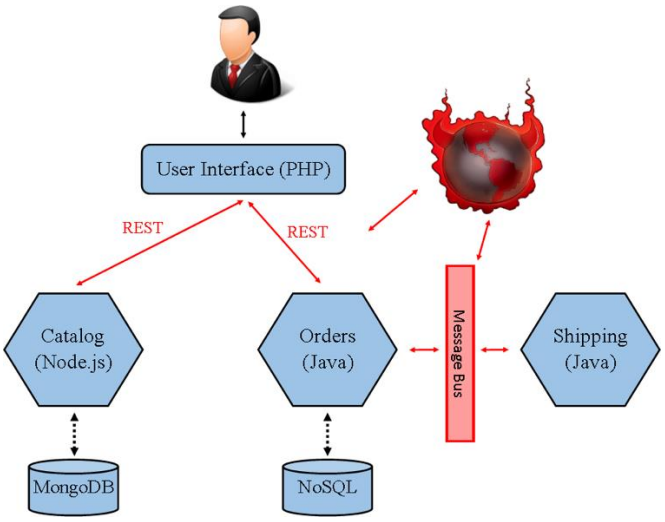
Направено е категоризиране на всеки един от моделите за сигурност спрямо анализът на заплахи базиран на STRIDE.

Добавени са препоръки към всеки един от моделите за сигурност относно местата, където биха могли да бъдат използвани.

3 III ГЛАВА. ПРЕДЛОЖЕНИЯ ЗА СМЕКЧАВАНЕ НА ЗАПЛАХИ В ЧАСТ КОМУНИКАЦИЯ

3.1 Концептуален модел

Една от целите е да се покаже уязвимостите, които биха могли да се проявят при комуникация към и между микросървиси разположени в един домейн – Фиг. 4. За по-голяма прецизност се взе решение да се раздели комуникацията на две подточки - „Междусървисна комуникация“ и „Съобщения“



Фиг. 4 Опит за подслушване на комуникация между микросървиси

3.2 Анализ на заплахите

В текущата точка се прави анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от инструменти за комуникация разположени в същия домейн. Взимайки в предвид условията описани за всяка една от точките от способа за откриване на заплахи STRIDE и тълкувайки ги спрямо контекста на избраната среда „Комуникация“ и в частност „Междусървисна комуникация“ и „Съобщения“, може да бъде направен следния анализ:

Измама (Spoofing) е вид злоупотреба, при която нарушител се опитва да получи достъп до шина, към която се свързват всички слушащи микросървиси, заблуждавайки, че е оторизиран както всички останали.

Подправяне (Tampering) означава извършване на незаконни промени в потока от данни, когато такива действия са забранени.

„***Отхвърлянето***“ (Repudiation) бива интерпретирано като отрицание да се приеме дадено твърдение за вярно.

Разкриване на информация (Information Disclosure) може да се случи, когато неоторизиран потребител достъпи информация, която е забранено за разкриване.

Отказът от услуга (Denial of Service) се отнася най-вече до изчерпване на ресурсите на един или няколко сървърни компонента, които се използват от микросървиси.

Повишаване на привилегии (Elevation of Privilege) означава да се позволи на потребител да се свърже към шина, за която няма права.

3.3 Препоръки при избор на модели за сигурност

Предложени са модели за сигурност, които имат връзка с частта „Комуникация“ и в частност „Междусървисна комуникация“ и „Съобщения“. Към всеки един модел са добавени и препоръки за места, където биха могли да бъдат използвани. Списъкът с всички модели, както и STRIDE категорията, към която спадат, е показан на Фиг. 5 и Фиг. 6. Публикуван е научен труд в тази насока [18].

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Authoritative Source of Data		▯			▯	
Content–independent Processing		▯		▯		
Input Guard		▯				
Multiple Secure Observers				▯		
Output Guard		▯				
Secure Channels		▯		▯		
Secure Communication				▯		
Security Association	▯	▯				
XML Firewall	▯	▯				

Фиг. 5 Модели за сигурност при Междусървисна комуникация

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Anonymity Set				▯		
Cryptographic Protocol				▯		
DATA INTEGRITY IN P2P-SYSTEMS		▯		▯		
Hidden Metadata				▯	▯	
Layered Encryption				▯		
Morphed Representation				▯		
XML Encryption Syntax and Processing		▯		▯		

Фиг. 6 Модели за сигурност при Съобщения

Заключение

Разгледано е микросървисно приложение в част Комуникация. За по-голяма прецизност частта Комуникация е разделена на две подточки „Междусървисна комуникация“ и „Съобщения“ .

Предложен е концептуален модел на микросървисно приложение използващо инструменти за комуникация REST и Message Bus, респективно са взети в предвид подходите за комуникация – Синхронен и Асинхронен.

Отчетени са различните видове пакети, които могат да преминават през конекторите на микросървисно приложение – Binary, XML и JSON.

Направен е анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия домейн. Използваният подход за анализ на заплахите е STRIDE.

Предложени са модели за сигурност, които имат връзка с частта Комуникация и по-конкретно с „Междусървисна комуникация“ и „Съобщения“.

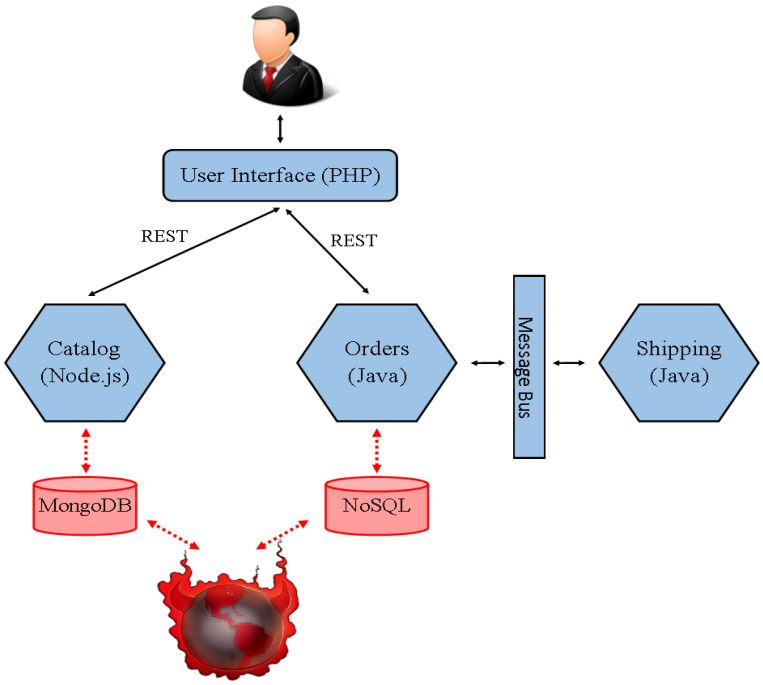
Направено е категоризиране на всеки един от моделите за сигурност спрямо анализът на заплахи базиран на STRIDE.

Добавени са препоръки към всеки един модел за сигурност относно местата, където биха могли да бъдат използвани.

4 IV ГЛАВА. ПРЕДЛОЖЕНИЯ ЗА СМЕКЧАВАНЕ НА ЗАПЛАХИ В ЧАСТ СЪХРАНЯВАНЕ НА ДАННИ

4.1 Концептуален модел

Основната цел на този раздел е да бъдат показани уязвимостите, които биха могли да се проявят при управление и съхраняване на данни от страна на микросървиси –Фиг. 7.



Фиг. 7 Опит за манипулиране на данни съхранявани от микорсървиси

4.2 Анализ на заплахите

В текущата точка се прави анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от инструменти подпомагащи управлението и съхраняването на данни. Взимайки в предвид условията описани за всяка една от точките от модела за откриване на заплахи STRIDE и тълкувайки ги спрямо контекста на избраната среда „Съхраняване на данни“ се прави следния анализ на заплахи:

Измама (Spoofing) е вид злоупотреба, при която нарушител се опитва да получи достъп до данни, притежавани от друг микросървис.

Подправяне (Tampering) означава извършване на незаконни промени в съхранените от микросървис данни, когато такива действия са забранени.

Отхвърляне (Repudiation) е интерпретирано като отрицание да се приеме дадено твърдение за вярно.

Разкриване на информация (Information Disclosure) може да се прояви, когато неоторизиран потребител достъпи файлове, за които няма права.

Отказът от услуга (Denial of Service) се отнася най-вече до изчерпване на ресурси на един или няколко елемента от сървъри, върху които работят микросървиси.

Повишаване на привилегии (Elevation of Privilege) означава да се позволи на потребител да достъпи хранилище на данни, за което няма права.

4.3 Препоръки при избор на модели за сигурност

Предложени са модели за сигурност, които имат връзка с частта „Съхраняване на данни“. Към всеки един модел са добавени и препоръки за места, където биха могли да бъдат използвани. Списъкът с всички модели както и STRIDE категорията, към която спадат, е показан на Фиг. 8. Публикуван е научен труд в тази насока [19].

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Clear Sensitive Information[8]				▯		
Encrypted Storage[11]		▯		▯		
File Authorization[1]	▯	▯		▯		
Limited View[10]				▯		
Multilevel Security[1]		▯		▯		
Secure Directory[8]		▯				

Фиг. 8 Модели за сигурност при Съхраняване на данни

Заключение

Разгледано е микросървисно приложение в част Съхранение на данни.

Отчетено е изискването при конструиране на микросървисно приложение - всички данни собственост на даден микросървис да не бъдат предоставени за ползване от останалите микросървиси.

Разгледан е вариант за обмен на данни между микросървиси. Инструментът, който позволява такъв обмен е Message Bus.

Отчетено е предимството на микросървисни приложения лесно да бъдат трансформирани и поставени в облачна система.

Направен е анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия домейн. Използваният подход за анализ на заплахите е STRIDE.

Предложени са модели за сигурност, които имат връзка с частта Съхранение на данни.

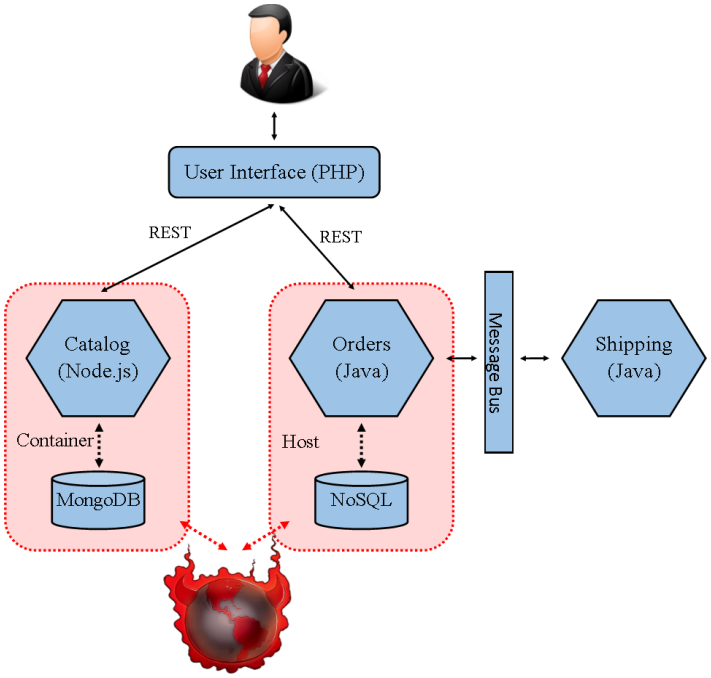
Направено е категоризиране на всеки един от моделите за сигурност спрямо анализът на заплахи базиран на STRIDE.

Добавени са препоръки към всеки един модел за сигурност относно местата, където биха могли да бъдат използвани.

5 V ГЛАВА. ПРЕДЛОЖЕНИЯ ЗА СМЕКЧАВАНЕ НА ЗАПЛАХИ В ЧАСТ МИРКОСЪРВИСНА СРЕДА

5.1 Концептуален модел

Проблемът, който е изследван тук е, как може да се създаде сигурна среда, която оптимално да побира микросървисите. В повечето случаи микросървисите могат да бъдат разпространявани или в самостоятелна операционна система, или чрез използване на контейнери [1] [20] – Фиг. 9.



Фиг. 9 Опит за въздействане върху средата, която помещават микросървиси

5.2 Анализ на заплахите

В текущата точка се прави анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от инструменти подпомагащи управлението и съхраняването на данни. Взимайки в предвид условията описани за всяка една от точките от модела за откриване на заплахи STRIDE и тълкувайки ги спрямо контекста на избраната част от „Микросървисна среда“ се прави следния анализ на заплахи:

Има вид измама, при която нарушител се опитва да получи достъп както до система помещавана от микросървиси така и до информация, която микросървис притежава, като се преструва, че има нужната оторизация. Тези случаи обикновено водят до нежелани последствия и се отнасят до категория „***Измама***“ (Spoofing).

Запазването на целостта на тази информация е строго наложително. STRIDE категория отнасяща се към опазване целостта на данни е „***Подправяне***“ (Tampering).

Ако потребител извърши действие, но твърди, че не го е направил, това би намалило отговорността. STRIDE категорията, към която спадат такъв тип заплати е „***Отхвърляне***“ (Repudiation).

Друга ситуация е, когато неоторизиран потребител/микросървис оперира с информация, която е забранена за разкриване. Най-близката STRIDE категория е ***Разкриване на информация*** (Information Disclosure).

Следваща заплаха, е когато има опит за препълване на хардуерен ресурс - памет, процесор, хранилище на данни или други. Категорията, която е отнесена към работоспособността на микросървис е „***Отказ от услуга***“ (Denial of Service).

Следващ пример е, когато потребител има възможност да направи нещо без необходимите специфични права за достъп. Категорията, която разглежда ситуации от такъв вид е „***Повишаване на привилегии***“ (Elevation of Privilege).

5.3 Препоръки при избор на модели за сигурност

Предложени са модели за сигурност, които имат връзка с частта „Микросървисна среда“. Към всеки един модел се добавя и препоръки за места, където би могъл да бъде използвани. Списъкът с всички модели както и STRIDE категорията, към която спадат, е показан на Фиг. 10 и Фиг. 11. Публикуван е научен труд в тази насока [21].

Patterns	Threat modelling approach					
	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Administrator Hierarchy	▯		▯			▯
Building the Server from the Ground Up		▯			▯	
Checkpointed Systems					▯	
Controlled Execution Environment		▯				
Documenting the Server Configuration		▯		▯	▯	
Patching Proactively		▯		▯	▯	
Pathname Canonicalization		▯				
Protection Rings		▯				
Testing on a Staging Server					▯	
Secure IaaS/open IaaS/OpenStack	▯	▯	▯	▯	▯	▯

Фиг. 10 Модели за сигурност при Микросървисна среда

Patterns	Deployment approach	
	Standalone OS	Container
Administrator Hierarchy	▯	▯
Building the Server from the Ground Up	▯	
Checkpointed Systems		
Controlled Execution Environment	▯	
Documenting the Server Configuration	▯	▯
Patching Proactively	▯	▯
Pathname Canonicalization	▯	
Protection Rings	▯	
Testing on a Staging Server	▯	▯
Secure IaaS/open IaaS/OpenStack		

Фиг. 11 Видове подходи за поставяне на микорсиървиси (deployment) отнесени към всеки един от моделите за сигурност

Заключение

Разгледано е микросървисно приложение в част Микросървисна среда. За повече яснота са разгледани няколко среди, върху които могат да бъдат разположени микрисървиси: Операционна Система, Контейнер и IaaS.

Предложен е концептуален модел на микросървисно приложение разположено върху три от избраните среди за съхранение: Операционна Система, Контейнер и IaaS. Отчетени са предимствата и недостатъците на трите вида среди, в които микросървиси могат да бъдат разположени

Направен е анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия домейн. Използваният подход за анализ на заплахите е STRIDE.

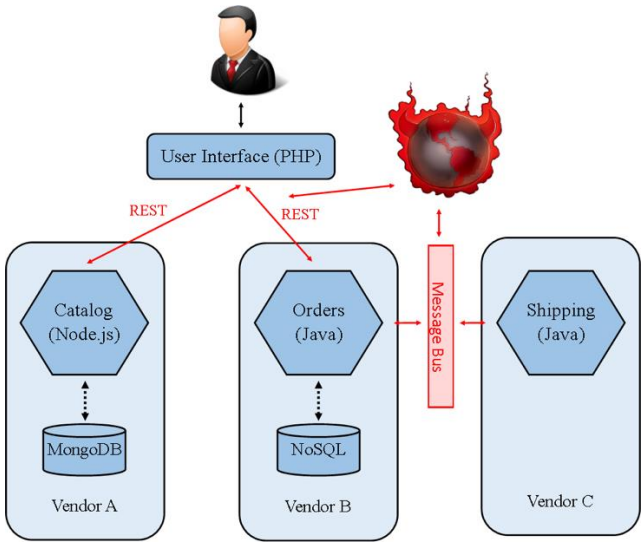
Предложени са модели за сигурност, които имат връзка с частта „Микросървисна среда“. Направено е категоризиране на всеки един от моделите за сигурност спрямо анализът на заплахи базиран на STRIDE.

Добавени са препоръки към всеки един модел за сигурност относно местата, където биха могли да бъдат използвани.

6 VI ГЛАВА. ПРЕДЛОЖЕНИЯ ЗА СМЕКЧАВАНЕ НА ЗАПЛАХИ В ЧАСТ МИРКОСЪРВИСИ РАЗПРЕДЕЛЕНИ ВЪРХУ ПЛАТФОРМИ НА РАЗЛИЧНИ ДОСТАВЧИЦИ

6.1 Концептуален модел

Една от задачите на този раздел е да се покажат уязвимостите, които биха могли да се проявят в софтуерни приложения, за които е взето решение да бъдат разделени и позиционирани в два или повече доставчика – Фиг. 12



Фиг. 12 Опит за въздействане върху комуникацията между микросървиси разпределени в различни доставчици

6.2 Анализ на заплахите

В текущата точка се прави анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от инструменти даващи възможност за установяване на комуникация между доставчици. Взимайки в предвид условията описани за всяка една от точките от модела за откриване на заплахи STRIDE и тълкувайки ги спрямо контекста на избраната част „Микросървиси разпределени върху платформи на различни доставчици“ се прави следния анализ на заплахи:

Пример за „***Измама***“ (Spoofing) е, когато даден микросървис се опитва да поиска данни от микросървис разположен в друг доставчик, който вече е под контрол на злонамерен потребител.

Подправяне (Tampering) означава извършване на незаконни дейности с цел промяна на настройки на защитна стена.

Отхвърляне (Repudiation) или в частност отрицание да се приеме дадено твърдение за вярно.

Разкриване на информация (Information Disclosure) може да се случи, когато неоторизиран потребител манипулира конфигурационните файлове на микросървисната среда без да има нужните права.

Отказът от услуга (Denial of Service) разглежда случаи, в които има изчерпване на ресурси.

Повишаване на привилегии (Elevation of Privilege) разглежда случаи, в които даден потребител е придобил права позволяващи му да управлява софтуерно приложение разположено върхо няколко доставчици.

6.3 Препоръки при избор на модели за сигурност

Предложени са модели за сигурност, които имат връзка с частта „Микросървиси разпределени върху платформи на различни доставчици“. Към всеки един модел са добавени и препоръки за места, където биха могли да бъдат използвани. Списъкът с всички модели, както и STRIDE категорията, към която спадат, е показан на Фиг. 13.Публикуван е научен труд в тази насока [22].

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
3rd Party Communication		▯		▯		
AGENCY GUARD		▯		▯		
AGENT AUTHENTICATOR	▯					
Application Firewall	▯				▯	
Cloud Access Security Broker	▯					▯
Integration Reverse Proxy					▯	
Known Partners	▯			▯		▯
Log for Audit		▯	▯			
Secure Assertion		▯	▯		▯	
Secure Logger			▯			

Фиг. 13 Модели за сигурност при микросървиси разпределени върху платформи на различни доставчици

Заключение

Разгледано е микросървисно приложение в част „Микросървиси разположени върху платформи на различни доставчици“. Сървис моделът, който е взет под внимание е PaaS (Platform as a Service).

Предложен е концептуален модел на микросървисно приложение разпределено върху два доставчика.

Направен е анализ на заплахите върху система базирана на микросървисна архитектура. Заплахите биха могли да идват както от външни потребители, така и от микросървиси разположени в същия доставчик. Използваният подход за анализ на заплахите е STRIDE.

Предложени са модели за сигурност, които имат връзка с частта „Микросървиси разположени върху платформи на различни доставчици“.

Направено е категоризиране на всеки един от моделите за сигурност спрямо анализът на заплахи базиран на STRIDE.

Добавени са препоръки към всеки един модел за сигурност относно местата, където биха могли да бъдат използвани.

7 VII ГЛАВА. РАЗРАБОТВАНЕ НА ЙЕРАРХИЧНА ТАКСОНОМИЯ ОТ МОДЕЛИ ЗА ПОДОБРЯВАНЕ НА СИГУРНОСТТА В СОФТУЕРНИ ПРИЛОЖЕНИЯ БАЗИРАНИ НА МИКРОСЪРВИСНА АРХИТЕКТУРА

7.1 Избор на методология за описание на йерархична таксономия

Избраната методология за описание на таксономия на модели за сигурност от йерархичен тип се осланя на правилата декларирани в CIM (Common Information Model) [7]. CIM е концептуален информационен модел за описание на различни управленчески субекти, приложни обекти и доставчици на услуги. Информационни модели от такъв тип използват различни по вид елементи като класове (Classes), свойства (Properties), методи (Methods) и асоциации (Assosiacion). В следващите подточки са предоставени примери на всеки един от използваните елементи.

За целите на текущите изследвания се използва Езика за дефиниране на интерфейса (Interface Definition Language (IDL)) или така нареченият Управляем формат на обекта (Managed Object Format (MOF)). Синтаксисът на MOF е описан под формата на нотации, които са дефинирани в стил на Augmented BNF for Syntax Specifications [23]. Главният способ за описание на обекти и инстанции към тях е в текстов вид. Коментари към всеки един от елементите е разрешен само в Unicode или UTF-8 формат. В текущият случай се използват модели за сигурност в контекста на софтуерно приложение базирано на микросървисна архитектура.

За име на схема е използвано общото название „Security”. Връзката Схема-Клас се обозначава с помощта на знак „_“:

class Security_MicroserviceSecurity {}

Йерархията на класовете започва с абстрактен клас, който дефинира основните елементи, които всички подкласове ще наследят. За абстрактен клас е избрано името „*MicroserviceSecurity*“. Класовете, които наследяват абстрактен клас *MicroserviceSecurity* представляват отделните области на микросървисна архитектура дефинирани в глави 2, 3, 4, 5 и 6. Всички избрани модели за сигурност също са представени под формата на класове. Релации между класовете, представящи моделите за сигурност, и съпътстващите ги области, са описани в подточка 7.2. Връзката между класове и абстрактен клас се дефинира чрез „:“ :

[Description (" ... „]
class Security_APatternforWSTrust : Security_MicroserviceSecurity { }

Свойствата представляват стойност, използвана за характеризиране на клас. Свойствата са уникални и биват дефинирани в обхвата на класа. Те се състоят от име,

тип данни и стойност. Свойствата са дефинирани в абстрактен клас „*MicroserviceSecurity“* като:

[Override ("Context"), MaxLen (64), Description ("... "]
string Context;

Стойността им е дадена в подкласовете на всички модели за сигурност:

Context = " ... ";

Референцията е специален тип свойство използвано като указател към инстанция на клас. Декларира се с ключова дума REF:

[Override ("PatternAuthenticator"),
Max (1)]
Security_Authenticator REF PatternAuthenticator;

Асоциацията е тип клас, който има две или повече референции. Асоциациите представляват връзки между два или повече класа. С негова помощ се осъществява групиране на отделните модели за сигурност към съответната им категория:

[Abstract, Description (" .. ")]
class Security_DataPersistance : Security_Environment {

[Override ("PatternClearSensitiveInformation"),
Max (1)]
Security_ClearSensitiveInformation REF PatternClearSensitiveInformation;
}

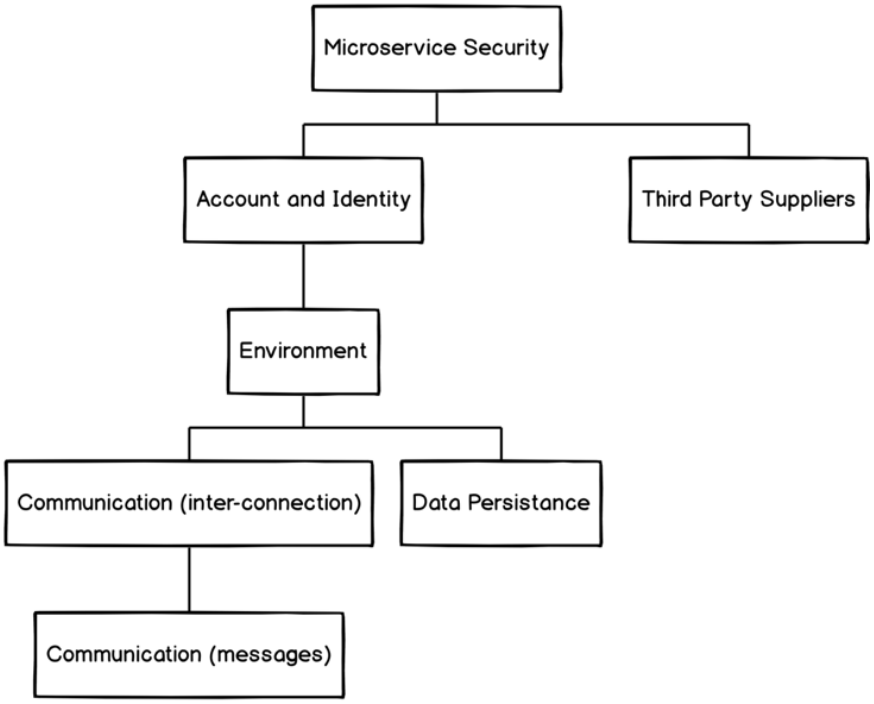
Квалификаторите са стойности, които предоставят допълнителна информация за класове, асоциации, свойства или референции. Всички квалификатори имат име, тип, стойност и обхват. Използваните квалификатори се използват, за да се обособят колко пъти (*Max (1)*) даден модел за сигурност може да бъде използван в контекста на категорията, към която спада:

[Override ("PatternAdministratorHierarchy"),
Max (1)]
Security_AdministratorHierarchy REF PatternAdministratorHierarchy;

7.2 Дефиниране на релации между отделните области

В глави 2, 3, 4, 5, 6 се разглеждат всичките пет области – „Акаунт и Идентичност (Account and Identity)“, “Комуникация (Communication)”, „Съхраняване на данни (Data Persistance)“, “Среда (Environment)“ и „Различни доставчици (Third Party Suppliers)“. За по-голяма прецизност областта “Комуникация (Communication)” е разделена на две

подобласти – “ Междусървисна комуникация (Interconnection)“ и „Съобщения (Messages)”. На Фиг. 14 е показана подредбата на всички области:



Фиг. 14 Йерархия на областите в микросървисна архитектура

Повишаването на сигурността (Microservice Security) стои на най-високо ниво в йерархията. Тя се разклонява в две посоки. Първата е в контекста на повишаване сигурността само в обсега на един доставчик на услуги [24] без значение дали софтуерно приложение е разположено върху повече от един доставчик. Ако има наличие на два или повече доставчици то това разклонение се прилага за всеки доставчик по отделно. Второто разклонение е покрива единствени случаите, в които има наличие на повече от един доставчик на услуги. Той се грижи за повишаване на сигурността между отделните доставчици.

Областта „Account and Identity” заема първо място в първото разклонение на „Microservice Security”. Причината за това е, че той е от най-голямо значение, поради фактът, че крайните потребители са тези, които създават предпоставки за изтичане на информация. Ако те имат неограничени права към цялата система, това би помогнало на злонамерени външни потребители да се сдобият с тези права и да направят непоправими неща. Следваща по ред област е „Environment”. Правилното конфигуриране и експлоатиране на средата е от съществено значение, за да се избегне нежелано поведение. Обновяването на работните библиотеки, които се използват от приложение, ще смекчи уязвимости породени от бъгове в тях. Средата “Environment” се разклонява на две подобласти. Първата е свързано с комуникацията “Communication”, която се осъществява между отделните микросървиси. Втората е свързано с информацията, която се съхранява от страна на всеки един микросървис “Data Persistance”.Комуникацията (Communication) е разделена на две подобласти - “Междусървисна комуникация

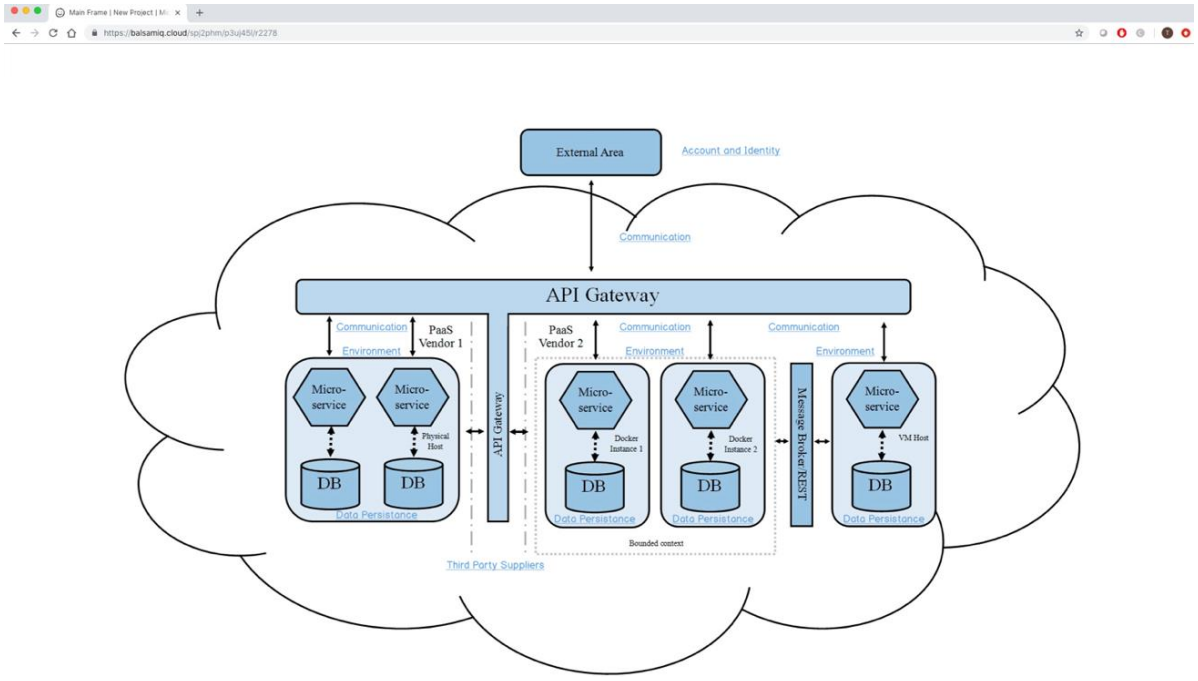
(Interconnection)“ и „Съобщения (Messages)“. Първа по ред е поставена “ Междусървисна комуникация (inter-connection)“, защото тя е основополагаща при комуникация между отделните микросървиси. Първо се установява връзката между микросървисите и тогава започва прехвърлянето на съобщения (Messages).

Моделите за сигурност са групирани и представени в глави 2, 3, 4, 5, и 6 и по конкретно във Фиг. 3, Фиг. 5, Фиг. 6, Фиг. 8, Фиг. 10 и Фиг. 13. Те са представени в табличен вид с пояснение към коя STRIDE категория принадлежат. Всяка една от областите показани на Фиг. 14 има към себе си списък от модели за сигурност. Всички релации в CIM между отделните области се осъществяват посредством една от функционалностите наречена наследяване. Всички категории са представени под формата на класове.

7.3 Графично изобразяване на йерархична таксономия

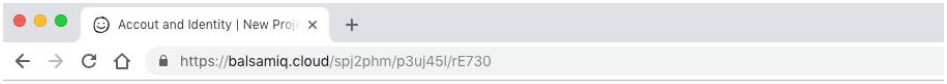
Данните предоставени под формата на CIM обектно моделиране могат да бъдат изобразени с помощта на различни инструменти. Примерът, който се разглежда е чрез използване на UI Wireframing [8]. Wireframing е способ за представяне на идеи върху хартия, така че да могат да бъдат графично изобразени. Това спомага по-доброто онагледяване на йерархичната таксономия представена по-горе.

На Фиг. 15 е показана схема на софтуерно приложение използващо микросървисна архитектура. Към схемата са прибавени и линкове, които предоставят достъп до списъци до всяка една от категориите описани в глави 2, 3, 4, 5 и 6.



Фиг. 15 Схема на софтуерно приложение използващо микросървисна архитектура

Натискайки върху всяка една от категориите ще отвори втора страница, която показва всички модели за сигурност свързани към съответстващата им категория. На Фиг. 16 е показан пример с натискането на линк към „Account and Identity”.

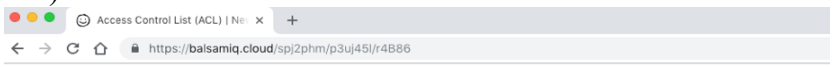


Patterns for Account and Identity

Patterns	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
A Pattern for WS-Trust	✓		✓			✓
Access Control List	✓				✓	
Account Lockout	✓					
Actor and Role Lifecycle Pattern			✓	✓		
Administrator Objects	✓					✓
Authenticated Session	✓					
Authenticator	✓					
Authorization						✓
Biometrics Design Alternatives	✓					
Capability	✓				✓	
Client Input Filters	✓					✓
Credential Delegation				✓		
Directed Session		✓				
Grant-Based Access Control Pattern				✓		
Password Design and Use						✓
Privilege-Limited Role				✓		
Role-Based Access Control (RBAC)	✓					
Session-Based Attribute-Based Authorization	✓					✓

Фиг. 16 Списък от модели за сигурност представени в графичен вид

Всеки един от моделите за сигурност има хиперлинк към информация, която е записана в CIM класовете. На Фиг. 17 е даден пример с модел за сигурност „Access Control List (ACL)“.



Pattern = "Access Control List (ACL)";

Context = "This applies to distributed systems where access to resources must be controlled. Those systems comprise a Policy Decision Point and Policy Enforcement Points, which enforce the access policy. A system is composed of subjects that need to access resources to perform their tasks. In the system, not every subject can access any object: access rights are defined and can be modeled as an access matrix, in which each row represents a subject and each column represents an object. An entry of the matrix is indexed by a specific subject and a specific object, and lists the types of actions that this subject can execute on this object.";

Solution = "Implement the Access Matrix by associating each object with an Access Control List (ACL) that specifies which actions are allowed to be performed on the object and by which authenticated users. Each entry of the list comprises a subject's identifier and a set of rights. Policy Enforcement Points (PEPs) of the system enforce the access policy by requesting to the PDP to search the object's ACL for the requesting subject identifier and access type. In order for the system to be secure, the subject's identity must be authenticated prior to its access to any objects. Since the ACLs may be distributed, like the objects they are associated with, several Policy Administration Points (PAPs) may be responsible for creating and modifying the ACLs.";

STRIDEAccronym = { S, D };

Reference = "N. Delessy, E. B. Fernandez, M. M. Larrondo-Petrie и J. Wu, „Patterns for Access Control in Distributed Systems," in Conference on Pattern Languages of Programs, New York, NY, USA, 2007.";

Фиг. 17 Графично представяне на модел за сигурност "Access Control List (ACL)"

7.4 Анализ на постигнатите резултати

Предимствата от използването на CIM обектно ориентирано моделиране са: възможността за структуриране на данни описващи всеки един от моделите за сигурност, осъществяване на асоциации, които да групират модели за сигурност отнесени към конкретна област и изграждане на релации между отделните области. Тези три аспекта формират йерархична таксономия от модели за сигурност за подобряване на сигурността в софтуерни приложения базирани на микросървисна архитектура.

Заключение

Избраната методология за описание на таксономия на модели за сигурност от йерархичен тип се осланя на правилата декларирани в CIM (Common Information Model).

Избраният езикът за описание на всички елементи от CIM методологията е решено да спазва условията поставени в Управляем формат на обекта (Managed Object Format (MOF)).

Използваните елементи, формиращи цялата йерархична таксономия на модели за сигурност са: Схема, Клас, Свойства, Връзки и Асоциации, Квалификатор.

Графично е предоставена йерархия на областите в микросървисна архитектура. Описани са всички връзки между отделните области.

Показани са отделни примери под формата на „Управляем формат на обекта (MOF)", представящи различните елементи: Схема, Клас, Свойства, Връзки и Асоциации, Квалификатор.

Резултатът получен от направата на йерархия на областите в микросървисна архитектура са графично представени с помощта на UI Wireframing.

Направен е анализ на постигнатите резултати.

8 VIII ГЛАВА. ПРИЛАГАНЕ НА МОДЕЛИ ЗА СИГУРНОСТ ЧРЕЗ ИЗПОЛЗВАНЕ НА СЪВРЕМЕННИ ТЕХНОЛОГИИ ЗА УПРАВЛЕНИЕ НА МИКРОСЪРВИСИ

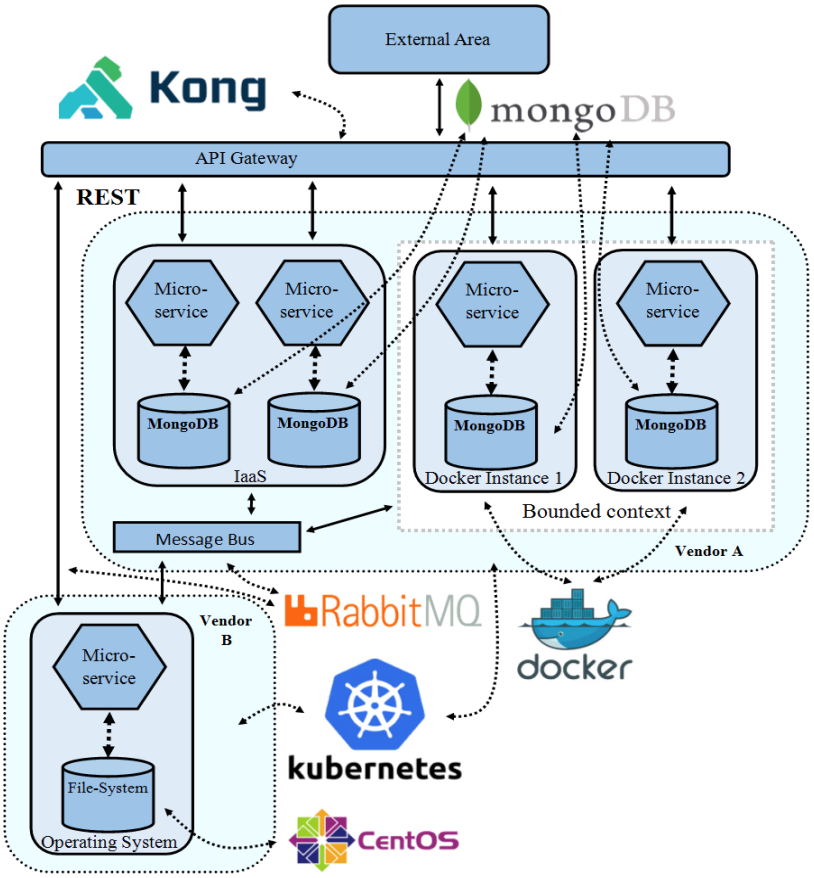
В следващите точки се разглеждат съвременни инструменти за управление на микросървиси, които могат да бъдат конфигурирани следвайки добрите практики на избраните модели за сигурност от Йерархичната таксономия от модели за сигурност описана в глава 7.

8.1 Разработване на платформа за микросървисно приложение използваща съвременни технологии

Платформата е изградена от компоненти с отворен код, като повечето от тях са модифицирани в различна степен според специфичните изисквания, а други са целево разработени за конкретната имплементация, но биха могли да бъдат интегрирани и в други системи.

В глава 2 е разгледана областта „Акаунт и Идентичност (Account and Identity)”. Тя може да бъде управлявана от различни инструменти, които предоставят функционалността „API Gateway”. Продуктът, който се използва в текущия концептуалния модел е Kong [9]. В глава 3 е разгледана областта Комуникация. Тя е разделена на две части „Междусървисна комуникация“ и „Съобщения“ . Инструментът, който може да се използва при осъществяване на комуникация между микросървиси е RabbitMQ [10]. В глава 4 е разгледана областта Съхраняване на данни. Тя основно е адресирана към едно от ограниченията на микросървисната архитектура – всеки микросървис може да оперира с информация, към която само той има достъп. При нужда от получаване на информация съхранявана от друг микросървис, то той трябва да се обърне към микросървисът разполагащ с тези данни, за да се избегне директния достъп. В глава 5 е разгледана областта Микросървисна среда. Тя е отнесена към средата, в която микросървиси могат да функционират. Видовете продукти предлагащи среди от такъв тип са: Docker container [14], CentOS [13] и IaaS [15]. Всяка една от средите разполага с набор от инструменти, които могат да допринесат за повишаване на сигурността. В глава 6 е разгледана областта „Микросървиси разположени върху платформи на различни доставчици“. В нея се разисква възможността бизнес логиката на микросървисно софтуерно приложение да бъде разделена и отделните елементи да бъдат разположени на два или повече доставчика на услуги. Подобен тип клъстеризация може да бъде управлявана с помощта на Kubernetes (K8s) [16].

На Фиг. 18 е представена архитектура на платформа за управление на микросървисни приложения, който представя всички пет области разгледани в предишните глави. Показани са всички инструменти, които дават възможност за успешно прилагане на по-голямата част от моделите за сигурност описани в йерархичната таксономия от модели за сигурност.



Фиг. 18 Архитектура на платформа за управление на микросървисни приложения включваща всички разглеждани области

8.2 Прилагане на модели за сигурност върху платформа изградена от съвременни бизнес решения

В следващите точки се разглеждат примери, в които с помощта на модерни бизнес технологии се прилагат модели за сигурност представени в йерархична таксономия на модели за сигурност от глава 7. Всяка една от технологиите е разгледана в предишната точка 8.1.

8.2.1 Акаунт и Идентичност (Account and Identity)

Бизнес приложението взето под внимание тук е Kong [9]. Важно е да се отбележи, че за да може да се използва Kong, то трябва да бъде инсталирано на всички хостове, върху които се поставят микросървиси. Списъкът с всички модели за сигурност от „Акаунт и Идентичност“ е представен на Фиг. 3.

8.2.2 Комуникация

В тази точка се разглеждат примери, в които се прилагат модели за сигурност представени в йерархична таксономия на модели за сигурност от глава 7 с помощта на модерни бизнес технологии. Инструментът, който може да се използва при осъществяване на комуникация между микросървиси е RabbitMQ [10]. Списъците с всички модели за сигурност от „Комуникация“ са представени на Фиг. 5 и Фиг. 6.

8.2.3 Съхраняване на данни

В точка се разглеждат примери, в които се прилагат модели за сигурност представени в йерархична таксономия на модели за сигурност от глава 7 с помощта на модерни бизнес технологии. Инструментите, които биха могли да бъдат използвани при съхраняване на данни са: база от данни MongoDB [11] и файлова система ext4 [12] предоставена от операционна система CentOS [13]. Списъкът с всички модели за сигурност от „Съхраняване на данни“ е представен на Фиг. 8.

8.2.4 Микросървисна среда

В точка се разглеждат примери, в които се прилагат модели за сигурност представени в йерархична таксономия на модели за сигурност от глава 7 с помощта на модерни бизнес технологии. За текущите цели са използвани инструменти като: контейнер Docker [14] и операционна система CentOS [13]. Списъкът с всички модели за сигурност от „Микросървисна среда“ са представени на Фиг. 10.

8.2.5 Микросървиси разпределени върху платформи на различни доставчици

В точка се разглеждат примери, в които се прилагат модели за сигурност представени в йерархична таксономия на модели за сигурност от глава 7 с помощта на модерни бизнес технологии. За текущите цели е изследван инструментът Kubernetes [25]. Той се използва главно за управление на клъстери състоящ се от Docker инстанции. Списъкът с всички модели за сигурност от „ Микросървиси разпределени върху платформи на различни доставчици “ е представен на Фиг. 13.

Заключение

Направено е проучване на съвременните технологии за имплементация, на база на което е предложена архитектура на платформа за управление на микросървисни приложения. Тя цели да покрие всички области дефинирани в глави 2, 3, 4, 5 и 6.

Представени са примерни тестови сценарии и решения за почти всеки от избраните модели за сигурност.

Представени са решения относно избраните модели за сигурност в област „Акаунт и Идентичност“ с помощта на продуктът Kong и неговата основна функционалност API Gateway.

Представени са решения относно избраните модели за сигурност в област „Комуникация“ с помощта на RabbitMQ. Той осигурява автономност на отделните микросървиси чрез предоставяне на канал, към който микросървиси да изпращат и получават съобщения.

Предоставени са решения относно избраните модели за сигурност в област „Съхраняване на данни“ с помощта на база от данни MongoDB и файлова система ext4 предоставена от операционна система CentOS.

Предоставени са решения относно избраните модели за сигурност в област „Микросървисна среда“ с помощта на набор от продукти: Docker container, CentOS и IaaS.

Предоставени са решения относно избраните модели за сигурност в област „Микросървиси разположени върху платформи на различни доставчици“ с помощта на продукт предоставящ способи за клъстеризиране на микросървисно приложение - Kubernetes (K8s).

9 ЗАКЛЮЧЕНИЕ

В настоящата разработка е предложено решение за повишаване на сигурността в информационни системи базирани на микросървисна архитектура.

Разработената йерархична таксономия на модели за сигурност осигуряваща лесен и удобен начин за намиране на подходящият модел за сигурност или съвкупност от модели за сигурност

Таксономията, представена с помощта на обектно ориентиран език за моделиране, позволява лесно присъединяване на нови модели за сигурност.

Разработеният език позволява графично представяне на различните релации както и на моделите за сигурност с помощта на съвременни графични интерфейси.

Разгледани са съвременни технологии, които могат да бъдат използвани в изграждането и управлението на цялостно софтуерно микросървисно приложение.

Предложена е архитектура на платформа за управление на микросървисни приложения, както и решения за имплементиране на почти всички модели за сигурност описани в йерархична таксономия от модели за сигурност чрез използване модерни бизнес инструменти.

10 ПРИНОСИ

- Извършено е изследване и анализ на архитектурите базирани на микросървиси с цел повишаване на сигурността.
- Предложен е концептуален модел, прилагащ микросървисна архитектура, с помощта на който са дефинирани уязвими области.
- Извършен е анализ на заплахите върху дефинираните уязвими области, като за всяка са предложени съответни модели за сигурност и обосновката на решенията.
- Разработен е йерархичен модел и е представена йерархична таксономия от модели за сигурност с помощта на обектно ориентирано моделиране.
- Разработен е графичен интерфейс, който онагледява връзките между уязвимите области в микросървисните архитектура и избраните модели за сигурност.
- Представена е архитектура на платформа имплементираща предложените модели, чрез използване на съвременни технологии за управление на микросървиси.

11 ИЗПОЛЗВАНА ЛИТЕРАТУРА

[1] C. Richardson и F. Smith, MICROSERVICES From Design to Deployment, NGINX, 2016.

[2] M. Weiss и H. Mouratidis, „Selecting Security Patterns that Fulfill Security Requirements,“ *Proceedings of the 16th IEEE International Conference on Requirements Engineering*, pp. 169-172, 2008.

[3] E. Gamma, J. Vlissides, R. Johnson и R. Helm, Design Patterns: Elements of Reusable Object-Oriented Software, 1994.

[4] „Understanding software design patterns,“ [Онлайн]. Available: https://opensource.com/article/19/7/understanding-software-design-patterns.

[5] „Design Pattern - Overview,“ [Онлайн]. Available: https://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm.

[6] A. Shostack, THREAT MODELING: Designing for Security 1st Edition, John Wiley & Sons, Inc., 2014.

[7] „DMTF Releases CIM 2.52,“ [Онлайн]. Available: https://www.dmtf.org/content/dmtf-releases-cim-252.

[8] „Balsamiq Cloud,“ [Онлайн]. Available: https://balsamiq.cloud/.

[9] „Kong Gateway,“ [Онлайн]. Available: https://konghq.com/kong.

[10] „RabbitMQ,“ [Онлайн]. Available: https://www.rabbitmq.com/.

[11] „MongoDB,“ [Онлайн]. Available: https://www.mongodb.com/.

[12] „An introduction to Linux's EXT4 filesystem,“ [Онлайн]. Available: https://opensource.com/article/17/5/introduction-ext4-filesystem.

[13] „CentOS Project,“ [Онлайн]. Available: https://www.centos.org/.

[14] „Docker Homepage,“ [Онлайн]. Available: https://www.docker.com/.

[15] E. B. Fernandez, H. Washizaki и N. Yoshioka, „Patterns for Secure Cloud IaaS (Infrastructure as a Service),“ в *Asian Pattern Languages of Programs (PLoP) Conference*, 2016.

[16] „Kubernetes (K8s),“ [Онлайн]. Available: https://kubernetes.io/.

[17] T. Tenev и D. Birov, „Security Patterns for Microservice Account and Identity,“ в *15th International Conference on Informatics and Information Technologies*, Mavrovo, 2018.

[18] T. Tenev и D. Birov, „Security Patterns for Microservice Communication,“ в *Четиридесет и седма пролетна конференция на Съюза на математиците в България*, Borovets, 2018.

[19] T. Tenev, „SECURITY PATTERNS FOR MICROSERVICE DATA MANAGEMENT,“ в *Doctoral Conference: Young Scientists*, Sofia, 2018.

[20] C. Posta, Microservices for Java Developers, 2016.

[21] T. Tenev и S. Tsvetanov, „Enhancing security in Microservice environments,“ в *9th Balkan Conference in Informatics*, Sofia, 2019.

[22] T. Tenev и D. Birov, „SECURITY PATTERNS FOR MICROSERVICES LOCATED ON DIFFERENT VENDORS,“ в *VII International Conference on Engineering, Technologies and Systems TECHSYS*, Plovdiv, 2018.

[23] [Онлайн]. Available: https://www.dmtf.org/sites/default/files/standards/documents/DSP0221_3.0.0.pdf.

[24] P. Mell и T. Grance, „The NIST Definition of Cloud Computing,“ 2011. [Онлайн]. Available: https://csrc.nist.gov/publications/detail/sp/800-145/final.

[25] „Kubernetes,“ [Онлайн]. Available: https://kubernetes.io/docs/tasks/access-application-cluster/configure-access-multiple-clusters/.

[26] O. Ajaj и E. B. Fernandez, „A pattern for the WS-Trust standard for web services,“ в *in Proceedings of the Asian Conference on Pattern Languages of Programs*, 2010.

СПИСЪК С ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИЯТА

1. Tihomir Tenev, Dimitar Birov, Security Patterns for Microservice Account and Identity, In proceedings of 15th International Conference on Informatics and Information Technologies, 2018, pages:124-128, ISBN:978-608-4699-08-8,

2. Tihomir Tenev, Dimitar Birov, Security Patterns for Microservice Communication, Доклади на Четиридесет и седма пролетна конференция на Съюза на математиците в България, 2018, ISSN (online):1313-3330

3. Tihomir Tenev, SECURITY PATTERNS FOR MICROSERVICE DATA MANAGEMENT, In proceedings of Doctoral Conference: Young Scientists, 2018, pages:575-581, ISBN:978-954-07-4611-1

4. Tihomir Tenev, Dimitar Birov, SECURITY PATTERNS FOR MICROSERVICES LOCATED ON DIFFERENT VENDORS, VII International Conference on Engineering, Technologies and Systems TECHSYS 2018, Technical University – Sofia, Plovdiv, 2018, pages:130-133, ISSN (online):2535-0048

5. Tihomir Tenev, Simeon Tsvetanov, Enhancing security in Microservice environments, 9th Balkan Conference in Informatics, ISec2019 Workshop, 2019