DOI 10.36622/VSTU.2023.26.2.013
УДК 004.056.47

СОЗДАНИЕ КИБЕРПОЛИГОНА: ФОРМИРОВАНИЕ БЛОКА СИМУЛЯЦИИ

С.С. Куликов, В.К. Федоров

В настоящей статье описаны особенности разработки и реализации блока симуляции киберполигона, предназначенного для проведения тренировок и аудита кибербезопасности. В рамках работы были проанализированы существующие реализации киберполигонов, основные принципы их функционирования, на основе чего был определен подход к моделированию и реализован соответствующий программный блок симуляции. Реализованный программно-технический комплекс (ПТК) позволяет динамически наполнять инфраструктуру, отображать сетевые топологии в виде динамического графа. Также предусмотрено формирование отчётов по найденным уязвимостям и эксплойтам. Работа с блоком симуляции осуществляется посредством использования веб-страницы, функционал которой позволяет добавлять узлы, изменять топологию сети, запускать сетевое сканирование и формировать отчёт безопасности. Гибкость ПТК достигается применением микросервисной архитектуры совместно с технологией контейнеризации. Экспериментальные результаты показывают, что акцент на симуляцию процессов, а также графовое представление инфраструктуры позволяет не только повысить уровень осведомленности в сфере информационной безопасности (ИБ), но и эффективно выявлять и анализировать уязвимости, прогнозировать потенциальные угрозы от действий злоумышленников и принимать соответствующие меры для защиты информационных систем.

Ключевые слова: киберполигон, граф, уязвимость, эксплойт.

Введение

В настоящее время явление киберпреступности становится все более распространенным. Так по данным «Positive tech.» с 2022 года резко выросли атаки на веб-ресурсы в следующих сферах: транспортная отрасль (на 33%), СМИ (на 26%), госучреждения (на 18%), банковский сектор (на 5%) [1]. Увеличение числа атак на веб-ресурсы также обусловлено появлением уязвимостей, найденных, в частности, в популярных плагинах, таких как «WordPress», «Magento». Наиболее популярными среди злоумышленников оказались следующие уязвимости:

- CVE-2021-44228, или «Log4Shell» (в прикладном библиотеке логирования «Apache Log4j 2»),
- CVE-2022-22965, или «Spring4Shell» (в Java Spring Framework),
- CVE-2022-24086 (уязвимость в прикладном ПО «Adobe Commerce»),
- CVE-2021-32648 (уязвимость в прикладном ПО «October CMS»),
- CVE-2022-3180 (уязвимость в плагине «WPGateway» популярной системы управления контентом «WordPress»).

В ряде компаний изменился подход к ИБ: на замену иностранным решениям пришли отечественные, спрос на которые возрос на 20% [2]. Атаки трансформируются и время на написание эксплойтов сокращается. В связи с этим организациям важно знать и понимать в реальном времени, какая инфраструктура может быть скомпрометирована и как этого не допустить. Необходимо проводить киберучения для специалистов по ИБ. Решить данную проблему позволяют киберполигоны, позволяющие проводить атаки на информационные ресурсы.

Существующие решения зачастую содержат готовые шаблоны атак и уязвимых компонентов [2, 3]. Заказчик после приобретения получает окружение с заготовленными уязвимостями. Данный подход имеет недостатки, связанные с тем, что в нём нет связи с реальной инфраструктурой организации и её уязвимостями, а также цена решений, основанных на симуляции виртуального окружения, крайне высока.

Исходя из выше сказанного, существует проблематика отсутствия в отечественных ПТК процесса симуляции организации с моделированием уязвимостей программного

обеспечения и операционных систем самих узлов сети. Для разрешения данной проблемы необходимо разработать инструмент, который позволит эффективно обнаруживать уязвимости и предотвращать угрозы ИБ. В ходе решения данной задачи был создан программный компонент симуляции киберполигона, позволяющий:

- динамически заполнять архитектурные составляющие и топологию сети организации,
- находить и обновлять данные о паспортах уязвимостей,
- моделировать инфраструктуру в виде интерактивного графа,
- симулировать сетевое взаимодействие узлов сети и сканирование узлов,
- искать уязвимости и эксплойты прикладных программ и операционных систем.

Исследование существующих киберполигонов и особенностей их применения

Исследование существующих киберполигонов, их типов и особенностей - важный шаг на пути создания собственного ПТК киберполигона с элементами симуляции процессов работы сети организации. Полигоны очень важны в обеспечении ИБ организаций и государственных структур.

Одним из наиболее распространенных типов является виртуальный киберполигон, который представляет среду, в которой участники могут тренироваться в защите от кибератак [4]. Виртуальный киберполигон может быть использован как для индивидуальной тренировки, так и для совместных работ в команде.

Еще одним типом является физический киберполигон, который представляет собой специально оборудованное помещение с серверами, в котором участники могут тренироваться в реальных условиях, имитирующих кибератаки и защиту от них. Физический киберполигон позволяет участникам получить более реалистичный опыт в виду физического, а не удаленного присутствия на киберучениях [5].

Киберполигоны могут быть использованы для обучения студентов, сотрудников компаний и государственных структур, а также для проведения тренировок военных и правоохранительных органов [6]. Кроме того, киберполигоны могут быть использованы для проведения как хакатонов, так и CTF-соревнований, что позволяет участникам продемонстрировать свои навыки.

Рассмотрим и выделим особенности, которые присутствуют в отечественных реализациях киберполигонов (табл. 1), западные аналоги ссхожи с отечественными и поэтому в сравнении не участвуют [7].

Таблица 1

Сравнение существующих отечественных киберполигонов

Наименование киберполигона, разработчика	**Эмуляция**	**Симуляция**	**Доступность**	**Формат использования**
Национальный киберполигон (ПАО Ростелеком)	+	-	В рамках соревнований / покупка или аренда	Состязания команд на виртуальной инфраструктуре
«Киберполигон» (ООО «Киберполигон»)	+	-	Покупка или аренда	Состязания команд на виртуальной инфраструктуре
«The Standoff» (ПАО «Positive Technologies)	+	-	Только в рамках соревнований	Состязания команд на физической и виртуальной инфраструктурах
«The Standoff 365» (ПАО «Positive Technologies)	+	-	Онлайн	Состязания команд на виртуальной инфраструктурах

Продолжение табл. 1

«BI.ZONE Cyber Polygon» (ООО «BI.ZONE» и ПАО «СБЕРБАНК»)	+	-	Только в рамках соревнований	Состязания команд на виртуальной инфраструктуре
«Jet CyberCamp» (АО «Инфосистемы Джет)	+	-	Покупка или аренда	Индивидуально на облачной инфраструктуре
«Ampire» (АО «Перспективный мониторинг»)	+	-	Покупка или аренда	Индивидуально в рамках учебного класса

Таким образом, каждое отечественный решение использует дорогостоящий процесс эмуляции [1-9]. Некоторые киберполигоны схожи с форматом соревнований CTF, в которых, участники решают задачи, связанные с хакингом, аудитом, криптографией, стеганографией и другими темами, чтобы получить доступ к «флагу» – секретной информации.

Среди отечественных аналогов не ведется открытой и публичной разработки, а также весь код проектов закрыт и зачастую поставляется, либо выдается в аренду в виде «черного ящика», то есть, в поставке такого рода киберполигонов присутствует физический сервер компании, а зачастую и сопутствующее сетевое оборудование [10].

Структура и функционал программного комплекса

Техническая составляющая

Программно-технический модуль симуляции киберполигона был реализован при помощи языка программирования Python, ввиду быстроты написания кода, низкого порога вхождения, кроссплатформенности и большого количества готовых библиотек. Однако стоит обратить внимание на то, что язык имеет один нюанс, он не компилируемый, а интерпретируемый, в следствие этого, проекты, написанные на его стеке, будут иметь проигрыш в производительности.

Функциональные библиотеки

Реализация данной системы была основана на фреймворке «Flask» в совокупности с инструментарием «Pyvis» для анализа данных и построения динамичных графов.

«Flask» – это фреймворк для создания веб-приложений на языке «Python». Он предоставляет минимальный набор инструментов для создания веб-приложений, включая маршрутизацию, обработку запросов и ответов, шаблонизацию и управление сессиями.

«Flask» является легковесным и гибким фреймворком, который не навязывает сильной структуры приложения. Он позволяет разработчикам создавать приложения любой сложности, от простых веб-страниц до сложных веб-порталов.

Основные особенности фреймворка включают:

- небольшой размер пакетов,
- гибкость и расширяемость,
- использование шаблонов «Jinja2» для удобного отображения данных в формате «HTML»,
- модули авторизации и аутентификации,
- встроенная поддержка тестирования приложений, представленная в виде готовых шаблонов,
- поддержку множества расширений и плагинов, которые упрощают разработку веб-приложений.

Модуль «Pyvis» – это библиотека для визуализации графов в «Python», она основана на библиотеке «Vis.js», которая в свою очередь предоставляет набор инструментов для создания интерактивных веб-графов.

Данный модуль позволяет создавать и настраивать графы с помощью кода, а затем отображать их в браузере с помощью интерфейса визуализации «Vis.js» (рис. 1). «Pyvis» также поддерживает анимацию

графов, что позволяет создавать динамически информативные визуализации для анализа и отображения данных. В данной работе был переписан модуль, отвечающий за отображение графа. Дело в том, что разработчиками не было предусмотрено встраивание графа в отдельные веб-формы, но ввиду того, что данный проект распространяется под лицензией свободного программного обеспечения, переписать модуль не составило труда.

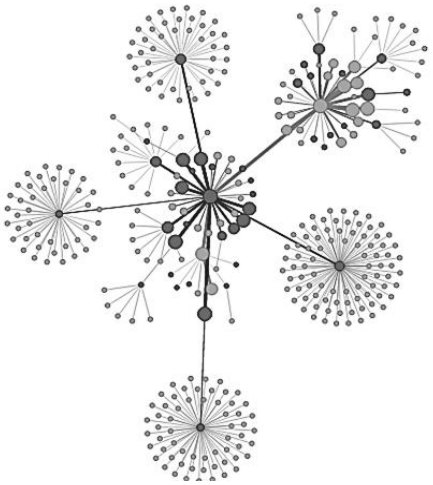

Рис. 1. Пример сетевого графа интерактивной связи узлов

В качестве фронтенд библиотеки выбран «Bootstrap5», позволяющий быстро создавать формы входа, авторизации, навигации, а также динамических форм ввода данных. Стоит отметить обширную документацию, предоставляющую подробную информацию о каждом элементе, будь то кнопки, таблицы, модальные окна, их стили и обработчики.

В качестве библиотеки обработчика запросов пользователя была выбрана библиотека «jQuery», позволяющая уменьшить количество «JavaScript» кода за счет готовых кодовых конструкций, например отправки форм данных для последующей обработки.

Чтобы приложение было кроссплатформенным и легко масштабируемым было решено использовать оркестратор «Docker», позволяющий собирать все зависимости ПТК в единый файл и запускать его на любой платформе, где используется процесс контейнеризации в независимости от архитектуры процессора и установленной операционной системы.

Для доступа к разным модулям программного решения используется высокопроизводительный веб-сервер «Nginx». С его помощью мы балансируем нагрузку, закрываем доступ к приложению на прямую, управляем механизмами кеширования, а также защищаем интернет-трафик от прослушивания.

Также используется сервер авторизации «Keycloak». Ввиду своей гибкости и простоты настройки именно он был выбран для процессов авторизации и аутентификации пользователей, предоставляя единый центр управления учетными записями, аудитом доступа, объединяя пользователей в удобные логические единицы, именуемые области безопасности.

Технологическое обеспечение

Блок симуляции ПТК киберполигона включает:

- «фронтенд-компонент» предоставляет конечному пользователю веб-интерфейс для работы с инфраструктурой,
- «бэкенд-компонент» отвечает за взаимодействие пользователя и сущностей киберполигона, создает узлы и графы, по данным, отправленным с веб ресурса,
- внешний компонент «Паспорт уязвимостей» представлен доработкой проекта «OpenCve» с открытым исходным кодом. Он позволяет получить список уязвимостей и эксплойтов программных средств по запросу. Доработка заключалась в контейнеризации данного приложения и его прикладных компонентов, где в качестве оркестратора компонента был выбран «Docker-compose»,
- база данных используется для хранения информации о пользователях, их активностях и других данных, необходимых для работы системы,
- аутентификация и авторизация - блок симуляции киберполигона использует механизмы аутентификации и авторизации для обеспечения собственной безопасности. С помощью данной технологии мы создаем и разграничиваем пользователей с использованием «рилмов».

Структурно-функциональная схема программного комплекса

Данный ПТК блока симуляции киберполигона состоит из следующих элементов, отраженных на рис. 2 и 3.

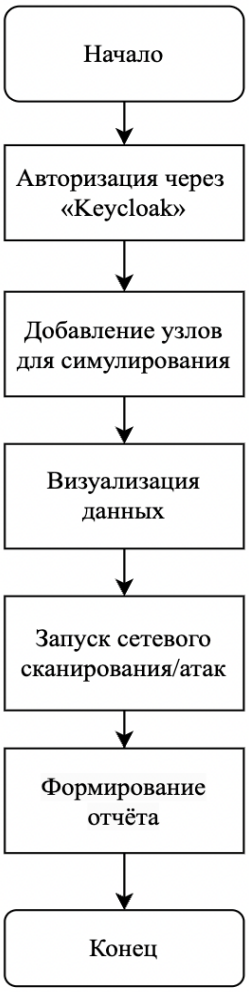

Рис. 2. Блок-схема работы «фронтенд-компонента»

На блок-схеме можно заметить, что после старта работы компонента «фронтенд-компонента», используется авторизация через компонент «Keycloak».

Для добавления узлов используются динамические формы и отправка запросов в формате «AJAX» на «бэкенд-компонент». В отправляемые данные входят:

- сессионные данные пользователя, для валидации прав доступа пользователя,
- имя узла,
- операционная система,
- версия дистрибутива,
- используемые порты,
- установленное программное обеспечение.

Обработка данных запросов выполняется с использованием «JavaScript» и модуля «jQuery», сама веб-страничка создается с использованием «Bootstrap5», «HTML5». Каждый запрос проверяется и валидируется на наличие специальный сессионных данных, во избежание несанкционированного доступа к элементам «API-интерфейса» и «фронтед- компоненту».

Рис. 3. Блок-схема работы «бэкенд-компонента»

При попытке злоумышленника получить доступ к какому-либо ресурсу без пройденной авторизации и аутентификации он получит ошибку доступа, так как данные проверки заложены в архитектуру проектного решения (рис. 4). Если специальных заголовков у пользователя не находится, ему возвращается приглашение авторизоваться, если же не проходит валидация самих данных о узлах, то пользователь получает в ответе от «бэкенд-компонента» ошибку с опознавательным кодом.

Таким образом мы защищаем приложение от неавторизированных пользователей и невалидных данных, отправленных через форму. Используя данные подходы в обработке запросов, мы повышаем защищенность программного продукта.

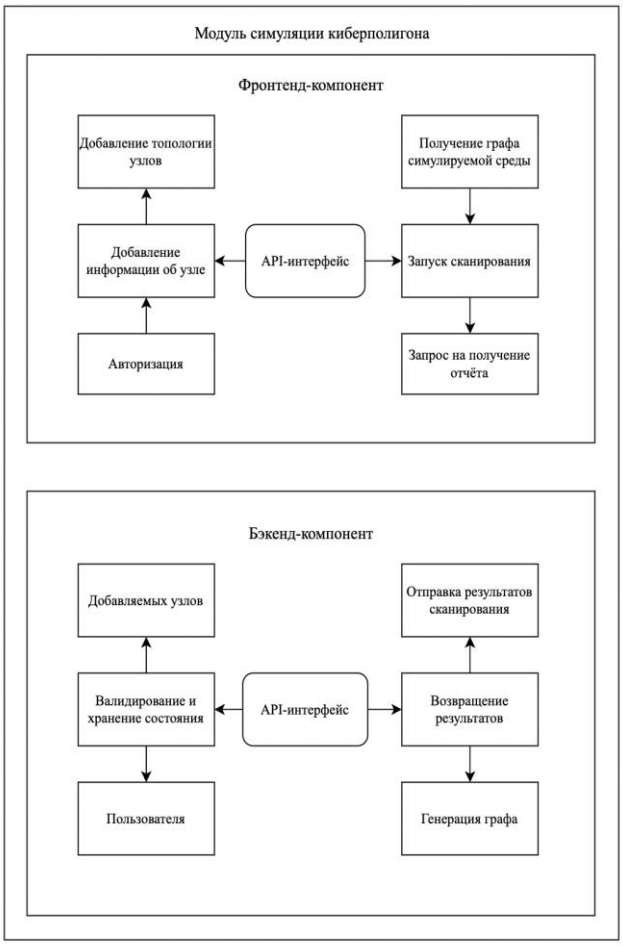

Рис. 4. Схема взаимодействия компонент

Компонент «Паспорт уязвимостей» представляет собой программный модуль, который был заимствован из программного комплекса «OpenCve», распространяемого под свободной лицензией. Данное программное обеспечение было доработано с целью улучшения функциональности и повышения эффективности процесса выявления и анализа уязвимостей в информационной системе. Данное программное решение выступает смежным компонентом между «бэкенд-компонентом» и базой данных об уязвимостях. С его помощью мы получаем свежую информацию о новых «CVE» как в операционных системах, так и на уровне прикладных программ, а также можем использовать интеграцию через «API-интерфейс».

Рис. 5. Блок-схема интеграционного слоя

Данное программное обеспечение позволяет в автоматическом режиме получать информацию о «CVE» уязвимостях по различным фильтрам, например, по производителю, вектору атаки. Стоит отметить, что сайт, предоставляемый разработчиками, имеет большое количество ограничений:

- дневное количество запросов к «REST-интерфейсу» ограничено 250 запросами,
- нельзя автоматически получить списки уязвимостей по нескольким программным продуктам,
- модуль оповещений работает только с почтовым протоколом «SMTP», где нет интеграций для оповещений.

Данная программа позволяет:

- изучать базу данных «CVE» и отфильтровывать,
- осуществлять поиск по вендору, продукту, оценке «CVSS» или «CWE»,
- получать уведомления о новых «CVE» как в общем виде, так и в формате подписки,
- видеть детальную оценку «CVE», векторы атаки на данную уязвимость,
- отображение истории каждого «CVE» и просмотр их последних изменений на главной странице,
- программа распространяется под свободной лицензией, то есть мы имеем доступ к её исходному коду.

Обзор программного комплекса

С помощью ПТК, можно воссоздать инфраструктуру сети компании, в которой происходят кибератаки и другие события, связанные с ИБ. Интерфейс при ближайшем рассмотрении предлагает нам интуитивно понятную форму ввода информации о узлах проекта.

В качестве примера добавим новый узел к существующей конфигурации (рис. 6), для этого:

- необходимо ввести имя узла (оно должно быть уникальным), хорошим тоном будет «nodeN», где N - порядковый номер,
- следующим шагом следует выбрать операционную систему (на выбор предоставляется «Linux» или «Windows»),
- если в предыдущем пункте выбран «Linux», то теперь необходимо выбрать дистрибутив, который в данный момент представлен «Centos» и «Ubuntu». В зависимости от выбора дистрибутива в следующей форме будет взаимозаменяться форма, так в случаях, если выбрана «Windows», выводится форма с выбором версии,
- в следующем пункте необходимо указать версию дистрибутива, в нашем случае это «Ubuntu 20.04 LTS»,
- далее необходимо выбрать используемые порты, в нашей конфигурации это ssh, http, https, однако если на предыдущем пункте дистрибутивом был «Windows», то мы могли также выбрать rdp порт,
- последним пунктом остается добавление информации о программном обеспечении узла. В нашем случае это веб-сервер «Nginx». По аналогии добавим еще 6 узлов, для дальнейшего рассмотрения связей. После необходимо кликнуть на кнопку «Добавить узел». Также можно использовать кнопку «Обновить узлы», если в приложение они были добавлены в обход веб-формы.

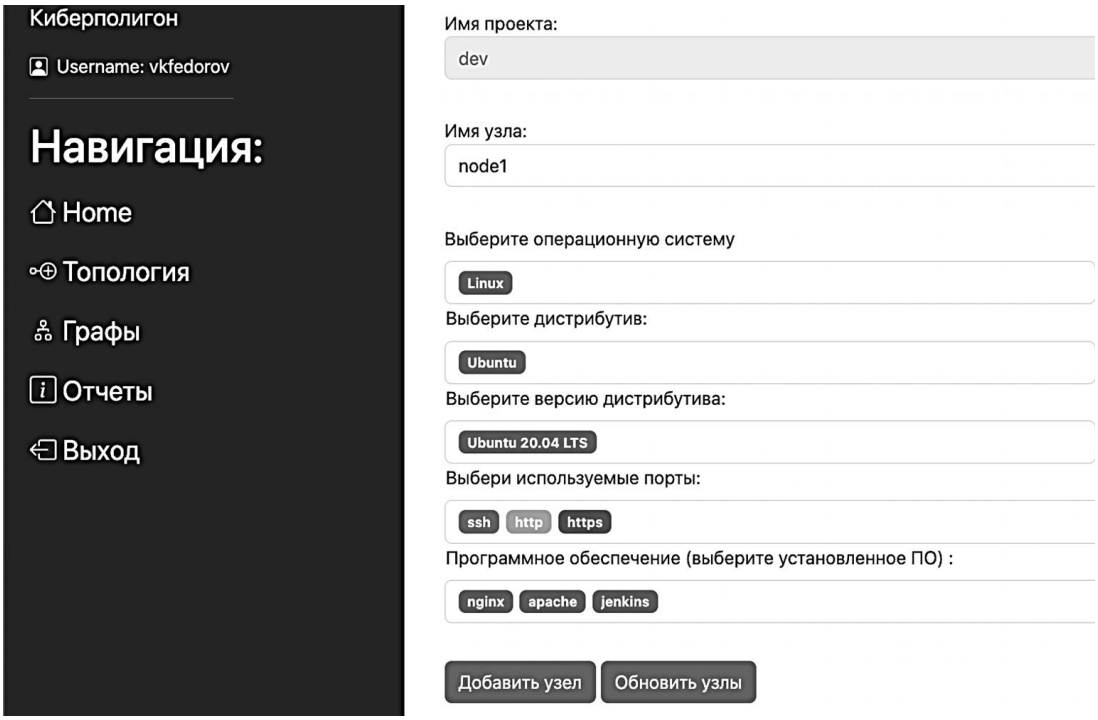

Рис. 6. Элементы интерфейса, используемые для создания нового узла

Теперь создадим топологию для будущего интерактивного графа (рис. 7) заполнив соседние узлы для каждой ноды инфраструктуры, используя интерактивную форму ввода, реализуемую «фронтенд-компонентом». После отправки запроса с заполненной топологией на стороне «бэкенд-компонента» будет вызвана функция, отвечающая за непосредственное представление динамического графа. В качестве входящих переменных используется генерируемый на стороне «фронтенд-компонента» «JSON», который содержит связи между узлами. Также есть

функция с переопределенным методом цветовой палитры графа, в качестве аргументов она принимает имена узлов атакующего и атакуемого.

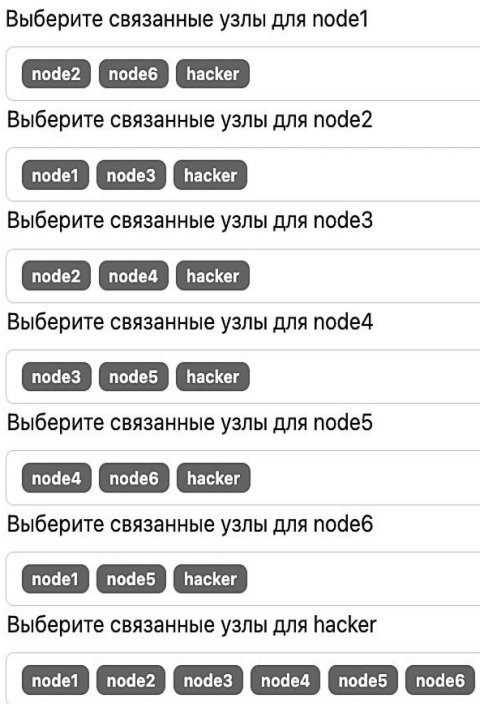

Рис. 7. Элементы интерфейса, используемые для добавления топологии узлов

После нажатия кнопки «Добавить топологию» необходимо перейти в раздел сайта «Графы». Уже непосредственно в нем можно взаимодействовать с графом: сканировать узлы, искать уязвимости и находить эксплойты к программным компонентам и дистрибутива операционных систем.

Перейдя во вкладку «Графы», увидим панель аудитора (рис.8). В данном окне можно выполнять поиск уязвимостей, эксплойтов, а также выполнять сканирование узла.

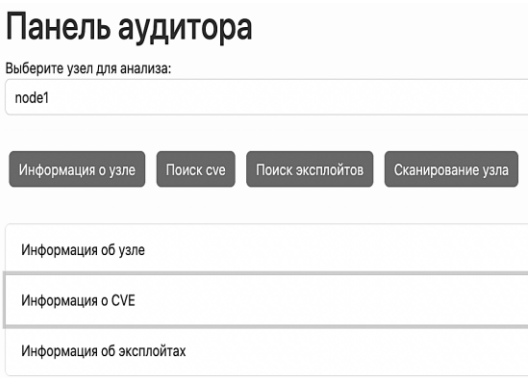

Рис. 8. Элементы интерфейса, используемые для управления графовым представлением

В правой части сайта мы увидим динамический сетевой граф, симулируемой инфраструктуры, который был построен по заданным параметрам (рис. 9).

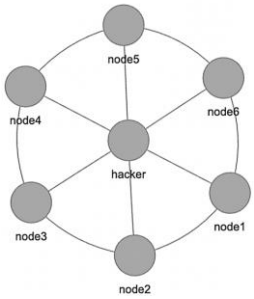

Рис. 9. Элементы интерфейса, отображающие сетевой граф симулируемой сети

До запуска сетевого сканирования мы можем вывести информацию о интересующем нас узле в «JSON» формате, например для узла «node1» (рис. 10). В форме данного бокса мы увидим информацию, заполненную ранее, а также логический проект, в котором расположена данная нода, в нашем случае это« dev».

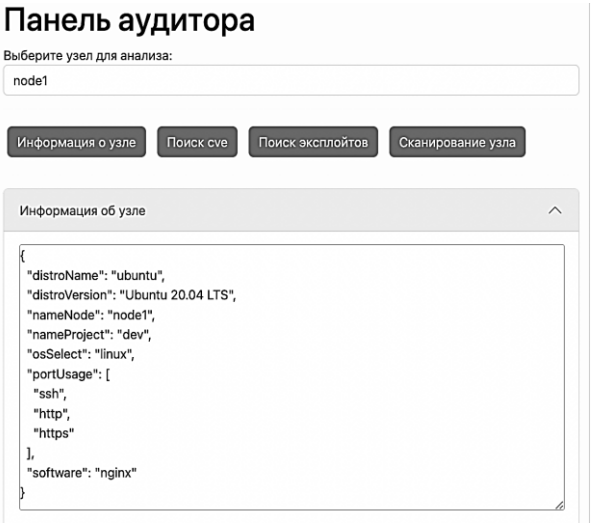

Рис. 10. Элементы интерфейса, используемые для получения информации об узле

При запуске сетевого сканирования графа у нас изменится подсветка ребра между атакующей стороной и сканируемым узлом сети (рис. 11). Теперь для каждого из узлов можно запустить сканирование. В нашем случае мы будем производить сетевое сканирование узла «node1» и искать уязвимости и эксплойты для прикладного программного обеспечения, а именно для веб-сервера «Nginx» (рис. 11 и 12).

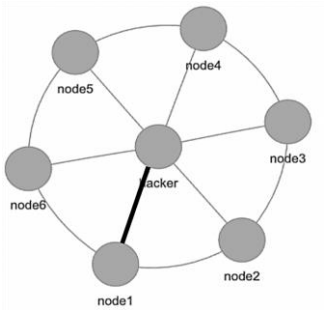

Рис. 11. Элементы интерфейса, отображающие сканирование узлов сетевого графа

При сканировании на «CVE» будет производится поиск не только прикладных программ, но и самой операционной системы, в виду того, что бывают случаи, когда уязвимость содержится на уровне операционной системы. Яркий тому пример недавняя уязвимость в ядре «Linux», под идентификатором «CVE-2023-32233».

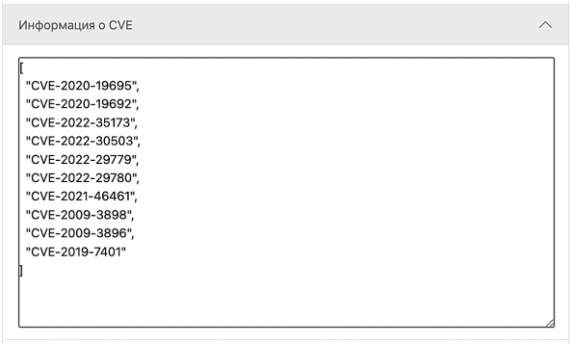

Рис. 12. Элементы интерфейса, используемые для получения информации об уязвимостях

В оснастке «Информация об эксплойтах» можно увидеть не только для каких «CVE» существуют эксплойты, но и получить описание самого эксплойта, методику его воспроизведения (рис. 13) и вектор атаки. Также можно выгрузить отчет по сканированию всей топологии сети, в нем будет содержаться информация:

- по каждому узлу (операционная система, версия, программное обеспечение),
- уязвимости и эксплойты для каждого узла («CVE», описание эксплойта, вектора атак, а также ссылка на исправление).

Например, в нашем случае были обнаружены 6 эксплойтов для уязвимостей в веб-сервере «Nginx». Из них 4 имеют сетевой вектор атаки, позволяя злоумышленнику выполнить произвольный код на удаленном узле и 2 уязвимости с вектором внутри виртуальной машины, содержащие POC код, вызывающий нарушение сегментации.

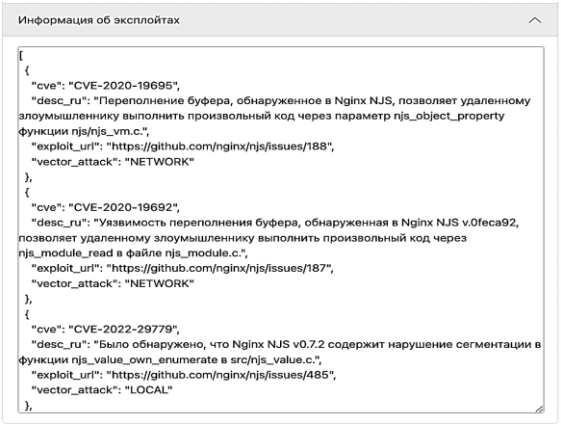

Рис. 13. Элементы интерфейса, используемые для получения информации об эксплойтах

Изучив подобный отчет, специалист по защите информации сможет в реальной информационной системе обновить уязвимые компоненты.

Заключение

В заключении хотелось бы отметить достоинства и возможные перспективы развития разработанного прототипа ПТК симуляции киберполигона:

- с точки зрения прикладного использования ПТК необходим для повышения осведомленности среди специалистов по ИБ и студентов;
- реализованный ПТК позволяет создавать инфраструктуру киберполигона с помощью симуляции процессов, не требуя больших вычислительных мощностей при работе с архитектурой симулируемой компании;
- необходима доработка ПТК в модуле загрузки графов, необходимо добавить загрузку графов в каком-нибудь формате, например «csv», «xlsx» или «xml»;
- открывается перспектива добавления функционала в компонент «OpenCve» для внешних оповещений, например «Telegram».

Список литературы

1. Актуальные киберугрозы: итоги 2022 года. URL: https://tinyurl.com/5n75tast (дата обращения 16.05.2023).

2. Информационная безопасности (рынок РФ) URL: https://clck.ru/34hzED (дата обращения 16.05.2023).

3. Хочешь мира – готовься к войне или зачем нужен киберполигон. URL: https://shorturl.at/efny7 (дата обращения 16.05.2023).

4. Ульянов А.Н. Качество плюс наглядность применение технологий виртуализации вычислительных ресурсов в информационно-образовательной среде. / А.Н. Ульянов, М.Г. Столяров, И.В. Стельмах // ВВО. 2021. № 6 (33).

5. Остапенко Г.А. Разработка архитектуры киберполигона для повышения качества и результативности учебного процесса в исследовании атак на информационные системы и сети. / Г.А. Остапенко, С.С. Куликов, А.В. Коноплин, А.А. Остапенко. // Информация и безопасность. 2023. Т. 26. Вып. 1. С. 101-108.

6. Защита от цифровых атак. URL: https://shorturl.at/mpJLU (дата обращения 16.05.2023).

7. Киберучения: Зарубежный опыт. URL:https://cyberleninka.ru/article/n/kiberucheniya-zarubezhnyy-opyt-zaschity-kriticheskoy-infrastruktury (дата обращения 16.05.2023)

8. Проектирование киберполигона в области информационной безопасности. URL: https://clck.ru/34hwzX (дата обращения 16.05.2023).

9. Обзор рынка киберполигонов. URL: https://shorturl.at/ovP01 (дата обращения 16.05.2023).

10. Черный ящик в программировании. URL: https://2no.co/2U5Jp5 (дата обращения 16.05.2023).

Воронежский государственный технический университет
Voronezh State Technical University

Поступила в редакцию 18.05.2023

Информация об авторах

Куликов Сергей Сергеевич – канд. техн. наук, доцент, Воронежский государственный технический университет, e-mail: alexanderostapenkoias@gmail.com
Федоров Василий Константинович– студент, Воронежский государственный технический университет, e-mail: f3dor0wvass@yandex.ru

CREATING A CYBER POLYGONE: GENERATING A SIMULATION BLOCK

S.S. Kulikov, V.K. Fedorov

This article describes the development and implementation of a cyber-polygon simulation unit designed for cybersecurity training and testing. As part of the work, the following were considered: a set of domestic implementations of cyber polygons, the basic principles of their functioning, an approach to its modeling was chosen, and an appropriate simulation software block was implemented. The implemented software and hardware complex allows you to dynamically fill the infrastructure, display network topologies in the form of a graph. It also provides for the generation of reports on found vulnerabilities and exploits. Working with the simulation block is carried out through the use of a web page where you can: add nodes, change the network topology, run a network scan and analyze the security report. The flexibility of the software and hardware complex is achieved by using a microservice architecture coupled with containerization technology. Experimental results have shown that the emphasis on process simulation, as well as graph manipulation, allows not only to increase the level of information security awareness, but also to effectively identify and analyze vulnerabilities, predict potential threats from the actions of intruders and take appropriate countermeasures to protect information systems.

Keywords: cyber polygons, graphs, vulnerabilities, exploits.

Submitted 18.05.2023

Information about the authors

Sergey S. Kulikov – Cand. Sc (Technical), Associated Professor, Voronezh State Technical University, e-mail: alexanderostapenkoias@gmail.com
Vasiliy K. Fedorov– Student, Voronezh State Technical University, e-mail: f3dor0wvass@yandex.ru