ОБЕСПЕЧЕНИЕ СОВМЕСТИМОСТИ ТЕХНИЧЕСКИХ КОМПОНЕНТОВ ПРИ СОЗДАНИИ СИСТЕМЫ МОНИТОРИНГА ИНЦИДЕНТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Девицына С. Н.[1], Пилькевич П. В.[2]

DOI: 10.21681/2311-3456-2024-4-38-44

Цель исследования: *создание прототипа системы мониторинга инцидентов информационной безопасности типа pre-commit на рабочих станциях разработчиков программного обеспечения для внедрения DevSecOps в процесс разработки технических продуктов.*

Методы исследования: *анализ способов модернизации исходного кода компонентов системы мониторинга, синтез системы мониторинга инцидентов информационной безопасности, имитационное моделирование инцидентов информационной безопасности, обрабатываемых системой мониторинга, эксперимент.*

Результаты исследования. *В работе предложено решение по обеспечению защищенности разрабатываемого программного обеспечения в рамках методологии DevSecOps. Приведено описание процесса редактирования исходного кода для обеспечения совместимости программного модуля gitleaks и Filebeat при создании системы мониторинга инцидентов информационной безопасности. Показано, что процессы создания программного продукта должны проводиться параллельно с процедурами обеспечения безопасности исходного кода. В результате получен прототип многокомпонентной системы мониторинга инцидентов типа pre-commit, обнаруживающей и предоставляющей статистику по событиям, связанным с оставлением критической информации внутри произвольного исходного кода. Апробация работы и оценка эффективности системы мониторинга реализована на основе имитационного моделирования и эксперимента. Доказана работоспособность и эффективность системы мониторинга, в рамках эксперимента сделано нагрузочное тестирование в формате отправки большого потока инцидентов в систему с целью проверки корректности обработки каждого из них, и исключения потери инцидентов из-за большой нагрузки на сеть и технические модули. В результате исследования и моделирования предложен эффективный прототип системы мониторинга инцидентов информационной безопасности, который может быть использован отечественными компаниями-разработчиками для обеспечения и повышения эффективности кибербезопасности объектов информатизации с учетом требований импортозамещения.*

Новизна: *впервые предложено использовать систему мониторинга для исследования инцидентов DevSecOps с автоматизированным поиском уязвимостей в анализируемом исходном коде.*

Ключевые слова: *информационная безопасность, кибербезопасность, SIEM-системы, системы мониторинга, инциденты информационной безопасности, Opensearch, DevSecOps, безопасность программного обеспечения, импортозамещение.*

METHOD FOR ENSURING COMPATIBILITY OF TECHNICAL COMPONENTS WHEN CREATING A SYSTEM FOR MONITORING INFORMATION SECURITY INCIDENTS

Devitsyna S. N.[3], Pilkevich P. V.[4],

The purpose of the research *is to create a prototype of a pre-commit information security incident monitoring system at software developers' workstations to implement DevSecOps in the process of developing technical products.*

Research methods: *analysis of ways to modernize the source code of the monitoring system components, synthesis of the information security incident monitoring system, simulation modeling of information security incidents processed by the monitoring system, experiment.*

1 Девицына Светлана Николаевна, кандидат технических наук, доцент, ФГАОУ ВО «Севастопольский государственный университет», Севастополь, Россия. E-mail: sndevitsyna@sevsu.ru, ORCHID 0009-0009-1647-6701

2 Пилькевич Павел Вадимович, студент, ФГАОУ ВО «Севастопольский государственный университет», Севастополь, Россия. E-mail: pavel.piksel2012@mail.ru

3 Svetlana N. Devitsyna, PhD., Associate Professor, Sevastopol State University, Sevastopol, Russia. E-mail: sndevitsyna@sevsu.ru, ORCHID 0009-0009-1647-6701

4 Pavel V. Pilkevich, student, Sevastopol State University, Sevastopol, Russia. E-mail: pavel.piksel2012@mail.ru

Results of the study. *The paper proposes a solution to ensure the security of the developed software within the framework of the DevSecOps methodology. The article describes the process of editing the source code to ensure the compatibility of the gitleaks and Filebeat software modules when creating an information security incident monitoring system. It is shown that the processes of creating a software product should be carried out in parallel with the procedures for ensuring the security of the source code. As a result, a prototype of a multi-component pre-commit incident monitoring system that detects and provides statistics on events related to the retention of critical information within arbitrary source code. Approbation of the work and assessment of the effectiveness of the monitoring system was implemented on the basis of simulation modeling and experiment. The operability and efficiency of the monitoring system were proved, as part of the experiment, load testing was carried out in the format of sending a large stream of incidents to the system in order to check the correctness of processing each of them, and exclude the loss of incidents due to a heavy load on the network and technical modules. As a result of the research and modeling, an effective prototype of an information security incident monitoring system is proposed, which can be used by domestic development companies to ensure and improve the efficiency of cybersecurity of informatization objects, taking into account the requirements of import substitution.*

Novelty: *for the first time, it is proposed to use a monitoring system to investigate DevSecOps incidents with an automated search for vulnerabilities in the analyzed source code.*

Keywords: *information security, cybersecurity, SIEM systems, monitoring systems, information security incidents, Opensearch, DevSecOps, software security, import substitution.*

Введение

Реализация сквозных цифровых технологий заключается во внедрении ИБ-решений в процесс автоматизации технологических процессов сборки, настройки и развёртывания разрабатываемого ПО (DevSecOps), при этом методология создания безопасного продукта заключается в совместной работе команд информационной безопасности с командами, непосредственно разрабатывающими приложения. Наиболее перспективным стеком технологий для создания систем мониторинга на данный момент является совокупность технических средств Opensearch, Opensearch Dashboards, Logstash, Filebeat и gitleaks. Это обусловлено возможностью реализации требования по импортозамещению вследствие открытого исходного кода каждого из технических компонентов. Однако, существует проблема: не предусмотрено возможности корректного взаимодействия между модулями gitleaks и Filebeat, что создаёт трудности в объединении данных средств и делает невозможным создание системы мониторинга в текущий момент [1].

Целью работы является обеспечение совместимости программного модуля gitleaks и Filebeat при создании системы мониторинга инцидентов информационной безопасности.

Проблема модернизации исходного кода компонентов системы мониторинга

Для выбора решения проведен анализ актуальных научных работ и исследований по данной тематике. В работе Степанова Я. В. и др. [1] рассмотрены теоретическое основы создания собственного Центра обеспечения безопасности (SOC) на основе стека ELK и классификации MITRE. Авторами предлагается openssource стек ELK. Предлагается построить все процессы SOC на основе классификации MITRE, подобрать квалифицированный персонал и при помощи стека ELK эффективно реализовывать сбор и анализ больших данных в реальных проектах. Машанов В. В. [6] рассматривает важность проверки git-репозиториев на утечки и уязвимости для обеспечения безопасности проекта и конфиденциальности данных. Описывает несколько инструментов, таких как Gitrob, GitLeaks, TruffleHog и GitGuardian, которые помогают обнаружить и исправить проблемы безопасности в git-репозиториях. Каждый инструмент обладает своими особенностями. Работа содержит полезную информацию для разработчиков, которые заинтересованы в обеспечении безопасности своих проектов. Вместе с тем, никто из авторов не предлагает использовать систему мониторинга для исследования инцидентов DevSecOps с автоматизированным поиском уязвимостей в анализируемом исходном коде.

Предлагаемый подход

В ходе анализа особенностей функционирования системы мониторинга выявлено, что программная утилита для обнаружения оставленной конфиденциальной информации в произвольном исходном коде записывает результаты своей работы в файл формата JSON. Определён способ получения информации от gitleaks путём считывания данного файла указанного формата средствами сборщика данных Filebeat, у которого, в свою очередь, имеется специальный конфигурационный параметр для считывания информации именно в формате JSON [2]. Выяснилось, что ключевым различием между способом считывания средством Filebeat и способом записи средством gitleaks является то, что запись информации осуществляется блоками по соответствующим отчётам в один массив внутри файла, а считывания данных

предполагается построчное, причём в каждой строке должен находиться объект формата JSON. Такое различие приводит к успешной записи инцидентов в файл и неудачной попытке считывания этой информации сборщиком данных Filebeat, что не позволяет передать информацию об инцидентах оставления конфиденциальной информации в репозитории в Систему мониторинга данных инцидентов [3].

Для устранения выявленного несоответствия был изменён программный код внутри системной утилиты gitleaks, отвечающий за запись отчёта об обнаруженном инциденте с целью формирования постраничной записи отдельных объектов формата JSON в лог-файл для дальнейшего корректного считывания его сборщиком данных. Код представляет собой отдельную функцию для кодировки информации в формат JSON, на вход которой подаётся список со всеми отчётами обнаружения конфиденциальной информации в репозитории проекта. Ранее кодировка информации осуществлялась путём установки параметра отступа для повышения «читабельности» формируемых отчётов, и непосредственного разового преобразования всех отчётов в формат JSON [4]. Изменение кода представляет собой замену данного механизма на использование цикла, который берёт каждый отдельный сформированный отчёт и преобразовывает его в необходимый формат. Далее данные объекты постранично записываются в лог-файл[5].

Помимо этого, по умолчанию, с целью упрощения и ускорения работы с утилитой детектирования конфиденциальной информации в репозитории, существует возможность настройки автоматического запуска работы утилиты при выполнении определённых команд в репозитории. В разрабатываемой системе мониторинга это действие — коммит или же попытка записи изменений в репозиторий [5–7]. Для того, чтобы убрать необходимость загрузки всех файлов технического решения для обнаружения конфиденциальной информации в исходном коде, существует возможность указания специального параметра в своём репозитории, который ссылается на удалённый адрес к проекту утилиты gitleaks и, тем самым, производится автоматический запуск утилиты при попытке записи данных в репозиторий из адреса, зафиксированного в данной конфигурации (git hooks). Описанный механизм будет работать в том случае, если в папке проекта, который пытается автоматически запуститься, будет находиться другой конфигурационный файл[6,7,8] (.pre-commit-hooks.yaml). В этом файле будет указан идентификатор действия, необходимый для выполнения, и команда, запускающаяся автоматически с помощью git hooks. По умолчанию, такой конфигурационный файл с командой в gitleaks уже присутствует, однако, в нём указана команда, служащая лишь для демонстрации работы продукта, а запись обнаруженных утечек конфиденциальных данных нигде не остаётся, только демонстрируется пользователю, который совершает попытку фиксации изменений в своём репозитории[9].

Таким образом, для настройки автоматического запуска корректной команды при обращении к удалённому репозиторию изменённой утилиты gitleaks, была изменена команда, находящаяся в поле entry, которая указывает на необходимость включения полноценного режима обнаружения всех инцидентов утечки информационной безопасности по корневому адресу анализируемого репозитория с дальнейшей записью всех сформированных отчётов формата JSON в файл с фиксированным названием.

Обнаружение утечки конфиденциальной информации в произвольном репозитории средством gitleaks производится путём применения им конфигурируемых правил, создаваемых пользователями. Данные правила базируются на регулярных выражениях, которые, при совпадении со строками внутри исходного кода анализируемого продукта, формируют оповещение об обнаруженном инциденте информационной безопасности. Правила включают в себя [8, 9]:

- наименование правила, отображаемое в дальнейшем в сформированном отчёте в случае «срабатывания» данного правила;
- краткое описание правила, указывающего характер обнаруженной утечки;
- регулярное выражение (опционально), по которому осуществляется поиск совпадений с целью обнаружения инцидентов утечек конфиденциальной информации;
- значение степени важности обнаруженного инцидента;
- ключевые слова, по которым также осуществляется поиск внутри файлов с исходным кодом.

5 Программные инструменты обработки и визуализации данных. Elasticsearch, Logstash, Kibana, Grafana, Prometheus = Software tools for data processing and visualization. Elasticsearch, Logstash, Kibana, Grafana, Prometheus: учебное пособие / [И. В. Никифоров, О. А. Юсупова, Н. В. Воинов [и др.]; Санкт-Петербургский политехнический университет Петра Великого, Институт компьютерных наук и технологий, Высшая школа программной инженерии. — Санкт-Петербург: Политех-Пресс, 2023.

6 Ушаков, М. GitLab: локальный хостинг в стиле GitHub // Системный администратор. 2013. – №. 5. – С. 87–91.

7 Беккер, М. Я. и др. Использование цифровых сертификатов и протоколов SSL/TLS для шифрования данных при облачных вычислениях // Научно-технический вестник информационных технологий, механики и оптики. — 2011. – №. 4 (74). – С. 125–130.

8 Котенко, И. В., Кулешов, А. А., Ушаков, И. А. Система сбора, хранения и обработки информации и событий безопасности на основе средств Elastic Stack // Информатика и автоматизация. 2017. – Т. 5. – №. 54. – С. 5–34.

9 Шепелев, А. Н. и др. Анализ подходов и средств обработки сервисных журналов // Инженерный вестник Дона. – 2013. – Т. 27. – №. 4 (27). – С. 89.

Существует определённый набор правил, доступных по умолчанию, и возможность указания своих собственных правил в отдельном файле для настройки гибкого поиска всевозможных утечек информации, соответствующих требованиям безопасности предприятия. В случае автоматизированного удалённого обращения к репозиторию с утилитой gitleaks не представляется возможности передачи каким-либо эффективным и удобным способом пользовательских правил обнаружения инцидентов информационной безопасности. Следовательно, возникает необходимость редактирования уже имеющихся конфигураций с правилами безопасности с целью добавления туда своих собственных правил [10–13].

Конфигурация компонентов серверного блока системы мониторинга

Конфигурация поискового движка Opensearch включает в себя перечень параметров, отвечающих за имя сервиса, его адрес, настройки безопасности и иные специализированные настройки для кластера. В случае конфигурации конкретного прототипа, используются наиболее оптимальные с точки зрения технических мощностей и функциональных требований к продукту параметры:

- устанавливается имя кластера, который будет использоваться во всей системе мониторинга данным поисковым движком;
- указывается необходимость прослушивания всех имеющихся сетевых интерфейсов на наличие подключений к нему;
- включается режим блокирования в области памяти RAM с целью избежания критических неполадок, вызванных отсутствием памяти;
- включается использование пороговых значений для распределения узлов на диске;
- устанавливаются верхние и нижние пороги использования диска поисковым движком с целью оптимизации работы с памятью;
- разрешается использование демонстративных сертификатов для настройки тестового шифрования между узлами в системе;
- включается режим использования шифрования SSL при обмене информацией с поисковым движком;
- указываются пути для сертификатов шифрования как для транспортных путей, отвечающих за получение информации, так и для протокола http, необходимого для доступа к поисковому движку посредством средства визуализации Opensearch Dashboards;
- для ускорения работы отключается проверка имени хоста в транспортных соединениях.

Для корректной конфигурации средства визуализации информации Opensearch Dashboards необходимо указать параметры, отвечающие за имя сервиса, прослушиваемый адрес, настройки безопасности и иные специализированные параметры[10]. В конкретном случае развёртывания экспериментального демонстрационного прототипа системы мониторинга, были сконфигурированы следующие параметры:

- указывается имя запускаемого сервиса;
- включается прослушивание всех сетевых интерфейсов на прослушивание клиентских подключений к средству визуализации;
- указывается имя учётной записи, по которой осуществляется подключение к поисковому движку Opensearch;
- указывается пароль для учётной записи, используемой для подключения к поисковому движку Opensearch;
- включается полноценная поддержка шифрования SSL во всех сетевых соединениях как между клиентами, так и между средством визуализации и поисковым движком;
- указываются пути к файлам ключа и сертификата, которые задействуются в налаживании шифрования при сетевых соединениях между техническими продуктами.

Конфигурация компонентов клиентского блока системы мониторинга

Для обеспечения корректной работы всех технических составляющих клиентского блока системы мониторинга инцидентов типа pre-commit, необходимо осуществить конфигурацию утилиты git hooks. Данная утилита будет отвечать за автоматический вызов команды при попытке фиксации новых изменений исходного кода в репозитории для поиска оставленной конфиденциальной информации в данном репозитории путём задействования модернизированной утилиты gitleaks. Помимо данной операции конфигурирования, необходимо также наладить работу сборщика инцидентов с рабочих станций анализируемых пользователей (Filebeat). Это необходимо для получения и отправки сведений об инцидентах оставления конфиденциальной информации в серверную составляющую разрабатываемой системы мониторинга [14].

Для корректной работы утилиты git hooks был создан файл .pre-commit-hooks.yaml с содержимым, реализующим требуемое поведение при попытке пользователя занести новые изменения в репозиторий. Код содержит в себе адрес репозитория, к которому нужно обратиться при выполнении пользователем команды git commit, хэш-сумму конкретной

10 Календарев, А. Горизонтальное масштабирование. Проблемы и пути решения // Системный администратор. – 2014. – №. 10. – С. 54–62.

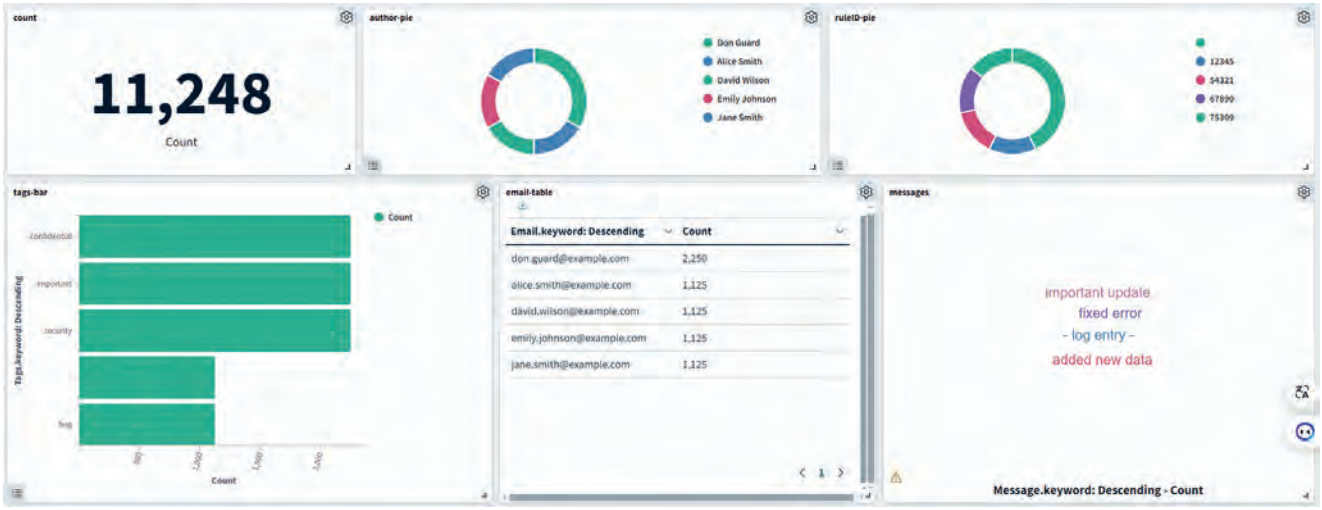

Рис. 1. Демонстрационный стенд, построенный в средстве визуализации Opensearch Dashboards

ветки удалённого репозитория gitleaks, содержащем изменения для обеспечения совместимости, а также идентификатор действия, по которому будет найдена команда, необходимая для выполнения в проекте утилиты gitleaks при запуске корректной команды поиска утечек конфиденциальной информации в анализируемом репозитории.

Для корректного функционирования сборщика данных Filebeat была создана специализированная конфигурация, соответствующая всем описанным требованиям для внедрения данного технического блока в состав разрабатываемой системы мониторинга и обеспечения бесперебойной связи клиентской составляющей с серверной составляющей. Конфигурация включает в себя параметры настройки разрешения системных вызовов процесса rseq с целью обеспечения бесперебойной работы сборщика данных и отсутствия конфликтов с нехваткой прав доступа к системным вызовам[11]. Отсутствие подобной настройки может вызывать отказ в запуске сборщика данных, что нарушит цепочку получения инцидентов с анализируемых рабочих станций пользователей. Помимо этого, добавлены четыре процессора, выполняющих следующие действия:

- добавление системной информации о хосте, на котором расположен Filebeat;
- добавление системной информации об облачном хранилище в случае, если Filebeat запущен на нём;
- добавление системной информации о контейнере docker в случае, если Filebeat запущен на нём;
- добавление системной информации о кластере Kubernetes в случае, если Filebeat развёрнут в рамках данного кластера.

11 Каменная, Е. В., Путилова, С. Е., Щербинина, И. А. Обзор современных подходов к обеспечению безопасности клиентской части веб-приложений // Транспортное дело России. – 2017. – №. 6. – С. 66–71.

Для полноценного функционирования разрабатываемой системы мониторинга произведено конфигурирование всех составляющих системы. В результате был получен пример полноценной и функционирующей системы мониторинга, и проведен анализ ее эффективности.

Проверка эффективности системы мониторинга

Смоделировав несколько различных вариантов инцидентов информационной безопасности, обрабатываемых системой мониторинга, было произведено нагрузочное тестирование, которое предполагало собой отправку большого потока инцидентов в систему с целью проверки корректности обработки каждого из инцидентов, и исключения потери инцидентов из-за большой нагрузки на сеть и технические модули разработанной системы мониторинга.

В средстве визуализации данных был создан демонстрационный стенд, показывающий информацию об общем количестве обнаруженных инцидентов. Стенд отражает: визуализацию соотношения количества событий относительно пользователей и идентификаторов сработанных правил gitleaks, график количества событий с агрегацией по тегам, таблицу электронных почт авторов коммитов, вызвавших инцидент информационной безопасности, перечень описаний коммитов, визуализацию количества событий относительно имени файлов, в которых был обнаружен инцидент (рис. 1).

Также имеются дополнительные информативные диаграммы, содержащие в себе агрегации по оставшимся полям инцидентов (рис. 2).

В ходе нагрузочного тестирования было подано 11 248 инцидентов, на демонстрационном стенде отображено такое же количество событий, что свидетельствует об отсутствии потерь событий. Выбранная

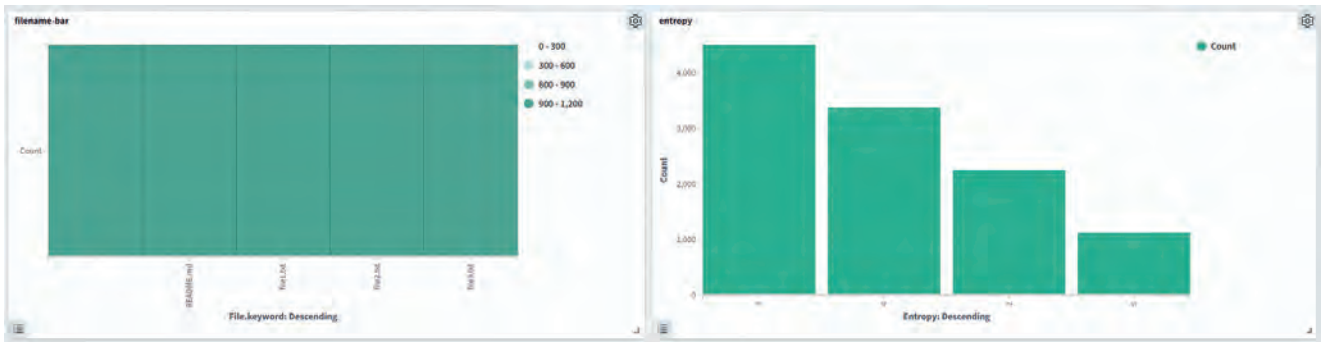

Рис. 2. Вспомогательная часть демонстрационного стенда, построенного в средстве визуализации Opensearch Dashboards

архитектура системы позволяет эффективно обнаруживать утечки разного типа и иные данные, соответствующие шаблонам поиска, в исходном коде произвольных проектов любых размеров.

Заключение

Для полноценного развёртывания системы мониторинга были внесены изменения в исходный код использующихся продуктов с целью обеспечения и улучшения совместимости продуктов друг с другом. Отредактирована система логирования инцидентов и изменены конфигурации запуска скриптов, проверяющих наличие утечек конфиденциальной информации в исходном коде проектов. Для полноценного функционирования разрабатываемой системы мониторинга произведено конфигурирование всех составляющих системы. В результате получен прототип полноценной и функционирующей системы мониторинга. Формирующиеся события содержат в себе максимально подробную информацию, доступную для визуализации в любом виде с целью досконального анализа и проведения подробного расследования инцидента. На основе имитационного моделирования была доказана эффективность прототипа системы, что позволяет внедрять её в промышленных масштабах отечественными компаниями-разработчиками. Практическая значимость заключается в разработке интеллектуального продукта на основе созданной архитектуры и востребованности подобного технического решения крупными отечественными ИТ-компаниями, заинтересованными в обеспечении кибербезопасности различных объектов информатизации, с учетом требований импортозамещения [15, 16].

Литература

1. *Степанов Я. В. и др. Создание собственного SOC при помощи классификации MITRE и Opensource стека ELK / Я. В. Степанов, Т. Н. Копышева, Т. В. Митрофанова, Т. Н. Смирнова // Информационные технологии в науке, управлении и образовании: междисциплинарный подход и тенденции развития: Сб. матер. Всероссийской научно-практической конференции (Димитровград, 12 ноября 2021 года). — Димитровград: Изд-во Димитровградского инженерно-технологического института — филиала федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет «МИФИ», 2021. С. 229—236.*
2. *Петров В. В., Брюханов, К. В., Авксентьева, Е. Ю. Сетевой мониторинг: анализ сетевого трафика с помощью ELK // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2020. №. 5. С. 102–105.*
3. *Еровлева Р. В., Еровлев, П. А. Мониторинг с помощью Micrometer, Prometheus и Grafana // Постулат. 2021. № 7.*
4. *Dhakal K. et al. Log Analysis and Anomaly Detection in Log Files with Natural Language Processing Techniques. — Appl. Sci. 2022, 12.*
5. *Шелепина О. Д., Хадорич Д. Д. Сравнительный анализ инструментов управления журналами на примере ELK и Graylog // Вызовы глобализации и развитие цифрового общества в условиях новой реальности: Сб. матер. IV Международной научно-практической конференции. Москва, 2022. – Изд-во: Алеф. 2022. С. 137–141.*
6. *Машанов В. В. Как обезопасить git-репозитории: обзор инструментов для обнаружения утечек и уязвимостей // Актуальные вопросы современной науки: сборник статей. – Изд-во: Наука и Просвещение (ИП Гуляев Г.Ю.). 2023. С. 53–56.*
7. *Вахрамов С. В. и др. Использование prettier и git hooks для автоматического поддержания культуры кода в typescript-проекте // Научное обозрение. Технические науки. 2020. №. 4. С. 24–28.*
8. *Сарнавский А. П. Разработка инструмента управления уязвимостями на основе Elasticsearch: выпускная квалификационная работа бакалавра: направление 10.03.01 «Информационная безопасность»; образовательная программа 10.03.01_03 «Безопасность компьютерных систем». 2022.*
9. *Симанков В. С., Петрова В. А. Мониторинг информационной безопасности в интеллектуальном ситуационном центре // Поведенческие теории и практика российской науки. 2021. С. 29–35.*
10. *Коптева А. В., Князев И. В. Анализ проблемы преобразования данных формата JSON в строго типизированных языках программирования на примере Golang // Проблемы науки. 2021. №. 7 (66). С. 5.*
11. *Кутузов К. О. Программирование RESTful приложений на языке программирования Golang // Молодость. Интеллект. Инициатива. 2021. С. 23–24.*

12. Разумков И. А. Автоматизация поиска уязвимостей в программах на языке Golang: выпускная квалификационная работа бакалавра: направление 10.03.01 «Информационная безопасность»; образовательная программа 10.03.01_03 «Безопасность компьютерных систем». 2023.
13. Палаш Б. В., Голубничий А. А. Основные способы обеспечения безопасности клиент-серверных приложений // Modern Science. 2020. №. 2-1. С. 383–385.
14. Черников А. С. и др. Обзор применения подхода микросервисной архитектуры при проектировании клиентской части веб-приложения //Дневник науки. 2020. №. 4. С. 31.
15. Девицына С. Н., Пилькевич П. В., Удод Е. В. Способы улучшения защищённости сервисов, использующих JWT-токены // Экономика. Информатика. 2023. Т. 50. №1. С. 144–151.
16. Адгемов И. Э., Девицына С. Н. Управление безопасностью беспроводной локальной вычислительной сети // Экономика. Информатика. 2023. Т. 50. № 1. С. 183–190.

References

1. Stepanov Ja. V. i dr. Sozdanie sobstvennogo SOC pri pomoshhi klassifikacii MITRE i Opensource steka ELK / Ja. V. Stepanov, T. N. Kopysheva, T. V. Mitrofanova, T. N. Smirnova // Informacionnye tehnologii v nauke, upravlenii i obrazovanii: mezhdisciplinarnyj podhod i tendencii razvitija: Sb. mater. Vserossijskoj nauchno-prakticheskoj konferencii (Dimitrovgrad, 12 nojabrja 2021 goda). — Dimitrovgrad: Izd-vo Dimitrovgradskogo inzhenerno-tehnologicheskogo instituta — filiala federal'nogo gosudarstvennogo avtonomnogo obrazovatel'nogo uchrezhdenija vysshego obrazovanija "Nacional'nyj issledovatel'skij universitet «MIFI», 2021. S. 229—236.
2. Petrov V. V., Brjuhanov, K. V., Avksent'eva, E. Ju. Setevoj monitoring: analiz setevogo trafika s pomoshh'ju ELK // Sovremennaja nauka: aktual'nye problemy teorii i praktiki. Serija: Estestvennye i tehnicheskie nauki. 2020. №. 5. S. 102–105.
3. Erovleva R. V., Erovlev, P. A. Monitoring s pomoshh'ju Micrometer, Prometheus i Grafana // Postulat. 2021. № 7.
4. Dhakal K. et al. Log Analysis and Anomaly Detection in Log Files with Natural Language Processing Techniques. — Appl. Sci. 2022, 12.
5. Shelepina O. D., Hadorich D. D. Sravnitel'nyj analiz instrumentov upravlenija zhurnalami na primere ELK i Graylog // Vyzovy globalizacii i razvitie cifrovogo obshhestva v uslovijah novoj real'nosti: Sb. mater. IV Mezhdunarodnoj nauchno-prakticheskoj konferencii. Moskva, 2022. – Izd-vo: Alef. 2022. S. 137–141.
6. Mashanov V. V. Kak obezopasit' git-repozitorii: obzor instrumentov dlja obnaruzhenija utechek i ujazvimostej // Aktual'nye voprosy sovremennoj nauki: sbornik statej. — Izd-vo: Nauka i Prosveshhenie (IP Guljaev G.Ju.). 2023. S. 53–56.
7. Vahramov S. V. i dr. Ispol'zovanie prettier i git hooks dlja avtomaticheskogo podderzhanija kul'tury koda v typescript-proekte // Nauchnoe obozrenie. Tehnicheskie nauki. 2020. №. 4. S. 24–28.
8. Sarnavskij A. P. Razrabotka instrumenta upravlenija ujazvimostjami na osnove Elasticsearch: vypusknaja kvalifikacionnaja rabota bakalavra: napravlenie 10.03.01 «Informacionnaja bezopasnost'»; obrazovatel'naja programma 10.03.01_03 «Bezopasnost' komp'juternyh sistem». 2022.
9. Simankov V. S., Petrova V. A. Monitoring informacionnoj bezopasnosti v intellektual'nom situacionnom centre // Povedencheskie teorii i praktika rossijskoj nauki. 2021. S. 29–35.
10. Kopteva A. V., Knjazev I. V. Analiz problemy preobrazovanija dannyh formata JSON v strogo tipizirovannyh jazykah programmirovanija na primere Golang // Problemy nauki. 2021. №. 7 (66). S. 5.
11. Kutuzov K. O. Programmirovanie RESTful prilozhenij na jazyke programmirovanija Golang // Molodost'. Intellekt. Iniciativa. 2021. S. 23–24.
12. Razumkov I. A. Avtomatizacija poiska ujazvimostej v programmah na jazyke Golang: vypusknaja kvalifikacionnaja rabota bakalavra: napravlenie 10.03.01 «Informacionnaja bezopasnost'»; obrazovatel'naja programma 10.03.01_03 «Bezopasnost' komp'juternyh sistem». 2023.
13. Palash B. V., Golubnichij A. A. Osnovnye sposoby obespechenija bezopasnosti klient-servernyh prilozhenij // Modern Science. 2020. №. 2-1. S. 383–385.
14. Chernikov A. S. i dr. Obzor primenenija podhoda mikroservisnoj arhitektury pri proektirovanii klientskoj chasti veb-prilozhenija // Dnevnik nauki. 2020. №. 4. S. 31.
15. Devicyna S. N., Pil'kevich P. V., Udod E. V. Sposoby uluchshenija zashhishhjonnosti servisov, ispol'zujushhih JWT-tokeny // Jekonomika. Informatika. 2023. T. 50. №1. S. 144–151.
16. Adgemov I. Je., Devicyna S. N. Upravlenie bezopasnost'ju besprovodnoj lokal'noj vychislitel'noj seti // Jekonomika. Informatika. 2023. T. 50. № 1. S. 183–190.