ПРОГРАММНОЕ СРЕДСТВО ОБНАРУЖЕНИЯ УЯЗВИМОСТЕЙ В ЛОКАЛЬНОЙ СЕТИ

Луцкевич А.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Капанов Н.А. – старший преподаватель

В представленной работе рассматриваются современные подходы к разработке прикладных инструментальных программных средств автоматизированного мониторинга и анализа полученной в ходе этого процесса информации.

На сегодняшний день обеспечение информационной безопасности локальных компьютерных сетей предприятий становится все более критичной задачей в условиях постоянного роста числа киберугроз. Область автоматизированного сканирования и анализа уязвимостей развивается более двадцати лет. На рынке существуют мощные коммерческие решения, такие как Nessus Professional, MaxPatrol VM и OpenVAS, каждое из которых обладает набором достоинств и недостатков. Современные коммерческие решения демонстрируют высокие показатели точности обнаружения уязвимостей благодаря регулярно обновляемым базам данных, содержащим информацию о известных угрозах.

Предназначение рассматриваемого программного средства – автоматизированный анализ и выявления уязвимостей в локальных компьютерных сетях предприятий.

Основными его функциями являются:

- сканирование сетевого диапазона с целью обнаружения активных хостов;
- определение открытых портов и запущенных сервисов на обнаруженных хостах;
- выявление известных уязвимостей в операционных системах и сетевых сервисах на основе сопоставления с базой CVE;
- анализ конфигурации сетевых служб на предмет проблем безопасности;
- классификация обнаруженных уязвимостей по уровню серьёзности (критичная, высокая, средняя, низкая);
- хранение истории сканирований в локальной базе данных; экспорт результатов в форматы Microsoft Word, Excel и CSV; управление модулями расширения (плагинами) для анализа уязвимостей;

Входными данными для программного средства являются:

- диапазон IP-адресов целевой сети в формате CIDR или явного диапазона;
- учётные данные администратора для проведения аутентифицированного сканирования (при необходимости).

Выходными данными являются:

- отчёты о выявленных уязвимостях с указанием IP-адреса хоста;
- описания уязвимости:
  - идентификаторы CVE,
  - уровень серьёзности,
  - рекомендации по устранению;
- экспортированные отчёты в форматах Word, Excel и CSV.
-

Так, на рисунке 1 представлена таблица с результатами сканирования.

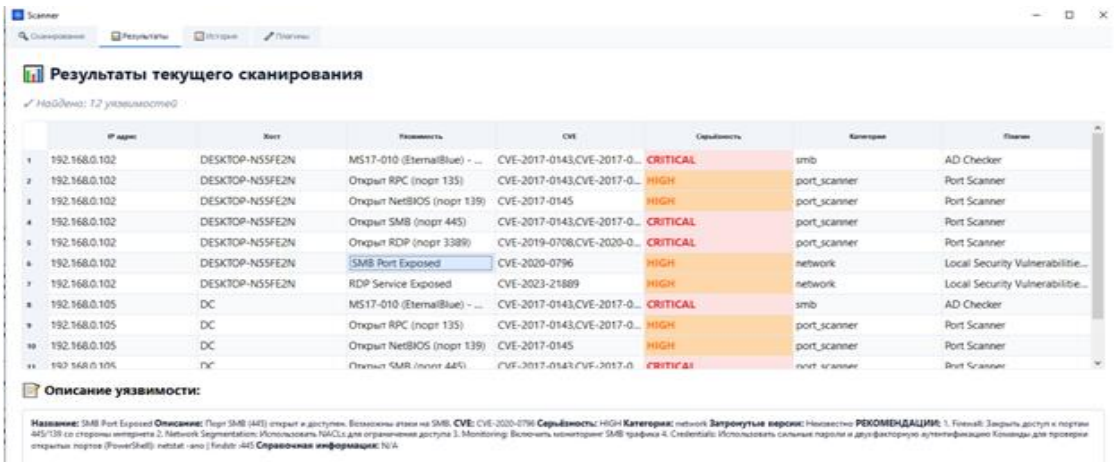

Рисунок 1 – Таблица результатов сканирования со списком обнаруженных уязвимостей

Для реализации программного средства выбран язык программирования Python,как основной инструмент разработки. Этот выбор обусловлен тем, что PyPI имеет тысячи пакетов для информационной безопасности, от nmap до virustotal-api. Это значит, что ресурсы и туториалы всегда под рукой [1]. Это качество особенно важно для целевой аудитории проекта – администраторов и инженеров, которые могут не иметь опыта в профессиональной разработке программного обеспечения. Поэтому Python в настоящее время является де-факто стандартом в области кибербезопасности именно благодаря наличию специализированных библиотек. Язык позволяет автоматизировать сканирование на наличие уязвимостей в системах и приложениях, что прямо соответствует целям проекта. Обширная экосистема библиотек (Impacket, python-nmap, Scapy, ldap3) позволяет избежать переделывания уже решённых задач и сосредоточиться на специфичной функциональности для анализа, например Active Directory.

Python работает кроссплатформенно на Windows, Linux и macOS без требования переписывания кода, что соответствует требованию потенциальной кроссплатформенности. Кроме того, активное сообщество разработчиков и специалистов по информационной безопасности обеспечивает постоянное развитие инструментов, доступность примеров кода и решения нестандартных задач.

Для реализации пользовательского интерфейса выбран фреймворк PyQt5, основанный на Qt Framework. PyQt5 был выбран как оптимальное решение среди доступных альтернатив. Библиотека обеспечивает создание профессиональных приложений с нативным внешним видом под операционную систему, на которой работает приложение.

Интересной особенностью PyQt5 является встроенная поддержка многопоточности через класс QThread. Это позволяет запускать длительные операции сканирования сети в отдельном потоке, предотвращая зависание графического интерфейса и обеспечивая удобство работы пользователя. PyQt5 предоставляет «богатый набор виджетов для создания интерфейса на Python», включая таблицы (QTableWidget) для отображения результатов, прогресс-бары (QProgressBar), диалоги и другие компоненты, необходимые для системы управления уязвимостями.

Дополнительное преимущество PyQt5 – наличие инструмента Qt Designer, позволяющего создавать интерфейсы визуально методом «drag-and-drop» без необходимости писать весь код вручную. Это существенно ускоряет процесс разработки интерфейса и снижает риск ошибок в расположении элементов.

Функциональность системы определяется как обязательными, так и опциональными функциями. Обязательные функции представляют функциональность, которая должна быть выполнена в процессе основного сценария использования. Опциональные функции предоставляют дополнительные возможности, которые расширяют функциональность основных вариантов использования, но не являются критичными для выполнения системой основного предназначения [2]. Такое разделение функциональности позволяет приоритизировать разработку и обеспечивает гибкость при расширении возможностей системы.

Основные сценарии использования системы включают процесс инициирования и выполнения сканирования сетевого диапазона с автоматическим обнаружением активных хостов, определением открытых портов и работающих сервисов, выявлением известных уязвимостей путём сопоставления с базой данных CVE, классификацией обнаруженных проблем по уровню серьёзности и предоставлением рекомендаций по их исправлению.

Дополнительный сценарий предусматривает сохранение результатов сканирования в базу данных для ведения истории проверок, возможность экспорта результатов в различные форматы (CSV, Excel, Word) для использования в других информационных системах, а также просмотр и анализ результатов предыдущих сканирований для отслеживания динамики изменения уязвимостей во времени.

Список использованных источников:

1. Средства защиты информации, общие сведения [Электронный ресурс]. – Режим доступа: https://www.oac.gov.by/activity/information-security-tools/technical-and-cryptographic-information-protection/general-information/.

2. Буч, Г. Язык UML. Руководство пользователя: практическое руководство / Г. Буч, Дж. Рамбо, И. Якобсон; пер. с англ. Н. Мухина. – 3-е изд. – М.: ДМК Пресс, 2022. – 495 с.