УДК 004.056.53

М. В. Полищук[1]*

Распределенный сканер уязвимостей на основе шаблонов

[1] Национальный исследовательский ядерный университет «МИФИ», г. Москва, Российская Федерация
* e-mail: maxpol1845@gmail.com

Аннотация. В данной статье представлен разработанный распределенный сканер уязвимостей, который функционирует на основе шаблонов и обеспечивает эффективное сканирование сетевой инфраструктуры через веб-интерфейс. Система сканера уникальна тем, что позволяет пользователям задавать и управлять задачами сканирования, используя динамически распределяемые ресурсы агентов сканирования, которые выполняют сканирование открытых портов с помощью инструмента naabu и уязвимостей через инструмент nuclei. Важной особенностью системы является возможность масштабирования числа агентов, что прямо пропорционально влияет на скорость сканирования, позволяя оптимизировать процесс в зависимости от текущих потребностей безопасности. Использование технологии сообщений rabbitmq позволяет эффективно распределять задачи между различными очередями, что способствует минимизации нагрузки на сетевое оборудование и повышает эффективность проведения сканирований из разных сегментов сети. Предложенная система не только повышает скорость и эффективность обнаружения уязвимостей, но и предлагает масштабируемое решение для крупных сетевых сред, обеспечивая тем самым значительное улучшение в области информационной безопасности. Разработка подчеркивает важность адаптивных и распределенных подходов в современной кибербезопасности и может быть интегрирована в существующие системы для обеспечения комплексной защиты.

Ключевые слова: веб уязвимости, сканирование, naabu, nuclei, шаблоны, rabbitmq

M. V. Polishyk[1]*

Distributed Vulnerability Scanner Based on Templates

[1] National research nuclear university MEPhI, Moscow, Russian Federation
* e-mail: maxpol1845@gmail.com

Abstract. This paper presents a developed distributed vulnerability scanner that functions in a template-based manner and provides efficient scanning of network infrastructure through a web-based interface. The scanner system is unique in that it allows users to set and manage scanning tasks using dynamically allocated resources of scanning agents that perform open port scanning via the naabu tool and vulnerability scanning via the nuclei tool. An important feature of the system is the ability to scale the number of agents, which has a direct proportional effect on scanning speed, allowing the process to be optimized based on current security needs. The use of rabbitmq messaging technology allows efficient distribution of tasks between different queues, which helps to minimize the load on network equipment and increases the efficiency of scans from different network segments. The proposed system not only improves the speed and efficiency of vulnerability detection, but also offers a scalable solution for large network environments, thus providing a significant improvement in information security. The development emphasizes the importance of adaptive and distributed approaches in modern cybersecurity and can be integrated into existing systems to provide comprehensive protection.

Keywords: web vulnerabilities, scanning, naabu, nuclei, templates, rabbitmq

Введение

В условиях непрерывно развивающегося цифрового мира угрозы информационной безопасности становятся всё более сложными и разнообразными. Традиционные методы обнаружения уязвимостей часто не позволяют справиться с быстро меняющимися тактиками злоумышленников, что требует разработки более гибких и масштабируемых подходов к сканированию сетей [1]. В этом контексте распределённые системы для обнаружения уязвимостей, основанные на шаблонах и параллельном сканировании, представляют собой важное направление [2] в улучшении реактивности и эффективности информационной безопасности.

Данная статья представляет разработку новой распределённой системы сканирования уязвимостей, которая интегрирует возможности настраиваемого сканирования с помощью инструментов naabu и nuclei, управляемых через Python Celery с использованием RabbitMQ в качестве брокера сообщений и MongoDB для хранения результатов. Система также включает сервер на базе FastAPI для координации задач и интерфейса пользователя, обеспечивая эффективное распределение задач между многочисленными агентами сканирования.

Особенностью предложенного решения является его способность к масштабированию и адаптации под конкретные нужды организации благодаря возможности добавления пользовательских шаблонов и расширенной конфигурации рабочих процессов. Это позволяет не только оптимизировать процессы обнаружения и реагирования на угрозы, но и значительно уменьшить время, необходимое для идентификации и устранения уязвимостей, тем самым повышая общую безопасность информационных систем.

На рынке уже существуют решения для данной задачи. В табл. 1 представлено сравнение с такими решениями как Nessus [3], OpenVas [4] и Rapid7 InsightVM [5].

Представленное в данной работе решение не только отвечает этим требованиям, но и предлагает значительные улучшения в области масштабируемости, интеграции и кастомизации. Сравнение с существующими решениями, такими как Nessus, OpenVAS и Rapid7 InsightVM, подчеркивает уникальность и преимущества нашего подхода, особенно в контексте распределенного сканирования и адаптивности к специфическим потребностям пользователей [6].

Таблица 1

Сравнительный анализ существующих на рынке решений

Характеристика	Предложенное решение	Nessus	OpenVAS	Rapid7 InsightVM
Тип лицензии	открытый исходный код	проприетарный	открытый исходный код	проприетарный
Распределенное сканирование	да, с использованием Celery/RabbitMQ	ограничено лицензией	ограничено лицензией	да
Интеграция с DefectDojo	да, через FastAPI и встроенные API	требуется модицикация DefectDojo	нет	нет
Кастомизация шаблонов	да, через пользовательские шаблоны nuclei	да, но только через вендора	да, но только через вендора	да, через модули metasploit
Веб-интерфейс	да, через FastAPI	да	да	да
Область применения	сканирование уязвимостей и открытых портов	широкий спектр уязвимостей	широкий спектр уязвимостей	широкий спектр уязвимостей
Возможность увеличения количества агентов сканирования	да	нет	нет	нет

Предлагаемое решение

Для проектирования данного решения был рассмотрен процесс работы с уязвимостями в крупнейших компаниях России. Главной проблемой является распределенность серверных мощностей. К примеру, если часть серверов, которые нужно сканировать, находится в ЦОД 1 (Центр Обработки Данных), а другая в ЦОД 2, то соединение между ними может быть или совсем заблокировано, или приводить к большей сетевой нагрузке, чем если бы сканирование производилось изнутри этого ЦОД [7]. Поэтому необходим функционал, при помощи которого можно будет создавать агентов отдельно в разных сегментах сети и распределять сканирование между ними.

Для разработки был выбран язык Python [8], поскольку он активно поддерживается сообществом, имеет широкий спектр доступных библиотек и модулей и позволяет быстро и оперативно вносить правки в код приложения в случае возникновения ошибок или для расширения функционала.

В качестве библиотеки, которая позволяет управлять задачами на агентах, был выбрана Celery [9] из-за её способности обеспечивать эффективное распределенное выполнение задач в реальном времени, поддерживая при этом масштабируемость и гибкую интеграцию с другими компонентами системы.

Также для работы приложения был необходим брокер сообщений, который будет обеспечивать взаимодействие между сервером и агентами, управлять очередями задач и собирать метрики. Был выбран RabbitMQ [10], потому что он поддерживает группировку задач и может координировать большое количество агентов в одной сети.

В конце была выбрана база данных для хранения результатов, ею стала MongoDB, поскольку она отлично взаимодействует с данными в формате json, а компоненты naabu и nuclei [11] получают результаты именно в таком формате.

Общая структура проекта представлена на рис. 1.

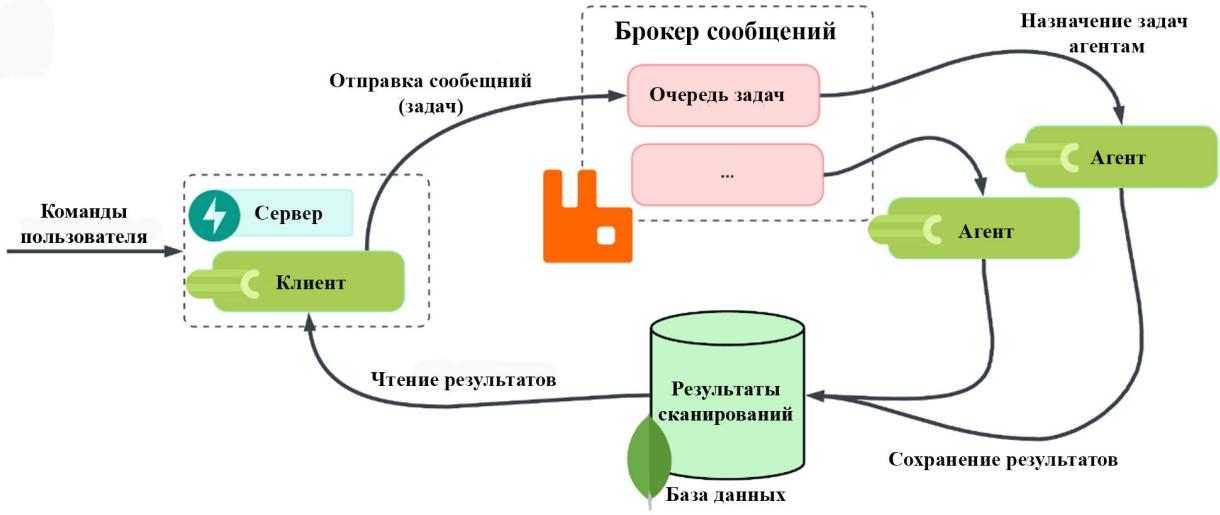

Рис. 1. Структура проекта

В проекте сканирование сети происходит в два этапа:

1) сканирование открытых портов при помощи naabu;
2) сканирование сервисов с открытыми портами при помощи nuclei.

После каждого из этапов результаты сохраняются внутри базы данных MongoDB [12], что позволяет получать отчеты как о количестве открытых различных портов в инфраструктуре компании, так и наличии уязвимостей различных уровней критичности.

Результаты

В ходе программной реализации был представлен набор программ и средства автоматизированной развертки на основе Docker и Ansible [13]. Упор делался на развертывание на серверные мощности, функционирующие на базе ОС Linux. Список реализованных компонентов:

– ansible: содержит разработанные роли и файлы конфигурации для автоматической развертки на большом количестве серверов;

– deployment: содержит ряд файлов docker [14] и docker-compose, обеспечивающих автоматическую установку зависимостей и развертку компонент решения;

– src: содержит исходный код сервера и агентов;

– nuclei-templates: содержит весь список шаблонов, которые используются для сканирования сети на предмет уязвимостей;

– scheduler: содержит файлы, обеспечивающие запуск сканирования по расписанию.

По результатам разработки была реализована интеграция с DefectDojo [15], что позволяет отслеживать и управлять всеми найденными уязвимостями, выставлять им приоритеты и по желанию редактировать описания.

Также было сформировано описание всего API (от англ. Application Programming Interface) в стандарте OpenAPI, где описаны все сущности, с которыми работает веб-сервер, и привязаны примеры запросов и ответов. Это позволяет сторонним разработчикам легко интегрироваться в данную систему. На рис. 2 представлена финальная версия API сервера.

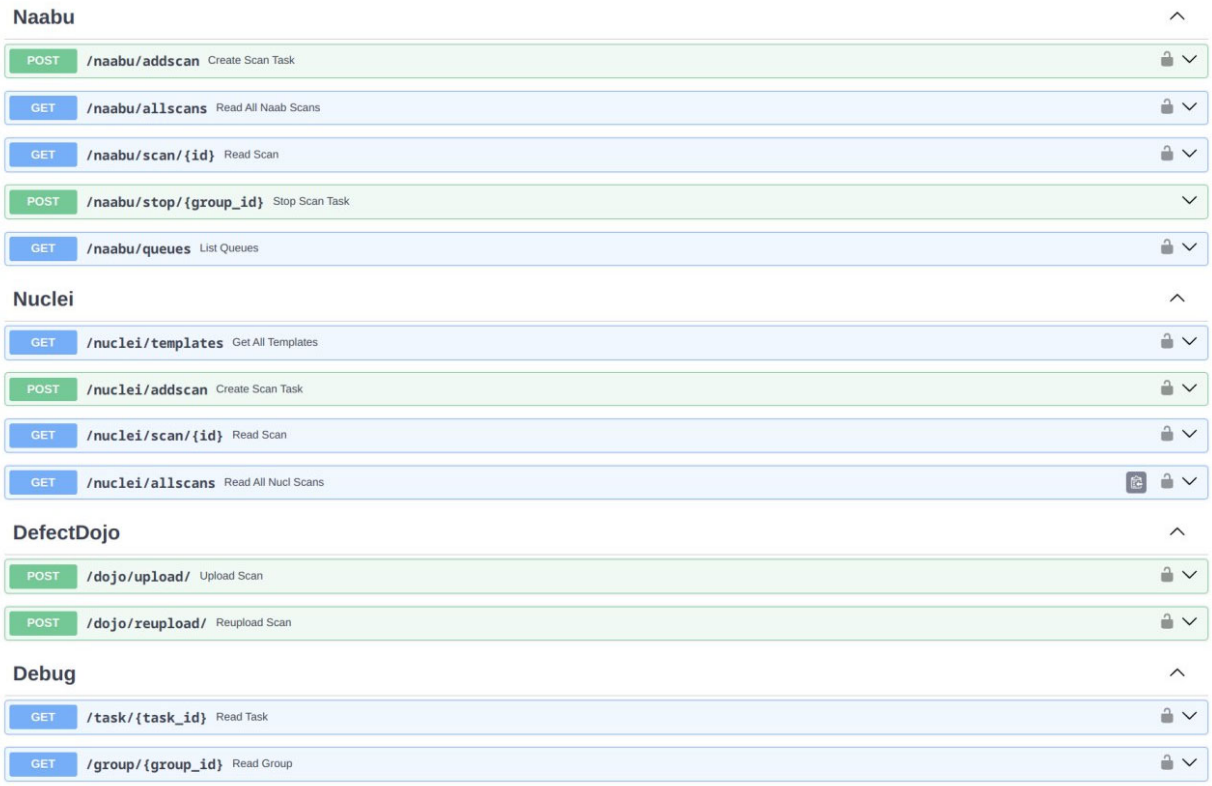

Рис. 2. Структура API веб-сервера

Решение было протестировано на инфраструктуре компании, которая имеет порядка 1 миллиона виртуальных машин во внутренней инфраструктуре. Сканирование проводилось каждый день и занимало в среднем 18 часов. На протяжении тестирования был дополнен список используемых шаблонов, увеличено количество агентов сканирования и модифицирован метод взаимодействия с DefectDojo.

Заключение

Разработанная система распределенного сканирования уязвимостей демонстрирует значительные преимущества в области гибкости, масштабируемости и кастомизации по сравнению с существующими аналогами на рынке. Применение технологий, таких как Python, Celery, RabbitMQ и MongoDB, в сочетании с возможностью добавления пользовательских шаблонов и эффективным распределением сканирования между агентами позволяет организациям адаптироваться к динамично меняющейся среде киберугроз, уменьшая время отклика на угрозы и увеличивая общую безопасность сетевой инфраструктуры.

Эффективность предложенного решения была подтверждена в ходе тестирования на инфраструктуре крупной компании, где оно успешно справилось с задачами по обнаружению и управлению уязвимостями, подчеркивая его пригодность для использования в масштабных и разнообразных сетевых средах. Интеграция с системой управления уязвимостями DefectDojo и разработка полнофункционального API обеспечивает легкую интеграцию в существующие корпоративные системы и упрощает процесс управления уязвимостями.

По мере развития проекта предполагается дальнейшее улучшение функциональности и оптимизация процессов, что сделает систему еще более адаптируемой и мощной в борьбе с киберугрозами. Благодаря своей открытости и модульности данное решение предоставляет уникальную возможность организациям эффективно обнаруживать и управлять уязвимостями в различных масштабах, адаптируясь под их конкретные потребности в кибербезопасности и защищая критически важную сетевую инфраструктуру.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ferdous J. et al. A review of state-of-the-art malware attack trends and defense mechanisms // IEEE Access. – 2023. – Т. 11. – С. 121118-121141.

2. Набатчиков Д.А. Средства анализа защищенности корпоративных сетей // Актуальные проблемы радиоэлектроники и телекоммуникаций: материалы Всерос. науч.-техн. конф. – 2012. – С. 15-17.

3. Beale J. et al. Nessus Network Auditing: Jay Beale Open Source Security Series. – Amsterdam: Elsevier, 2004.

4. Muharrom M., Saktiansyah A. Analysis of Vulnerability Assessment Technique Implementation on Network Using OpenVas // International Journal of Engineering and Computer Science Applications (IJECSA). – 2023. – Т. 2. – № 2. – С. 51-58.

5. Sahari P. Integration of vulnerability scanner reporting. – 2019.

6. Singh A. et al. Metasploit Penetration Testing Cookbook: Evade antiviruses, bypass firewalls, and exploit complex environments with the most widely used penetration testing framework. – Birmingham: Packt Publishing Ltd, 2018.

7. Басыня Е.А. Управление объектами критической информационной инфраструктуры предприятия // Защита информации. Инсайд. – 2020. – № 3. – С. 12-19.

8. Голошубова О.М., Наумов В.Ю. Python: происхождение, преимущества и перспективы // Инновационные, информационные и коммуникационные технологии. – 2016. – № 1. – С. 38-40.

9. Алтынбаев С.Т., Антаев М.П. Асинхронное выполнение задач в веб-сервисах: реализация и применение на примере библиотеки Celery. – 2021.

10. Dossot D. RabbitMQ essentials. – Birmingham: Packt Publishing Ltd, 2014.

11. Gupta D. A Critical Review of WordPress Security Scanning Tools and the Development of a Next-Generation Solution. – 2023.

12. Banker K. et al. MongoDB in action: covers MongoDB version 3.0. – New York: Simon and Schuster, 2016.

13. Басыня Е.А., Лукина М.С. Автоматизированная установка и конфигурирование серверных решений // Современные материалы, техника и технологии. – 2016. – № 2 (5). – С. 21-26.

14. Басыня Е.А. Программная реализация и исследование системы интеллектуально-адаптивного управления информационной инфраструктурой предприятия // Вестник Самарского государственного технического университета. Серия: Технические науки. – 2020. – № 1 (65). – С. 6-21.

15. Bernardo G. DevSecOps pipelines improvement: new tools, false positive management, quality gates and rollback. – Turin: Politecnico di Torino, 2022.