

В. Д. Злочевский¹✉

Метод сетевого зондирования объектов корпоративной вычислительной сети

¹Национальный Исследовательский Ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: vladzlotchevskiy@mail.ru

Аннотация. В статье рассматривается задача сетевого зондирования объектов корпоративной вычислительной сети для обнаружения уязвимостей и некорректных настроек сервисов, служб и средств защиты информации. Актуальность исследования обусловлена наличием требований к зондированию и поиску уязвимостей со стороны государства и индустрии информационной безопасности, а также недостатками существующих решений: низкая точность определения версий операционных систем в режиме «черного ящика» и отсутствие проверок на возможный спуфинг портов. Цель исследования заключается в разработке системы зондирования, обеспечивающей сбор информации о сетевых узлах, открытых портах, версиях служб и операционных систем, а также наличии средств защиты информации и возможных ошибок конфигураций в них. В результате работы был предложен метод сетевого зондирования и представлен в виде блок-схемы и создана база данных системы сетевого зондирования. Предлагаемый метод позволит снизить число ошибок первого рода при возможном использовании систем сокрытия сервисов.

Ключевые слова: сетевое зондирование, обнаружение уязвимостей, активный анализ сети, определение версий служб и сервисов

V. D. Zlochevskii¹✉

Method for network security probing of corporate network assets

¹National Research Nuclear University MEPhI, Moscow, Russian Federation
e-mail: vladzlotchevskiy@mail.ru

Abstract. The article addresses the task of network probing of corporate computing network assets for detecting vulnerabilities and misconfigurations of services, daemons, and network security systems. The relevance of the study is due to the presence of requirements for information gathering and vulnerability detection from the state and the information security industry, as well as the shortcomings of existing solutions: low accuracy of operating system version identification in black-box mode and the lack of checks for possible port spoofing. The aim of the study is to develop a probing system that provides collection of information about network nodes, open ports, versions of services and operating systems, as well as the presence of information security tools and possible configuration errors within them. Because of the work, a network probing method was proposed and presented in the form of a block diagram, and a database for the network probing system was created. The proposed method will reduce the number of type I errors when using service concealment systems.

Keywords: network probing, vulnerability detection, network active analysis, service version identification

Введение

Информационная безопасность является состоянием защищенности информации от несанкционированного доступа, изменения или уничтожения, соответственно процесс ее обеспечения обязан носить непрерывный и системный характер. Для оценки уровня защищенности информационных систем применяют различные методы анализа сетевого трафика, среди которых два основных: активный и пассивный. Активный анализ предполагает взаимодействие с целевой системой. На этом принципе работают инструменты тестирования на проникновение и сетевого зондирования. Под сетевым зондированием понимается процесс, при котором имитируются различные атаки и отслеживается реакция объекта с идентификацией эшелонов его защиты [1].

Актуальность темы исследования со стороны индустрии информационной безопасности обусловлена наличием пунктов, связанных с зондированием и поиском уязвимостей в общепризнанных мировых методологиях тестирования на проникновения, таких как: OWASP [2], OSSTMM [3], NIST 800-115 [4] и PTES [5].

В научной среде существует несколько групп, изучающих методы активного анализа трафика и зондирования устройств на наличие уязвимостей и некорректных настроек служб, сервисов и средств защиты информации. Одна из них во главе с создателем утилиты nmap Лионом [6] в своих работах предлагают варианты сканирования устройств, определения версий служб, сервисов и операционных систем с помощью специально сформированных пакетов данных. Основные проблемы существующих решений в данной области связаны с низкой точностью определения семейства операционных систем в режиме «черного ящика» и их непосредственной версии из-за схожести получаемых данных и регулярных обновлений. Также, авторы целенаправленно не используют методы пассивного анализа трафика и не имеют модулей для проверки безопасности настройки средств защиты информации и определения их вендора, что осложняет автоматизированный подбор возможных полезных нагрузок специалистами по тестированию на проникновение. Соответственно, в работе предлагается совместить методы активного и пассивного анализа трафика, что может повысить метрики обнаружения операционных систем как минимум при нахождении в локальной сети.

Одной из проблем активного анализа трафика является спуфинг портов – процесс, при котором автоматизированными системами [7], либо с помощью правил iptables и xtables-addons имитируются сервисы на закрытых портах. Существующие сканеры уязвимостей и системы сетевого зондирования не реализуют дополнительные проверки на данный тип противодействия, что понижает точность определения версий сервисов.

Как было отмечено ранее, одной из задач зондирования является идентификация эшелонов защиты. В работах [8, 9] исследуются межсетевые экраны и их реакция на определенные виды запросов. Из основных недостатков можно отметить искажение результатов по причине сетевых задержек, сложности их идентификации при изучении методом «черного ящика» и ограниченности данных

алгоритмов, так как уязвимыми является лишь определенный тип межсетевых экранов.

В связи с расширением процесса сетевого зондирования и комбинации алгоритмов пассивного и активного анализа, были рассмотрены основные работы в сфере обнаружения версий операционных систем методами пассивного анализа трафика. Так, в работе научной группы Ластовички [10] разбираются основные методы пассивного анализа трафика для определения возможной операционной системы устройства и выделяются основные характеристики протоколов канального, транспортного и прикладного уровня модели OSI. Также, были исследованы работы [11, 12] и анализ TLS-шифров до версии 1.2 включительно был добавлен в список значений для сбора статистики об операционной системе.

Итоговая цель работы заключается в разработке метода и системы сетевого зондирования объектов корпоративной вычислительной сети для обнаружения уязвимостей и некорректных настроек сервисов и служб. Основными задачами исследования являлись: разработка собственного метода сетевого зондирования, сочетающего в себе алгоритмы пассивного и активного анализа трафика и создание собственной базы данных, содержащей в себе основные характеристики используемых сервисов, служб, операционных систем и средств защиты информации.

Предлагаемый метод сетевого зондирования

Итоговый метод сетевого зондирования представлен на рис. 1 в виде блок-схемы. Параллельно ведется пассивный сбор метрик, таких как:

- 1) TTL, TOS, DF;
- 2) MAC-адреса;
- 3) порядок TCP-опций и размер окна TCP;
- 4) HTTP User-Agent, набор шифров TLS (до версии 1.2 включительно);
- 5) действующие в сети протоколы.

В качестве алгоритма поиска активных устройств используются классические алгоритмы ICMP Echo Request и Timestamp Request, в локальных сетях отправляется ARP-запрос, а для внешних сетей используется отправка TCP-SYN на 22, 80 и 443 порты и ожидания реакции на него. Открытые порты ищутся классическими методами – полуоткрытое TCP-сканирование, суть которого заключается в отправке SYN-запроса на интересующий порт. В случае, если сервер отвечает SYN-ACK, то считаем, что порт открыт и посылаем RST, тем самым завершая соединение без полноценного установления. Для определения открытого UDP-порта используется некоторый набор заготовленных дейтаграмм и ожидание ответа от целевых портов.

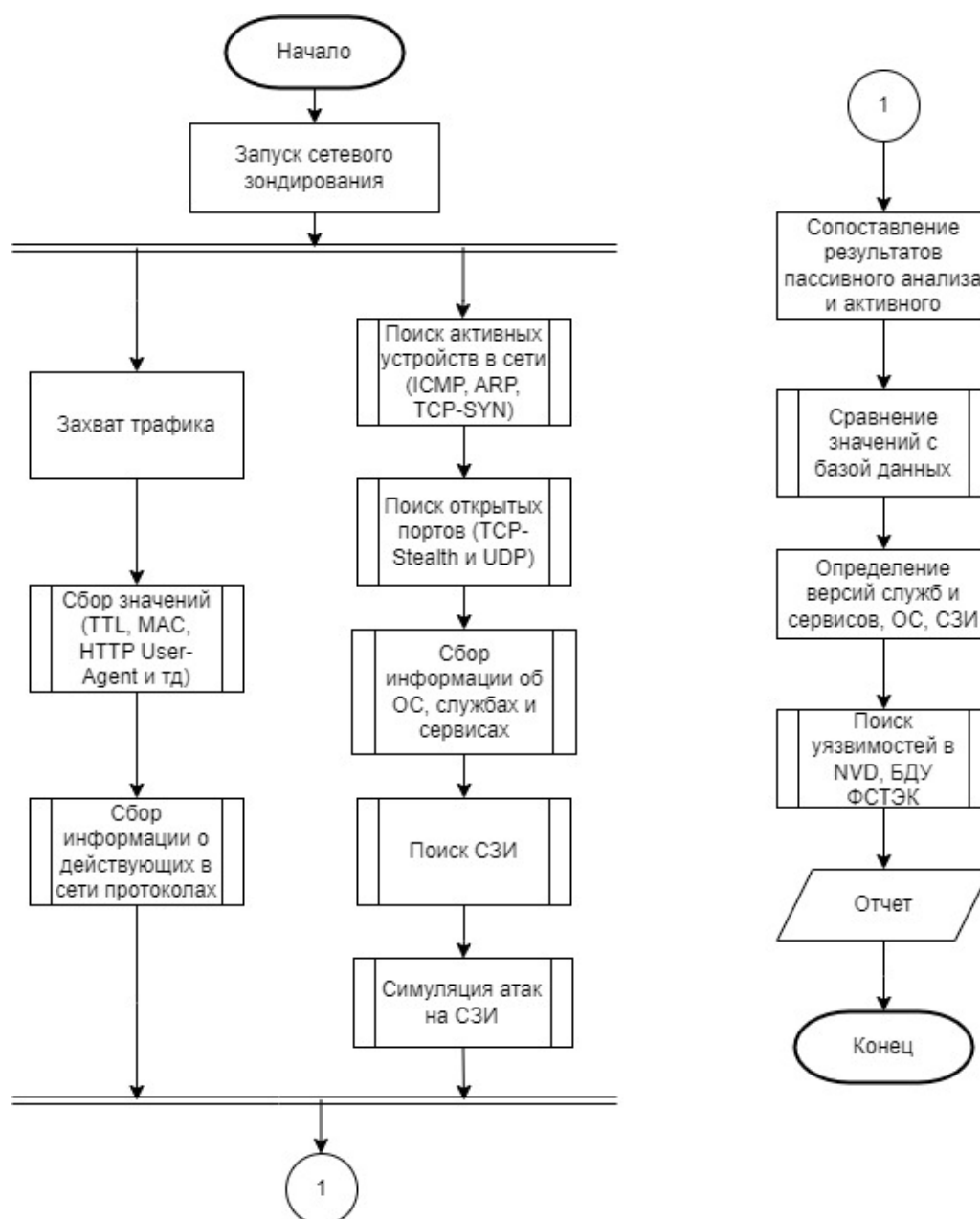


Рис. 1. Блок-схема метода сетевого зондирования объектов корпоративной вычислительной сети

Под сбором информации об ОС, службах и сервисах подразумевается алгоритм отправки специально сформированных запросов для разных сервисов, где также учитывается возможный спуфинг портов (рис. 2). Повторный запрос баннера сервиса через случайное время и с измененными параметрами подключения позволит обмануть системы автоматической маскировки портов, тем самым повысив точность их определения за счет понижения числа ложноположительных срабатываний сканера.

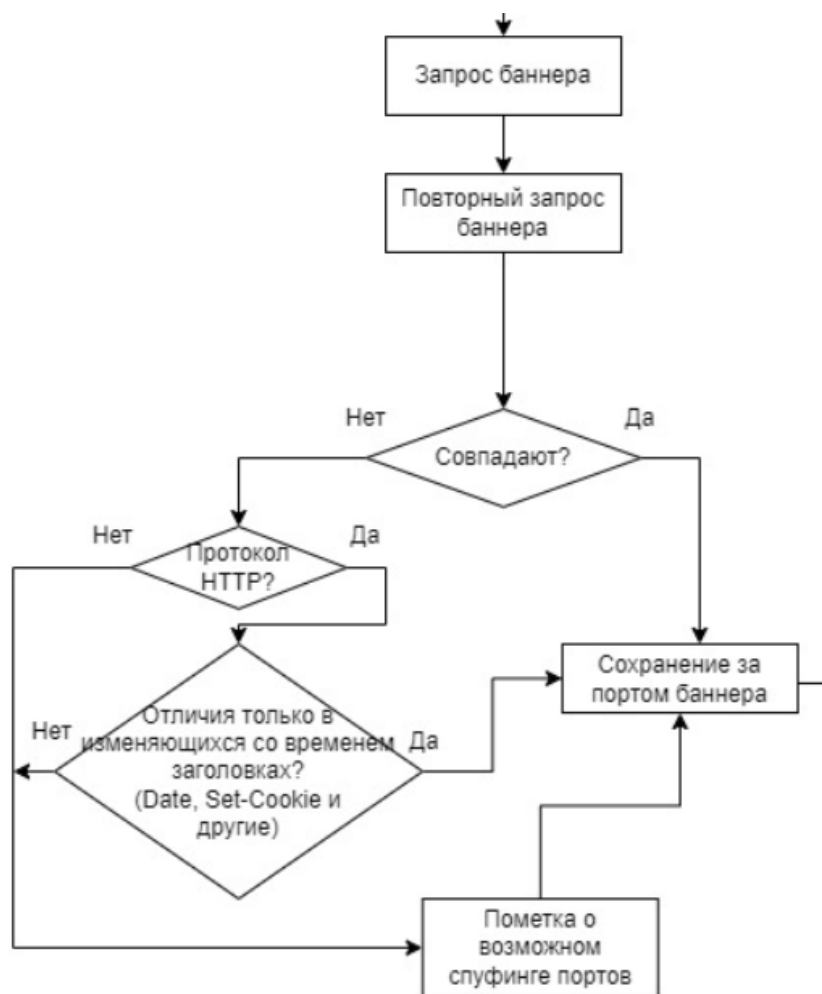


Рис. 2. Часть блок-схемы алгоритма сборки информации о сервисах, связанная с проверкой на спуфинг портов

В метод дополнительно заложены алгоритмы поиска уязвимостей и некорректных настроек часто используемых сервисов. В их число входит DNS, IKE, SNMP, SMTP и ряд других (всего на данный момент 20). Как пример, представлена блок-схема алгоритма проверки DNS на основные ошибки конфигурации (рис. 3). Поочередно производятся проверки на уязвимость к распределенной атаке в обслуживании, раскрытия информации через трансфер зон, отравление кэша и раскрытия информации через этот кэш. Также, их можно реализовать параллельно, тем самым повысив скорость проверки DNS-сервера, но повышает нагрузку на него.

Стоит отметить, что поиск CVE или уязвимостей из БДУ ФСТЭК происходит уже на этапе после сопоставления результатов пассивного и активного анализа, то есть в данном случае даже если в базе данных системы не будет какой-то записи с версией целевого сервера, либо если система не сможет определить ее, то основные проверки будут протестированы.

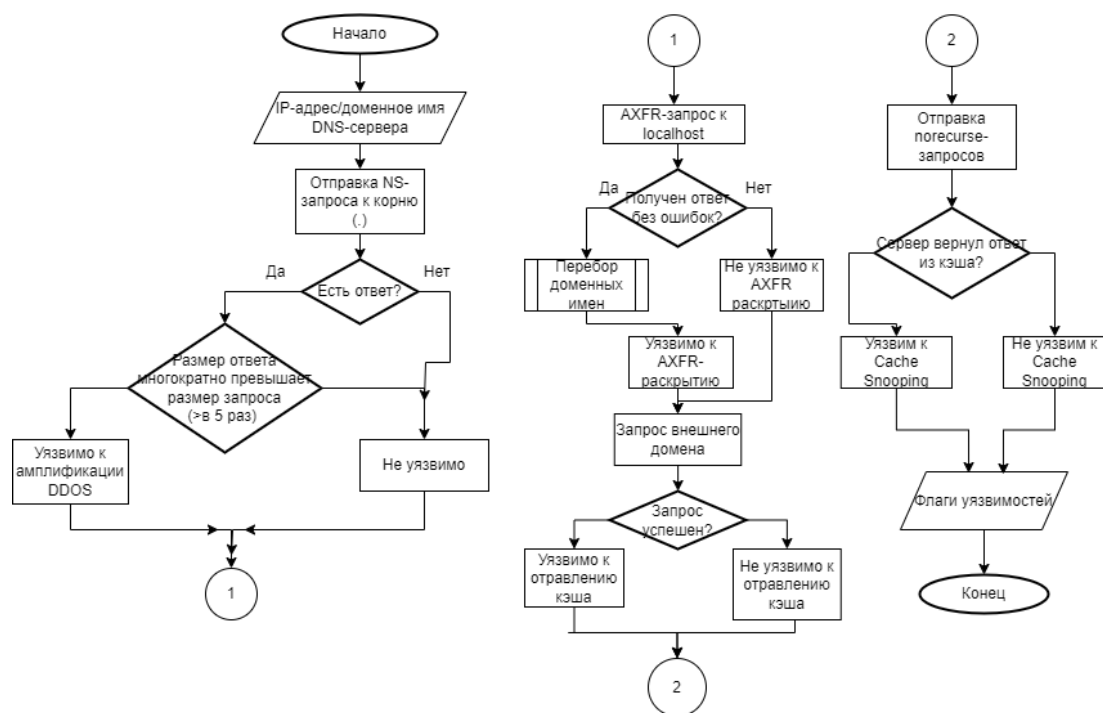


Рис. 3. Алгоритм проверки DNS-сервера на основные ошибки конфигурации

На текущий момент предложено два алгоритма поиска СЗИ: это межсетевые экраны и WAF. Поиск WAF заключается в отправке набора вредоносных полезных нагрузок (межсайтовый скриптинг, SQL-инъекция и другие) и анализ HTTP-ответов. Алгоритм же поиска и сбора метаданных межсетевого экрана представлен на рис. 4. Одна из симулируемых атак помечена отдельным процессом – BlackNurse. По ее успешности возможно определить stateful или stateless межсетевого экран, а также есть возможность узнать его вендора вплоть до версии ОС внутри.

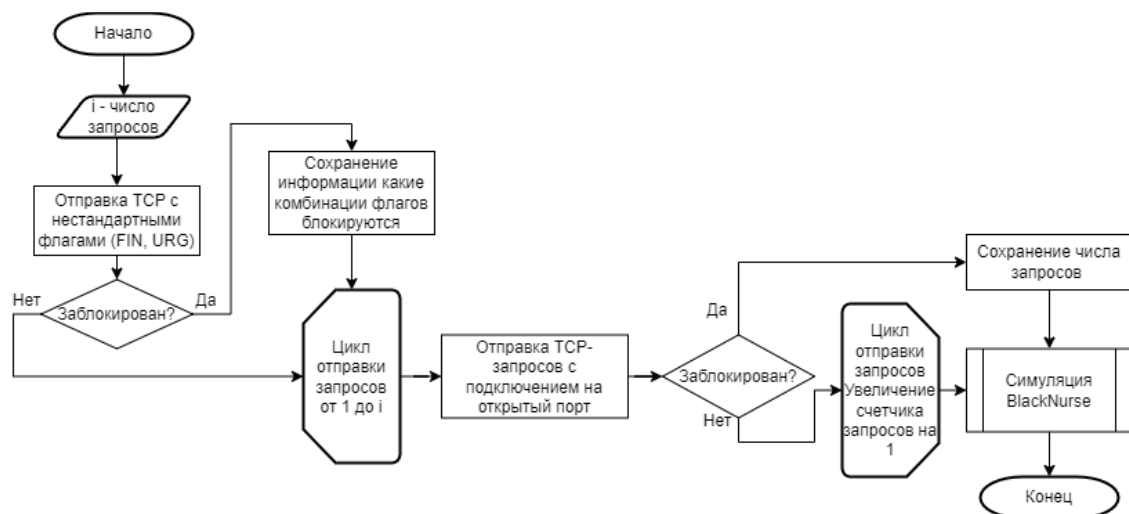


Рис. 4. Блок-схема алгоритма поиска межсетевых экранов

Стоит отметить, что в случае отсутствия четких показателей определенной операционной системы, в системе заложен поиск операционной версии по информации, полученной из открытых портов. Например, у Ubuntu отличаются пакеты (ssh, openssl) в зависимости от ее минорной версии, что было использовано в качестве дополнительного фактора при определении ОС.

Обсуждение

В отличие от лежащих в основе ряда существующих инструментов сетевого зондирования и сканеров уязвимостей, предложенный метод реализует собственные алгоритмы анализа ответов сетевых сервисов и проверки их поведения. Особое внимание в текущей работе уделяется механизму обнаружению признаков спуфинга портов и маскировки сетевых сервисов, что повышает точность определения версий за счет понижения числа ошибок первого рода. Также, созданная база данных содержит записи с информацией об отечественных средствах защиты информации, что отсутствуют во всех аналогичных инструментах в открытом доступе. Пассивный анализ предлагается использовать при возможности внедрения сниффера в исследуемую сеть – это возможно сделать при работах аудита и тестировании на проникновения при дополнительном согласовании. В случае невозможности внедрения, система может работать как классический сетевой зондер, используя исключительно методы активного анализа.

Заключение

В ходе исследования была рассмотрена задача сетевого зондирования объектов корпоративной вычислительной сети, необходимой для поиска уязвимостей и мисконфигураций сервисов и служб.

В результате работы был предложен метод сетевого зондирования объектов корпоративной вычислительной сети для поиска уязвимостей сервисов и служб, сочетающий в себе активный и пассивный анализ трафика. Также, была создана база данных системы, содержащей в себе более 1000 записей с известными веб-серверами, почтовыми сервисами, операционными системами и средствами защиты информации.

На текущем этапе разработан программный прототип системы зондирования, работающей на основе предложенного метода и проведено ее первичное тестирование в тестовой среде, подтвердившее ее работоспособность и корректность взаимодействия компонентов. Однако, подробные результаты исследования будут представлены в дальнейшем после завершения исследования ее работы по сравнению с другими решениями. В дальнейшем, возможна интеграция новых алгоритмов, связанных с поиском скрытых сервисов и межсетевых экраном, а также обогащение базы данных системы сетевого зондирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Басыня Е.А. Сетевая информационная безопасность: Учебник. – М.: НИЯУ МИФИ, 2023. – 224 с.
2. OWASP Testing Guide. Version 4.2. – 2020. – URL: https://www.owasp.org/index.php/OWASP_Testing_Project (дата обращения 14.11.2025)

3. Herzog P. OSSTMM – The Open Source Security Testing Methodology Manual. – New York: 2006. – 129 с. – URL: <https://www.isecom.org/OSSTMM.3.pdf> (дата обращения: 13.11.2024)
4. NIST 800-115. Technical Guide to Information Security Testing and Assessment - NIST, 2008
5. PTES Technical Guidelines: Penetration Testing Execution Standard [Электронный ресурс]. — URL: <http://www.pentest-standard.org/>. — PTES — (Дата обращения: 21.03.2026).
6. Lyon G. F. Nmap network scanning: The official Nmap project guide to network discovery and security scanning. – Insecure, 2009.
7. Piotr Duszyński. Portspooof. - URL - <https://github.com/drklwi/portspooof> (дата обращения 20.03.2026)
8. Krishna T. V., Karthik P. Dominance of hardware firewalls and denial of firewall attacks (case study blacknurse attack) //International Journal of Science and Research (IJSR). – 2022. – Т. 11. – С. 28-33.
9. Liu A. X. et al. Firewall fingerprinting and denial of firewalling attacks //IEEE Transactions on information forensics and security. – 2017. – Т. 12. – №. 7. – С. 1699-1712.
10. Laštovička M. et al. Passive operating system fingerprinting revisited: Evaluation and current challenges // Computer Networks. – 2023. – Т. 229. – С. 109782
11. Hagos D. H. et al. Advanced passive operating system fingerprinting using machine learning and deep learning //2020 29th International Conference on Computer Communications and Networks (ICCCN). – IEEE, 2020. – С. 1-11
12. Muehlstein J. et al. Analyzing HTTPS encrypted traffic to identify user's operating system, browser and application // 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC). – IEEE, 2017. – С. 1-6

© В. Д. Злочевский, 2026