ИСПОЛЬЗОВАНИЕ МИКРОЯДЕРНЫХ СРЕДСТВ ВИРТУАЛИЗАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СИСТЕМ С МИКРОСЕРВИСНОЙ АРХИТЕКТУРОЙ

Москвичев А. Д.[1], Долгачев М. В.[2]

Цель статьи: *увеличение отказоустойчивости, уровня безопасности и упрощения процесса обновления систем с микросервисной архитектурой с помощью средств виртуализации.*

Метод: *использование микроядерных средств виртуализации, образы которых называются Unikernel. Unikernel – это образы виртуальных машин, содержащие единственное приложение. Каждый микросервис системы запускается в образе Unikernel, иных процессов и служб в образе не содержится.*

Полученный результат: *дано определение Unikernel, произведено сравнение с существующими системами виртуализации. Перечислены существующие проекты, реализующие технологию Unikernel. Даны определения библиотечной операционной системы, микроядра. Дано определение Unikernel как технологии, объединяющей библиотечные операционные системы и микроядра. Перечислены основные преимущества использования Unikernel: обеспечение безопасности информационной системы и простота обновлений. Разработано программное средство для тестирования образов Unikernel на возможность проникновения в информационную систему в случае наличия в программном средстве уязвимости типа «удаленное выполнение произвольного кода». Произведено тестирование образа Unikernel на возможность исполнения произвольного кода внутри образа. В заключении приведены оценки об эффективности применения технологии Unikernel для построения систем с микросервисной архитектурой, в частности для построения SIEM-систем.*

Ключевые слова: *Unikernel, операционные системы, облачные вычисления, безопасность, уязвимость, SIEM.*

DOI:10.21681/2311-3456-2022-1-87-94

1. Введение

Существует несколько видов виртуализации сервисов. Одним из наиболее распространенных сегодня методов является использование виртуальных машин, размещенных на гипервизорах, таких как VMware ESXi [1].

Гипервизоры позволяют размещать несколько гостевых операционных систем на одной физической машине. Эти операционные системы выполняются в так называемых виртуальных машинах. Широкое использование гипервизоров обусловлено их способностью лучше распределять и оптимизировать рабочую нагрузку на физических серверах в отличие от устаревших инфраструктур по одной операционной системе на физическом сервере [2].

Контейнеры – это еще один метод виртуализации, который отличается от гипервизоров созданием виртуализированных сред и совместным использованием ядра хоста. Это обеспечивает меньшие затраты ресурсов в сравнении с гипервизорами [3].

Unikernel – это образы виртуальных машин, содержащие единственное приложение. Одной из ключевых особенностей является отсутствие отдельно пользовательского адресного пространства и пространства ядра [4]. Несмотря на свой относительно молодой возраст, идея Unikernel заимствуется у микроядерных операционных систем.

Системы Unikernel используют абстрактный гипервизор в дополнение к использованию библиотечных операционных систем для включения в приложение только требуемых подпрограмм ядра для представления самого легкого из всех трех решений.

Виртуальные машины представляют гораздо большую нагрузку на инфраструктуру в отличие от контейнеров и Unikernel.

В таблице 1 приведены преимущества и недостатки существующих технологий виртуализации [5, 6, 7].

Исходя из данных таблицы 1, для разработки систем с микросервисной архитектурой самым удачным решением будет использование микроядерной виртуализации, так как образы Unikernel являются самыми легковесными и безопасными.

В таблице 2 перечислены существующие проекты, использующие архитектуру Unikernel.

2. Микроядро. Сравнение монолитного ядра и микроядра

В отличие от монолитных ядер, которые содержат большое количество кода, что делает их довольно

1 Москвичев Антон Дмитриевич, аспирант, ФГБОУ ВО «Тихоокеанский государственный университет», г. Хабаровск, Россия. E-mail: anton.moskvichev.1996@yandex.ru. Moskvichev Anton Dmitrievich, postgraduate, Pacific National University. ORCID: 0000-0001-6532-2463

2 Долгачев Михаил Владимирович, кандидат технических наук, доцент, ФГБОУ ВО «Тихоокеанский государственный университет», г. Хабаровск, Россия. E-mail: 007428@pnu.edu.ru. Dolgachev Mihail Vladimirovich, Ph. D. (in Tech.), Pacific National University. ORCID: 0000-0003-1520-800X

Таблица 1

Преимущества и недостатки существующих технологий виртуализации

Средства виртуализации	Преимущества	Недостатки
Виртуальные машины	Позволяют развертывать разные операционные системы на одном узле	Требуются вычислительные мощности, пропорциональные количеству экземпляров
	Полная изоляция от узла	Требуется большая инфраструктура
	Доступен инструмент управления виртуальной средой	Каждый экземпляр загружает всю операционную систему
Контейнеры Linux	Легкая виртуализация	Уменьшение изоляции из-за общего ядра
	Быстрое время загрузки	Меньше гибкости (то есть зависит от ядра хоста)
	Доступен инструмент управления виртуальной средой	Сеть менее гибкая
	Динамическое распределение ресурсов	
Образы Unikernel	Легкие образы	Плохо развит
	Специализированное применение	Требуется разработка приложений с нуля
	Полная изоляция от хоста	Ограниченные возможности развертывания
	Повышенная защита от отсутствующих функций (например, удаленное выполнение команд)	Статическое выделение ресурсов
		Нет инструмента управления виртуальной средой

Таблица 2

Существующие решения Unikernel

Unikernel	Доступные языки программирования	Гипервизор
ClickOS	C++	Xen
HalVM	Haskell	Xen
IncludeOS	C++	KVM, VirtualBox, ESXi, Google Cloud
MirageOS	OCaml	KVM, Xen, RTOS/MCU
Nanos Unikernel	C, C++, Go, Java, Node.js, Python, Rust, Ruby	QEMU/KVM
OSv	Java, C, C++, Node, Ruby	VirtualBox, ESXi, KVM, Amazon EC2
Rumprun	C, C++, Erlan, Go, Java, Node.js, Python	Xen, KVM
Unik	Go, Node.js, Java, C, C++, Python, OCaml	VirtualBox, ESXi, KVM, XEN и другие
ToroKernel	FreePascal	VirtualBox, KVM, XEN, HyperV

большими, микроядра имеют меньший размер. Микроядра уменьшают объем кода в пространстве ядра в пользу модулей, выполняемых в пространстве пользователя [8].

Вторым преимуществом микроядра является надежность. Чем больше кода, тем выше вероятность возникновения ошибок, а также потенциальных недостатков безопасности в ядре. Сохраняя небольшой размер ядра, микроядра снижают риск ошибок и уязвимостей в ядре, которые могут оказаться фатальными для работы системы.

В настоящее время монолитные ядра используются для обеспечения единой версии операционной системы, которая потенциально может выполнять любую

требуемую функцию. Windows и Linux являются яркими примерами. Поскольку неизвестно, что пользователи собираются делать с операционной системой, ядро интегрирует как можно больше функций из коробки (например, общение в сети, доступ к файлам на жестком диске, запуск нескольких сервисов и т.д.) [9].

В случае использования микроядра в широко используемых операционных системах, рассматриваемая операционная система будет очень маленькой. Но каждому пользователю придется устанавливать различные модули в зависимости от того, что они хотят сделать, потому что операционная система микроядра включает в себя минимум из коробки. Любая дополнительная функция требует модуля, который выполняется в пространстве пользователя и взаимодействует с базовым микроядром.

Хотя микроядра не удобны для пользователей, в сравнении с монолитными, они полезны в тех областях, где требуется надежность. Поскольку модули работают в пользовательском пространстве отдельно от ядра и пользовательского пространства других модулей, проблема в одном модуле не может повлиять на другой модуль.

Например, в монолитном ядре функции управления файлами напрямую интегрированы в ядро. Если произойдет сбой управления файлами, это может повлиять на все ядро, что приведет к сбою всей системы (например, Microsoft Blue Screen of Death). Если другое приложение будет запущено на том же компьютере, эта служба будет зависеть от сбоя, даже если она не имеет отношения к функции управления файлами.

В реализации микроядра, использующей тот же вариант использования для доступа к файлам на диске, в текущей операционной системе микроядра должен быть загружен соответствующий модуль. То же самое касается предоставления сервисов в сети, требуется другой модуль и его необходимо загрузить. Однако, если произойдет сбой в работе модуля управления файлами, работая в своем собственном пользовательском пространстве, ядро не будет затронуто, и система будет работать. Кроме того, сетевой модуль также не будет затронут, поскольку он также выполняется в своем собственном пользовательском пространстве, отдельно от модуля управления файлами.

3. Библиотечные операционные системы

Библиотечная операционная система – другой метод построения операционной системы, где ядро и модули, требуемые приложением, выполняются в том же адресном пространстве, что и само приложение. В отличие от микроядра и монолитного ядра, это подразумевает отсутствие разделения между ядром и пространством пользователя, и приложение имеет прямой доступ к функциям уровня ядра, не требуя системных вызовов [10].

Целью библиотечной операционной системы является обеспечение расширения за счет раскрытия низкоуровневых аппаратных абстракций. К сожалению, проблема между абстракцией низкоуровневых аппаратных средств заключается в сложности поддержки большого количества аппаратных средств. Это означает, что для создания полной библиотечной операционной системы ядро должно быть скомпилировано с драйверами устройств, написанными для конкретного оборудования, что приводит к плохой переносимости библиотечных операционных систем.

В настоящее время виртуализация обеспечивает абстракцию основного оборудования за счет использования драйверов виртуализированного оборудования. Это позволяет библиотечным операционным системам поддерживать общий виртуальный драйвер, а не пытаться поддерживать различные аппаратные средства, что дает основу для создания однотипных приложений, объединяя уже протестированную технологию виртуализации с библиотечными операционными системами, загруженными драйверами гипервизора для полной переносимости в организованной среде.

4. Технология Unikernel

Технология микроядерной виртуализации объединяет в себе понятия микроядра и библиотечной операционной системы. Образы Unikernel содержат единственное программное средство. Образы Unikernel также являются библиотечными операционными системами. Такой подход позволяет включать в образ только необходимые для работы программного средства функции. В результате образы Unikernel предоставляют собой небольшие, легкие и высокоэффективные виртуализированные приложения [11].

В традиционных операционных системах реализовано два адресных пространства: пространство ядра и пространство пользователя. Пространство ядра содержит функционал самой операционной системы. Пользовательское адресное пространство содержит код программного средства (приложения).

Так как программному средству необходим доступ к функционалу операционной системы, то существует зависимость кода программного средства от кода ядра. Такой подход эффективен в том случае, когда нет информации о программных средствах, которые будут работать в данной операционной системе. Поэтому монолитное ядро становится громоздким, так как пытается предоставить широкий спектр сервисов.

Образы Unikernel представляют собой совершенно другую структуру. Приложение, работающее в одноуровневой системе, не представляет никакого разделения в своем адресном пространстве [12].

Для создания образа Unikernel используют средства кросс-компиляции, подключая необходимые для работы программного средства низкоуровневые функции из библиотечной операционной системы (предоставляется в компилируемой форме). В результате получается образ, который может работать изолированно для предоставления услуги программным средством.

4.1. Безопасность в Unikernel

Образы Unikernel обеспечивают большую безопасность в сравнении с виртуальными машинами и кон-

тейнерами. Это связано с уменьшением поверхности атаки и уменьшением кода операционной системы. Решения как для виртуальных машин, так и для контейнеров содержат больше инструментов, чем требуется для работающего приложения. Это значительно увеличивает поверхность атаки [13].

В качестве примера можно взять службу разрешения имен DNS. Единственная цель такого сервиса – принять запрос разрешения имени, выполнить поиск в базе данных и отправить ответ. В виртуализированном или контейнерном случае среда Linux или Windows, предоставляющая службу DNS, включает в себя гораздо больше, чем просто сетевой стек и локальную базу данных. Атака может быть предпринята на функции удаленного доступа, механизмы аутентификации пользователя, некорректный драйвер и другие. Все эти атаки имеют общую цель, а именно получить доступ к интерфейсу командной строки для удаленного выполнения вредоносного кода.

Принимая тот же вариант использования в образе Unikernel, требуемые библиотечные процедуры ограничены доступом к сетевому устройству. Даже доступ к базе данных может быть включен в код приложения. В отличие от виртуальной машины и контейнера, уникальное ядро не содержит ненужных функций операционной системы, таких как управление устройствами, удаленный доступ, механизмы аутентификации или даже интерфейс командной строки [14].

Одна и та же служба DNS может принимать только запросы DNS и отправлять ответы DNS. Злоумышленник не может использовать функции удаленного доступа, аутентификации или удаленного выполнения кода, просто потому, что необходимые для этого функции отсутствуют в коде цели.

4.2 Неизменяемость приложения

Неизменяемость приложения означает, что после запуска приложения его нельзя изменить. Когда упомянутое приложение нуждается в изменениях или обновлениях, а не в применении многочисленных изменений и исправлений, приложение удаляется, и загружается новая версия [15].

Преимущество применения подхода такого подхода состоит в том, что приложение остается легким, а не усложняется «заплатками» в течение многих лет. Идеальный пример тому – операционная система Windows. Хотя первая установка может быть не слишком громоздкой, с годами бесчисленные обновления и патчи со временем увеличат её размер, увеличат сложность кода и, возможно, добавят новые ошибки и уязвимости.

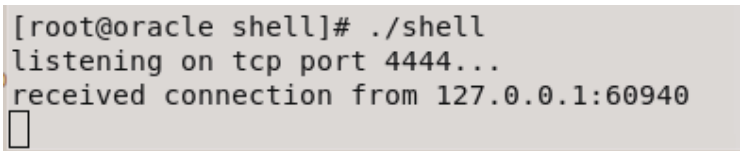
Так как образы Unikernel предназначены для разработки и развертывания, без возможности удалённого подключения к ней, образы неизменны по конструкции. Кроме того, их быстрое время загрузки обеспечивает возможность обновлений без прерывания работы.

5. Пример использования Unikernel

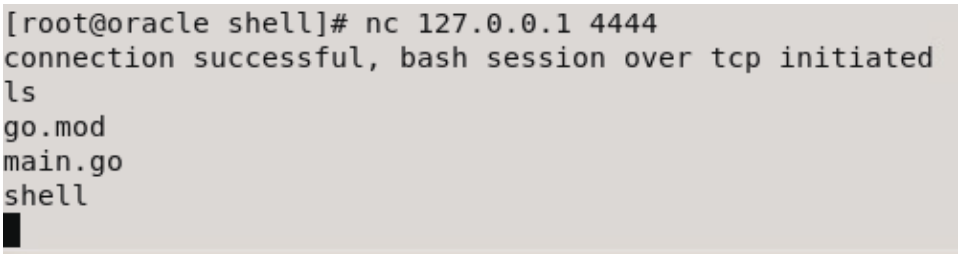
Пусть имеется программное средство, написанное на компилируемом языке программирования Go [16], представляющее собой «bind shell» – удаленная консоль, в которой в роли серверной части выступает удаленная машина (далее – ПС) [17]. Работа с ПС осуществляется по алгоритму (рис.1):

1. На сервере запускается ПС. ПС открывает доступ по порту 4444 для приема сообщений;
2. Клиент подключается к порту 4444 сервера с помощью утилиты «netcat». Netcat – утилита Unix, позволяющая устанавливать соединения TCP и UDP, принимать оттуда данные и передавать их [18];
3. Клиент передает в качестве сообщений команды операционной системе;
4. ПС исполняет команды, полученные от клиента, на стороне сервера;

Согласно концепции микроядра, в случае запуска



а) Серверная часть



б) Клиентская часть

Рис. 1. Пример работы ПС

```
[root@oracle shell]# ops run -p 4444 shell
 100% |
booting /root/.ops/images/shell.img ...
en1: assigned 10.0.2.15
listening on tcp port 4444...
en1: assigned FE80::98A1:C4FF:FE78:391D
received connection from 10.0.2.2:60948
```

а) Серверная часть

```
[root@oracle shell]# nc 127.0.0.1 4444
connection successful, bash session over tcp initiated
ls
```

б) Клиентская часть

Рис. 2. Пример работы ПС из образа Unikernel

ПС в образе Unikernel не должно последовать исполнение команд, так как образ не содержит командную оболочку. Для проверки гипотезы запустим программное средство в образе Unikernel, проект «Nanos Unikernel» (рис. 2).

Подключение клиента к ПС удалось, однако исполнение кода не последовало. Это означает, что исполнение кода в образе Unikernel невозможно. Следовательно, нет риска компрометации всей информационной системы в случае, если программное средство имеет уязвимость типа удаленное исполнение произвольного кода (remote command execution – RCE) и запущено в образе Unikernel.

6. Практическое применение

Использование Unikernel можно рассмотреть на примере архитектуры системы управления информацией о безопасности и событиями безопасности (далее SIEM – «Security informationand event management»).

Любая SIEM-система имеет следующие модули [19]:

1. модуль нормализации – модуль, преобразующий записи из журналов событий в единый формат;
2. модуль корреляций – модуль, выявляющий инциденты информационной безопасности с помощью правил корреляций;
3. модуль хранения событий;
4. модуль управления SIEM-системой.

Все перечисленные выше модули работают независимо друг от друга и связываются через шину данных. Следовательно, каждый из этих модулей удобно реализовать как микросервис, помещенный в контейнер Unikernel. Такой подход дает следующие преимущества [20]:

1. Увеличение уровня защищенности SIEM-системы за счет уменьшения площади атаки. Атака на SIEM-систему эквивалентна атаке на один из модулей. В случае успеха скомпрометированным будет считаться только контейнер Unikernel, содержащий атакованный модуль.
2. Увеличение отказоустойчивости. В случае отказа одного из модулей, неработоспособным останется только один контейнер. SIEM-система будет функционировать в полном объеме.
3. Простота обновления. Достаточно заменить образы Unikernel для обновляемых модулей, чтобы установить последние обновления.

7. Выводы

В результате проделанной работы показано использование технологии микроядерной виртуализации, приведен пример реализации и практического применения.

Unikernel – новая технология, но принципы, на которых они основаны, заложены в эпоху зарождения операционных систем. Эти же принципы были повторно использованы и адаптированы к новым технологиям, имеющимся сегодня для расширения их возможностей.

Технология Unikernel может найти широкое применения в системах с большим количеством модулей для предотвращения проблем с зависимостями, а также упрощения развертывания.

DOI:10.21681/2311-3456-2022-1-87-94

Литература

1. Рак, И. П. Технологии облачных вычислений: учебное пособие : [16+] / И. П. Рак, А. В. Платёнкин, Э. В. Сысоев ; Тамбовский государственный технический университет. – Тамбов : Тамбовский государственный технический университет (ТГТУ), 2017. – 82 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=499410 (дата обращения: 16.11.2021). – Библиогр.: с. 79. – ISBN 978-5-8265-1826-7. – Текст : электронный.
2. Системный администратор / изд. ООО «Синдикат 13» ; гл. ред. Г. Положевец. – Москва : Синдикат 13, 2017. – № 5(174). – 100 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=459117. – ISSN 1813-5579. – Текст : электронный.
3. Koller R., Williams D. Will serverless end the dominance of linux in the cloud? [Conference] // ACM/SIGOPS HotOS. - Whistler : [s.n.], 2017. pp. 169–173 DOI: 10.1145/3102980.3103008
4. Dan Williams, Ricardo Koller, Martin Lucina, Nikhil Prakash Unikernels as Processes [Conference] // Proceedings of the ACM Symposium on Cloud Computing: International Conference on Management of Data. - New York: ACM, 2018. - pp. 199-211. DOI: 10.1145/3267809.3267845
5. Беспалов, Д. А. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения: / Д. А. Беспалов, С. М. Гушанский, Н. М. Коробейникова; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – Часть 2. – 169 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=577699 (дата обращения: 16.11.2021). – Библиогр. в кн. – ISBN 978-5-9275-3368-8. – Текст : электронный.
6. Poulton Nigel Docker Deep Dive [Book]. - [s.l.] : Packt Publishing, 2020. – 249 p.
7. Kundan Ajit Pratap Intelligent Automation with VMware [Book]: Packt Publishing, 2019. – 344 p.
8. Dragoni N., Giallorenzo S., Lafuente A. L., Mazzara M., Montesi F., Mustafin R., Safina L Microservices: Yesterday, Today, and Tomorrow [Conference] // Springer International Publishing. - [s.l.] : Cham, 2017. - pp. 195–216.
9. Турулин, И. И. Виртуальные машины, операционные системы и приложения / И. И. Турулин, В. Г. Галалу, А. В. Дагаев ; Таганрогский институт им. А. П. Чехова (филиал) РГЭУ (РИНХ). – Таганрог : Таганрогский институт имени А. П. Чехова, 2015. – 64 с. : ил., граф. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=614532 (дата обращения: 16.11.2021). – Библиогр. в кн. – ISBN 978-5-87976-951-7. – Текст : электронный.
10. Басыня, Е. А. Системное администрирование и информационная безопасность / Е. А. Басыня. – Новосибирск: Новосибирский государственный технический университет, 2018. – 79 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=575325 (дата обращения: 16.11.2021). – Библиогр. в кн. – ISBN 978-5-7782-3484-0. – Текст : электронный.
11. Watada Junzo, Roy Arunava, Kadikar Ruturaj, Pham Hoang, Xu Bing Emerging Trends, Techniques and Open Issues of Containerization: A Review [Conference] // IEEE Access.[s.n.], 2017. –vol. 7 pp. 152443 - 152472 DOI: 10.1109/ACCESS.2019.2945930
12. Bruno Xavier, Tiago Ferreto, Luis Jersak Time Provisioning Evaluation of KVM, Docker and Unikernels in a Cloud Platform [Conference] // 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2016. DOI: 10.1109/CCGrid.2016.86
13. Kocher P., Horn J., Fogh A., , Genkin D., Gruss D., Haas W., Hamburg M., Lipp M., Mangard S., Prescher T., Schwarz M., Yarom Y. Spectre attacks: Exploiting speculative execution [Conference] // IEEE Security and Privacy. - San Francisco : [s.n.], 2019. DOI: 10.1109/SP.2019.00002
14. Li Y., Dolan-Gavitt B., Weber S., Cappos J. Lock-in-Pop: Securing privileged operating system kernels by keeping on the beaten path [Conference] // USENIX Annual Technical Conf. - Santa Clara : [s.n.], 2017. pp. 1-13
15. Odun-Ayo Isaac, Geteloma Victor, Eweoya Ibukun, Ahuja Ravin Virtualization, Containerization, Composition, and Orchestration of Cloud Computing Services [Conference]. - California : Computational Science and Its Applications – ICCSA, 2019. pp. 403–417 DOI: 10.1007/978-3-030-24305-0_30
16. Батчер М. Go на практике / Мэтт Батчер, Мэтт Фарина ; пер. с англ. Р. Н. Рагимова; науч. ред. А. Н. Киселев. – М.: ДМК Пресс, 2017. – 374 с.
17. Бирюков, А. А. Информационная безопасность: защита и нападение [Текст] / А. А. Бирюков. – 2-е изд., перераб. и доп. – М. : ДМК Пресс, 2017. – 434 с. : ил. – ISBN 978-5-97060-435-9.
18. Michael Kurth, Ben Gras, Dennis Andriesse, Cristiano Giuffrida, Herbert Bos, Kaveh Razavi NetCAT: Practical cache attacks from the network [Conference] // IEEE Symposium on Security and Privacy (SP). - San Francisco : Institute of Electrical and Electronics Engineers Inc., 2020. pp. 20-38 DOI: 10.1109/SP40000.2020.00082
19. Kai-Oliver Detken, Marcel Jahnke, Carsten Kleiner, Marius Rohde, Combining Network Access Control (NAC) and SIEM functionality based on open source [Conference]. - Bucharest, Romania : 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017. DOI: 10.1109/IDAACS.2017.8095094
20. Jain V., Qi S.,Ramakrishnan K.K. Fast Function Instantiation with Alternate Virtualization Approaches [Conference]. - California : IEEE Computer Society, 2021. - Vols. 2021-July. DOI: 10.1109/LANMAN52105.2021.9478808

USING MICROKERNEL VIRTUALIZATION MEANS TO ENSURE THE SECURITY OF SYSTEMS WITH MICROSERVICE ARCHITECTURE

Moskvichev A. D.[3], Dolgachev M. V.[4]

Purpose of the article: *to increase fault tolerance, security level and simplify the process of updating systems using a micro service architecture using virtualization tools.*

Method: *Using microkernel virtualization tools called Unikernel images. Unikernel are virtual machine images containing a single application. Each micro service of the system runs in the Unikernel image, the processes and services used are not contained in the image.*

The result: *a definition of Unikernel is given; a comparison is made with existing virtualization systems. The mechanisms that implement the Unikernel technology are listed. The definitions of the library operating system and microkernel are given. The definition of Unikernel is given as a technology that combines library operating systems and microkernels. The main advantages of using Unikernel are listed: ensuring the security of information systems and ease of updates. A developed software tool for testing Unikernel images for the possibility of penetrating an information system in the event of a vulnerability of the "remote execution of arbitrary code" type in the software tool. The image was tested. In conclusion, the assessment of the effectiveness of the use of Unikernel technology for building systems with a micro service architecture, in particular for building SIEM systems.*

Keywords: *Unikernel, virtualization, operating systems, cloud computing, security, vulnerability, SIEM.*

References

1. Rak, I. P. Tekhnologii oblachnyh vychislenij : uchebnoe posobie : [16+] / I. P. Rak, A. V. Platyonkin, E. V. Sysoev ; Tambovskij gosudarstvennyj tekhnicheskij universitet. – Tambov : Tambovskij gosudarstvennyj tekhnicheskij universitet (TGTU), 2017. – 82 s. : il. – Rezhim dostupa: po podpiske. – URL: https://biblioclub.ru/index.php?page=book&id=499410 (data obrashcheniya: 16.11.2021). – Bibliogr.: s. 79. – ISBN 978-5-8265-1826-7. – Tekst : elektronnyj.
2. Sistemnyj administrator / izd. OOO «Sindikat 13» ; gl. red. G. Polozhevec. – Moskva : Sindikat 13, 2017. – № 5(174). – 100 s. : il. – Rezhim dostupa: po podpiske. – URL: https://biblioclub.ru/index.php?page=book&id=459117. – ISSN 1813-5579. – Tekst : elektronnyj.
3. Koller R., Williams D. Will serverless end the dominance of linux in the cloud? [Conference] // ACM/SIGOPS HotOS. - Whistler : [s.n.], 2017. DOI: 10.1145/3102980.3103008
4. Dan Williams, Ricardo Koller, Martin Lucina, Nikhil Prakash Unikernels as Processes [Conference] // Proceedings of the ACM Symposium on Cloud Computing: International Conference on Management of Data. - New York : ACM, 2018. - pp. 199-211. DOI: 10.1145/3267809.3267845
5. Bespalov, D. A. Operacionnye sistemy real'nogo vremeni i tekhnologii razrabotki krossplatformennogo programmnogo obespecheniya: uchebnoe posobie : [16+] / D. A. Bespalov, S. M. Gushanskij, N. M. Korobejnikova ; YUzhnyj federal'nyj universitet. – Rostov-na-Donu; Taganrog : YUzhnyj federal'nyj universitet, 2019. – CHast' 2. – 169 s. : il. – Rezhim dostupa: po podpiske. – URL: https://biblioclub.ru/index.php?page=book&id=577699 (data obrashcheniya: 16.11.2021). – Bibliogr. v kn. – ISBN 978-5-9275-3368-8. – Tekst : elektronnyj.
6. Poulton Nigel Docker Deep Dive [Book]. - [s.l.] : Packt Publishing, 2020. – 249 p.
7. Kundan Ajit Pratap Intelligent Automation with VMware [Book]. - [s.l.] : Packt Publishing, 2019. – 344 p.
8. Dragoni N., Giallorenzo S., Lafuente A. L., Mazzara M., Montesi F., Mustafin R., Safina L Microservices: Yesterday, Today, and Tomorrow [Conference] // Springer International Publishing. - [s.l.] : Cham, 2017. - pp. 195–216.
9. Turulin, I. I. Virtual'nye mashiny, operacionnye sistemy i prilozheniya : uchebnoe posobie / I. I. Turulin, V. G. Galalu, A. V. Dagaev ; Taganrogskij institut im. A. P. CHekhova (filial) RGEU (RINH). – Taganrog : Taganrogskij institut imeni A. P. CHekhova, 2015. – 64 s. : il., graf. – Rezhim dostupa: po podpiske. – URL: https://biblioclub.ru/index.php?page=book&id=614532 (data obrashcheniya: 16.11.2021). – Bibliogr. v kn. – ISBN 978-5-87976-951-7. – Tekst : elektronnyj.
10. Basynya, E. A. Sistemnoe administrirovanie i informacionnaya bezopasnost' : uchebnoe posobie : [16+] / E. A. Basynya. – Novosibirsk : Novosibirskij gosudarstvennyj tekhnicheskij universitet, 2018. – 79 s. : il. – Rezhim dostupa: po podpiske. – URL: https://biblioclub.ru/index.php?page=book&id=575325 (data obrashcheniya: 16.11.2021). – Bibliogr. v kn. – ISBN 978-5-7782-3484-0. – Tekst : elektronnyj.

3 Anton Moskvichev, postgraduate, Pacific National University, Khabarovsk, Russia. E-mail: anton.moskvichev.1996@yandex.ru. ORCID: 0000-0001-6532-2463

4 Mihail Dolgachev, Ph.D. (in Tech.), Pacific National University, Khabarovsk, Russia. E-mail: 007428@pnu.edu.ru. ORCID: 0000-0003-1520-800X

11. Watada Junzo, Roy Arunava, Kadikar Ruturaj, Pham Hoang, Xu Bing Emerging Trends, Techniques and Open Issues of Containerization: A Review [Conference] // IEEE Access.[s.n.], 2017. –vol. 7 pp. 152443 - 152472 DOI: 10.1109/ACCESS.2019.2945930
12. Bruno Xavier, Tiago Ferreto, Luis Jersak Time Provisioning Evaluation of KVM, Docker and Unikernels in a Cloud Platform [Conference] // 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2016. DOI: 10.1109/CCGrid.2016.86
13. Kocher P., Horn J., Fogh A., , Genkin D., Gruss D., Haas W., Hamburg M., Lipp M., Mangard S., Prescher T., Schwarz M., Yarom Y. Spectre attacks: Exploiting speculative execution [Conference] // IEEE Security and Privacy. - San Francisco : [s.n.], 2019. DOI: 10.1109/SP.2019.00002
14. Li Y., Dolan-Gavitt B., Weber S., Cappos J. Lock-in-Pop: Securing privileged operating system kernels by keeping on the beaten path [Conference] // USENIX Annual Technical Conf. - Santa Clara : [s.n.], 2017. pp. 1-13
15. Odun-Ayo Isaac, Geteloma Victor, Eweoya Ibukun, Ahuja Ravin Virtualization, Containerization, Composition, and Orchestration of Cloud Computing Services [Conference]. - California : Computational Science and Its Applications – ICCSA, 2019. pp. 403–417 DOI: 10.1007/978-3-030-24305-0_30
16. Batcher M. Go na praktike / Mett Batcher, Mett Farina ; per. s angl. R. N. Ragimova; nauch. red. A. N. Kiselev. – M.: DMK Press, 2017. – 374 s.
17. Biryukov, A. A. Informacionnaya bezopasnost': zashchita i napadenie [Tekst] / A. A. Biryukov. – 2-e izd., pererab. i dop. – M. : DMK Press, 2017. – 434 s. : il. – ISBN 978-5-97060-435-9.
18. Michael Kurth, Ben Gras, Dennis Andriesse, Cristiano Giuffrida, Herbert Bos, Kaveh Razavi NetCAT: Practical cache attacks from the network [Conference] // IEEE Symposium on Security and Privacy (SP). - San Francisco : Institute of Electrical and Electronics Engineers Inc., 2020. pp. 20-38 DOI: 10.1109/SP40000.2020.00082
19. Kai-Oliver Detken, Marcel Jahnke, Carsten Kleiner, Marius Rohde, Combining Network Access Control (NAC) and SIEM functionality based on open source [Conference]. - Bucharest, Romania : 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017. DOI: 10.1109/IDAACS.2017.8095094
20. Jain V., Qi S.,Ramakrishnan K.K. Fast Function Instantiation with Alternate Virtualization Approaches [Conference]. - California : IEEE Computer Society, 2021. - Vols. 2021-July. DOI: 10.1109/LANMAN52105.2021.9478808