УДК 004.056.53

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЦЕНТРАЛИЗОВАННОГО УСТРАНЕНИЯ УЯЗВИМОСТЕЙ ХОСТОВ ПОД УПРАВЛЕНИЕМ UNIX-ПОДОБНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

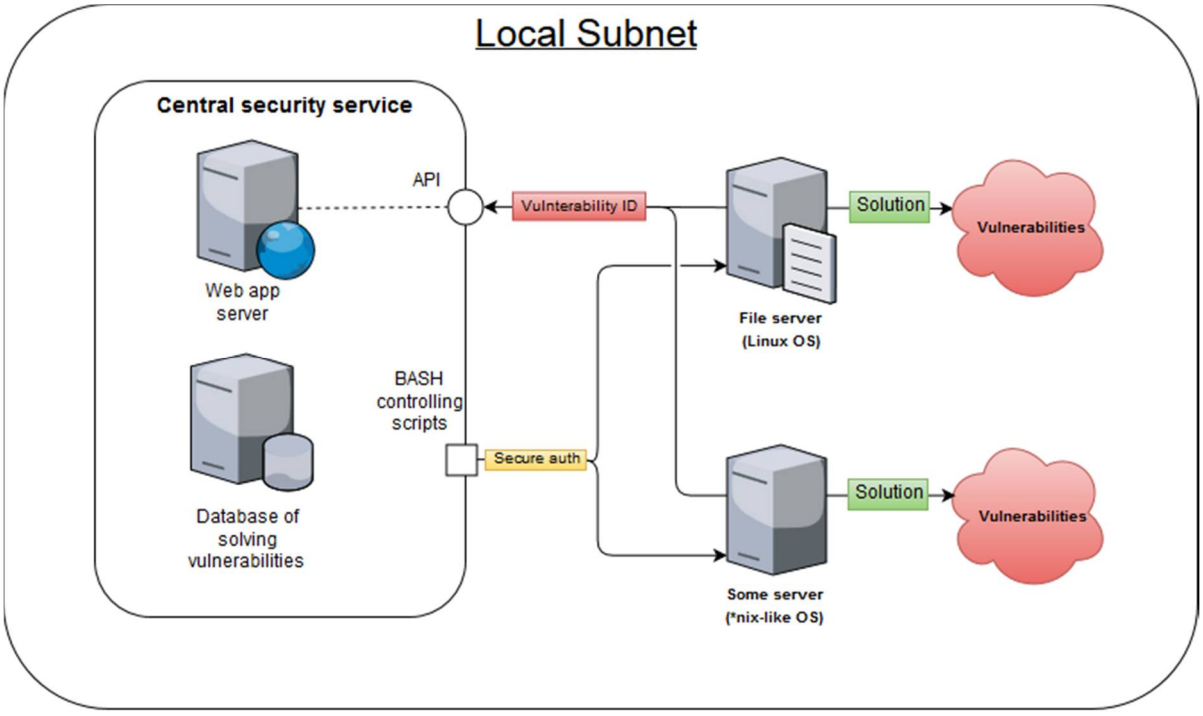
Статья поступила в редакцию 20.04.2019, в окончательном варианте – 12.05.2019.

Горкавенко Владимир Сергеевич, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а
магистрант, e-mail: slipmetal@mail.ru
Ажмухамедов Искандар Маратович, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а
доктор технических наук, профессор, заведующий кафедрой информационной безопасности, e-mail: iskander_agm@mail.ru

Рассмотрено одно из возможных решений задачи защиты от несанкционированного доступа рабочих станций под управлением Unix-подобных операционных систем, находящихся в одной подсети, посредством разработки централизованной системы устранения уязвимостей, позволяющей повысить уровень защищенности. Описано разработанное программное обеспечение для централизованного устранения уязвимостей хостов под управлением unix-подобных операционных систем. Составлена и описана структурная схема централизованной системы устранения уязвимостей. Перечислены основные средства программирования, использованные для реализации задачи разработки: язык программирования, фреймворки, СУБД. Описан алгоритм функции анализа и отправки данных, а также структура таблиц базы данных. Представлено описание основных интерфейсов для взаимодействия специалиста информационной безопасности и системного администратора с централизованной системой устранения уязвимостей. Составлена диаграмма последовательности действий акторов.

Ключевые слова: разработка системы защиты, системный администратор, информационная безопасность, защита информации, централизованная защита, Unix, linux, Ubuntu

Графическая аннотация (Graphical annotation)



DEVELOPMENT OF SOFTWARE FOR CENTRALIZED ELIMINATION OF HOST VULNERABILITIES UNDER THE MANAGEMENT OF UNIX-LIKE OPERATIONAL SYSTEMS

The article was received by editorial board on 20.04.2019, in the final version – 12.05.2019.

Gorkavenko Vladimir S., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation,

undergraduate student, e-mail: slipmetal@mail.ru

Azhmukhamedov Iskandar M., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation,

Doct. Sci. (Engineering), Professor, Head of the Department of Information Security, e-mail: iskander_agm@mail.ru

We consider one of the possible solutions to the problem of protecting unauthorized access of workstations running unix-like operating systems located on the same subnet through the development of a centralized vulnerability mitigation system that allows to increase the level of security. The developed software for centrally addressing host vulnerabilities under unix-like operating systems is described. A block diagram of a centralized system for eliminating vulnerabilities has been compiled and described. The main programming tools used to accomplish the development task are listed: programming language, frameworks, DBMS. The algorithm for analyzing and sending data is described, as well as the structure of database tables. A description of the main interfaces for the information security specialist and the system administrator with a centralized vulnerability management system is presented. A diagram of the sequence of actions of actors.

Keywords: protection system development, system administrator, information security, information protection, centralized protection, unix, linux, Ubuntu

Введение. В условиях интенсивно развивающейся отрасли информационных технологий необходимо развивать средства повышения уровня защиты информации [6, 7].

На текущий момент существует множество программных решений для устранения уязвимостей операционных систем (ОС) [9], но они имеют высокую стоимость для развертывания их в малых и средних коммерческих организациях [3]. Также ни одно из таких программных решений не имеет функциональной возможности централизованного устранения уязвимостей в подсети хостов под управлением Unix-подобных ОС.

Из вышесказанного можно сделать вывод, что разработка централизованной системы устранения уязвимостей (ЦСУУ) подсети хостов под управлением Unix-подобных ОС является весьма актуальной задачей.

Ранее в работе [2] была приведена формализация задачи, предложено реализовать пять основных этапов, которые будет включать в себя алгоритм поиска и применения решения для устранения диагностированных уязвимостей на рабочих станциях. Это позволило разработать логическую структуру, построить проектные диаграммы, описать взаимодействия основных модулей и сервисов [2]. Предлагаемая в работе структура ЦСУУ позволила перейти к программной реализации, поэтому ***целью данной статьи*** является описание основных интерфейсов и соответствующих им модулей разработанного программного обеспечения (ПО) ЦСУУ.

Описание разработанного программного обеспечения. Для описания разработанного ПО необходимо составить структурную схему ЦСУУ (рис. 1).

В качестве сервиса установки защищенного соединения между центральным сервером и хостами-клиентами используется протокол SSH [2]. Использование данного протокола возможно только с помощью консольного интерпретатора команд ОС. Для совместимости командных интерпретаторов центрального сервера и хоста-клиента необходимо, чтобы сервер работал под управлением Unix-подобной операционной системы [14]. В качестве одного из возможных вариантов такой ОС может быть использована Ubuntu [1, 11].

Рисунок 1 – Структурная схема ЦСУУ

Функции анализа и отправки данных в ЦСУУ о диагностированных уязвимостях хоста выполняет BASH скрипт-анализатор [5]. С помощью разработанного скрипта по протоколу SSH [2] ЦСУУ удаленно выполняет следующую последовательность действий:

- устанавливает утилиту для реализации http(s) подключения;
- устанавливает анализатор Lynis или обновляет его до последней версии, если он был установлен ранее;
- удаляет неактуальные логи работы Lynis, если они есть в файловой системе;
- запускает анализ ОС с помощью Lynis;
- обрабатывает информацию об уязвимостях ОС из логов работы Lynis;
- преобразовывает информацию об уязвимостях в base64;
- отправляет Base64 на сервер и получает информацию об устранении найденных уязвимостях;
- записывает все выполняемые скриптом действия в лог-файл.

Результатом выполнения этих действий является сгенерированный скрипт, который содержит в себе решение для устранения уязвимостей, найденных на анализируемом хосте. Этот скрипт запускается после выполнения всех этапов алгоритма, описанного выше.

Для выполнения автоматического запуска командного интерпретатора ОС и реализации алгоритма [13], описанного ранее в [2], было разработано серверное Java-приложение с использованием фреймворка Spring.

Разработанное Java-приложение использует ресурсы nosql базы данных MongoDB [4, 8], в которой содержится информация о хостах, находящихся в анализируемой подсети, а также база знаний с информацией по методам устранения обнаруженных на анализируемых хостах уязвимостях. Базу данных можно представить в виде двух таблиц.

1. Таблица «Информация о хостах в подсети» со следующими полями:

- IP;
- статус верификации;
- статус подключения к сети;
- статус авторизации по ключу;
- дата последнего проведения анализа.

2. Таблица «Информация для устранения обнаруженных уязвимостей» со следующими полями:

- информация об уязвимости;
- информация о методе устранения уязвимости.

Web-интерфейс приложения реализован с помощью фреймворка Thymeleaf и генерируется серверным Java-приложением. Разработанный интерфейс состоит из нескольких HTML-шаблонов, которые заполняются информацией из базы данных. Пользовательский интерфейс представлен на последующих рисунках.

На рисунке 2 приведена стартовая страница выполненной программной разработки.

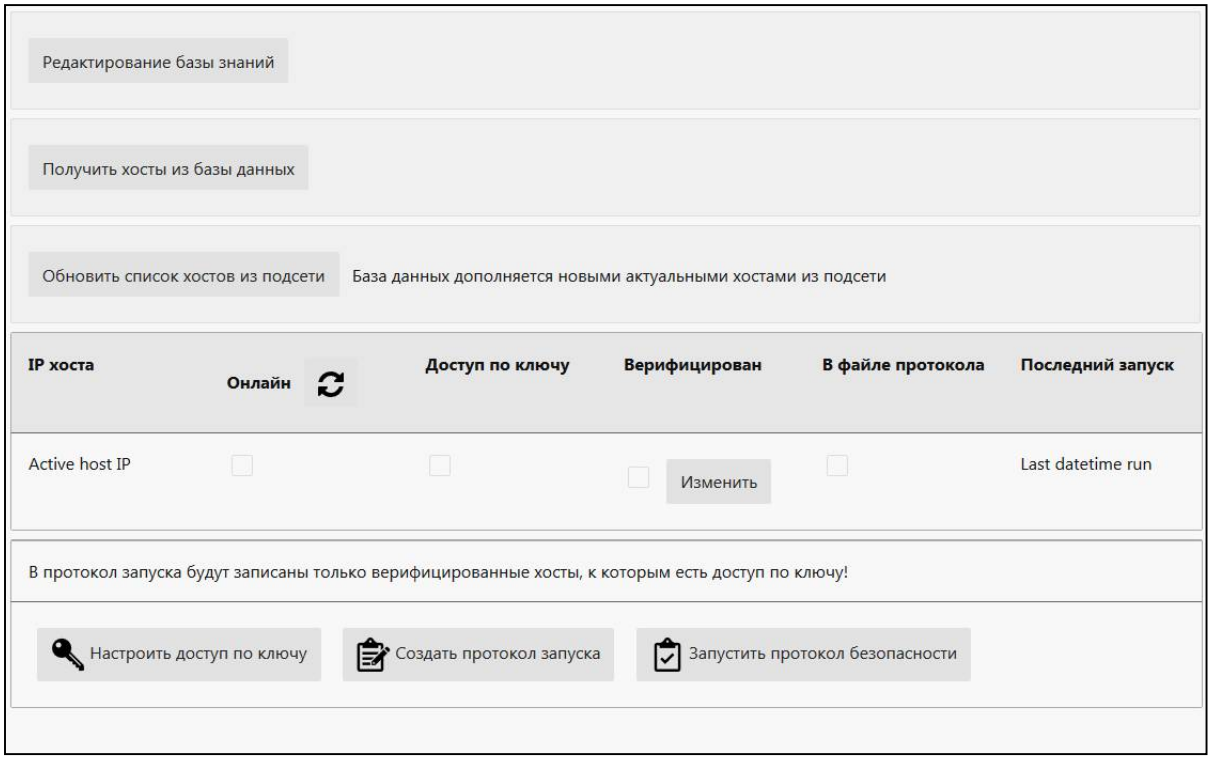


Рисунок 2 – Основной интерфейс серверного приложения ЦСУУ

С помощью инструментов, представленных на этой форме, специалист по информационной безопасности (ИБ) может произвести следующие действия:

- произвести загрузку информации о хостах из базы данных;
- инициировать запуск процедуры анализа подсети с целью получения актуальной информации о хостах, находящихся в данный момент в подсети;
- просмотреть отраженную в таблице актуальную информацию о хостах: IP, статус подключения к сети, статус настроенного защищенного соединения между центральным сервером и хостом-клиентом, статус верификации, находится ли хост в запускаемом файле протокола анализа защищенности хостов подсети, дату и время последнего запуска анализа и устранения уязвимостей на этом хосте;
- инициировать запуск автоматической настройки защищенного соединения между центральным сервером и хостами, для которых такое соединение не настроено;
- выбрать хосты и создать протокол запуска анализа защищенности и устранения уязвимостей на них;
- инициировать запуск протокола анализа и устранения уязвимостей на выбранных хостах;
- верифицировать хост и подтвердить это действие с помощью диалогового окна (рис. 3).

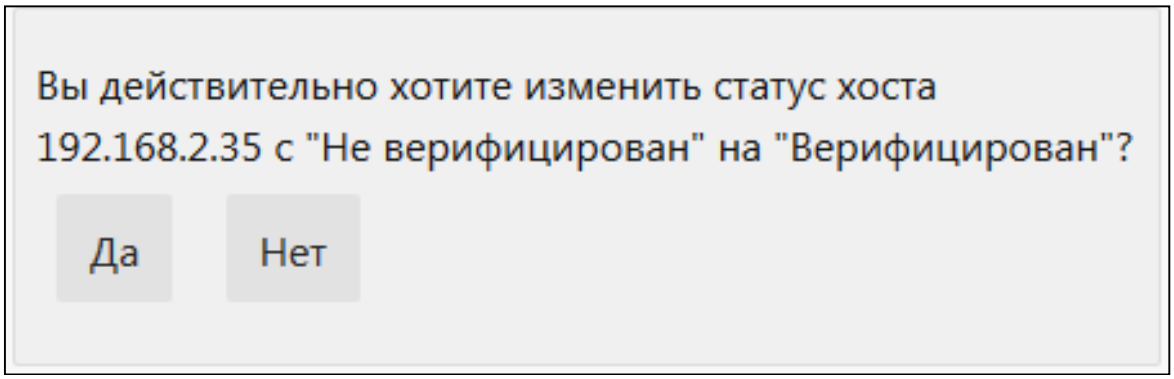


Рисунок 3 – Интерфейс диалогового окна подтверждения верификации хоста

В процессе анализа защищенности и устранения найденных уязвимостей хостов специалист по ИБ может просматривать информацию об успешности или неуспешности завершения процесса устранения уязвимостей на выбранных ранее хостах (рис. 4).

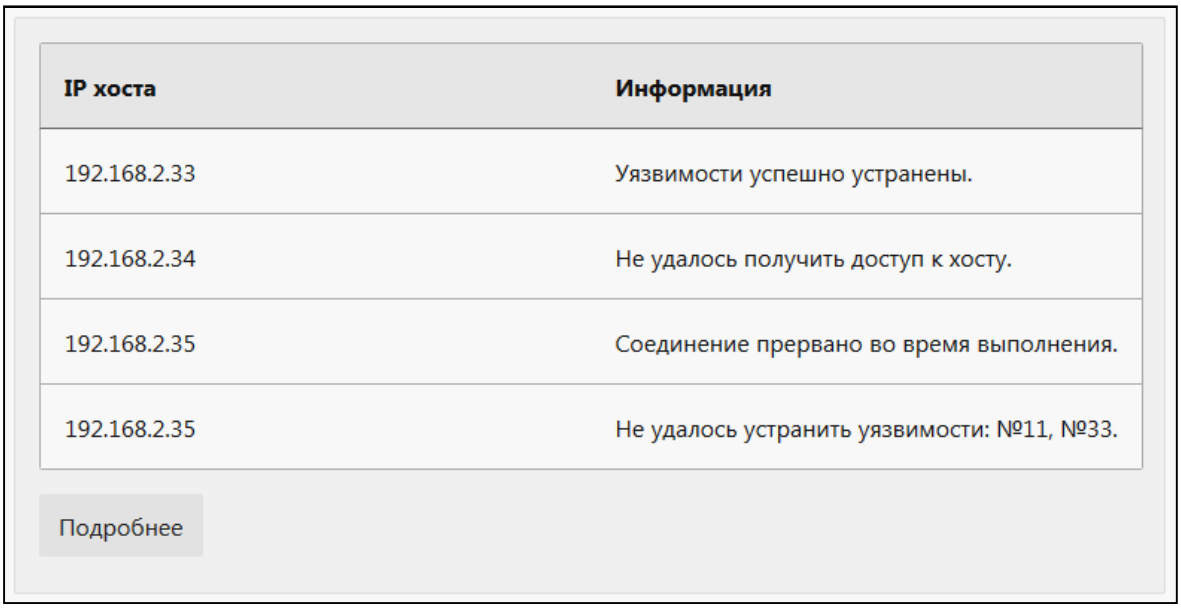


Рисунок 4 – Интерфейс просмотра информации о статусе устранения уязвимостей на хостах

Также доступен просмотр текстового лог-файла, который содержит подробную информацию о процессе устранения уязвимостей на анализируемых хостах.

Для заполнения и редактирования базы знаний, содержащей информацию по методам (средствам) устранения найденных уязвимостей хоста, системный администратор может использовать интерфейс, представленный на рисунке 5.

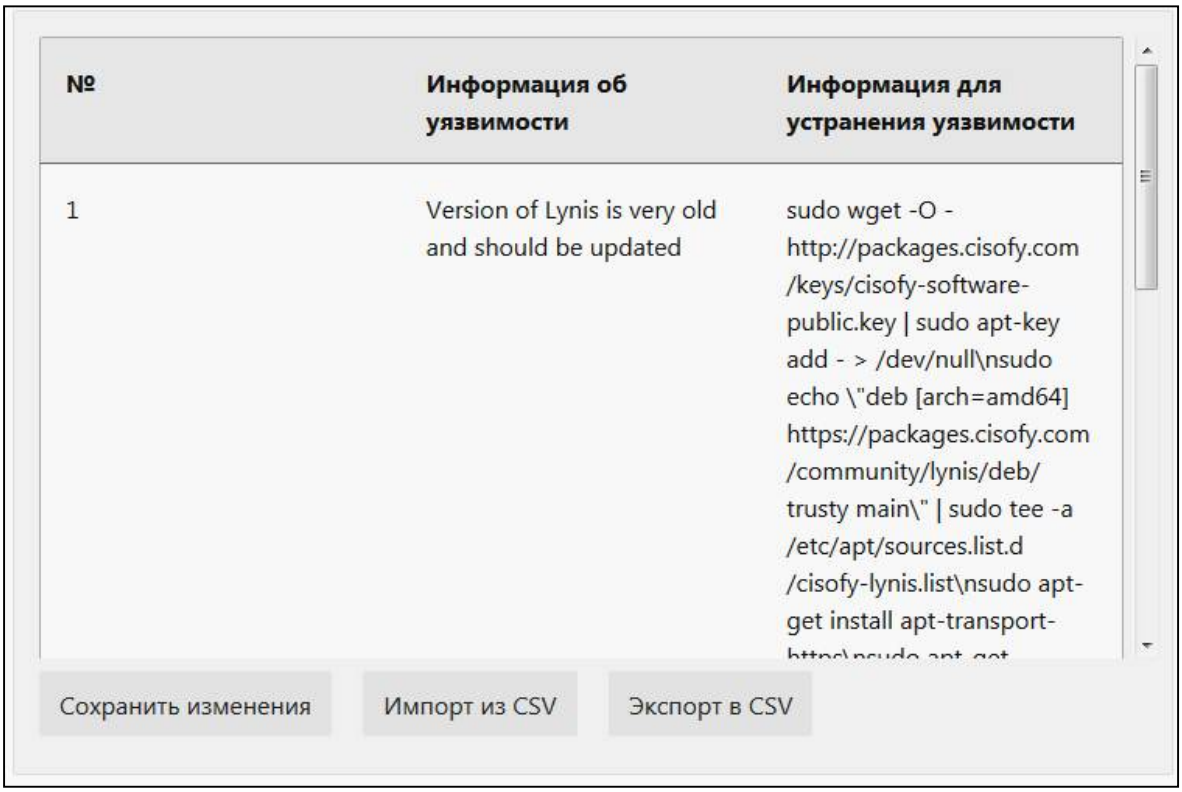


Рисунок 5 – Интерфейс заполнения и редактирования базы знаний

Также администратор может импортировать базу знаний из CSV-файла или экспортировать базу знаний в CSV-файл.

Для описания входных и выходных данных, отправляемых и принимаемых акторами, а также отражения их основных операций (прецедентов) [2], была составлена диаграмма последовательности действий ПО для централизованного устранения уязвимостей (рис. 6).

Составленная диаграмма последовательности действий отражает взаимодействие между акторами при помощи стрелок, передающих управление от отправителя к получателю [10]. Стрелки демонстрируют ход сценария и те события, которые происходят во время анализируемых операций (прецедентов) [2].

Диаграмма последовательностей действий является одним из основных артефактов, необходимых для произведения приемных испытаний. Причина состоит в том, что эта диаграмма является основой при составлении документации для тестирования ЦСУУ.

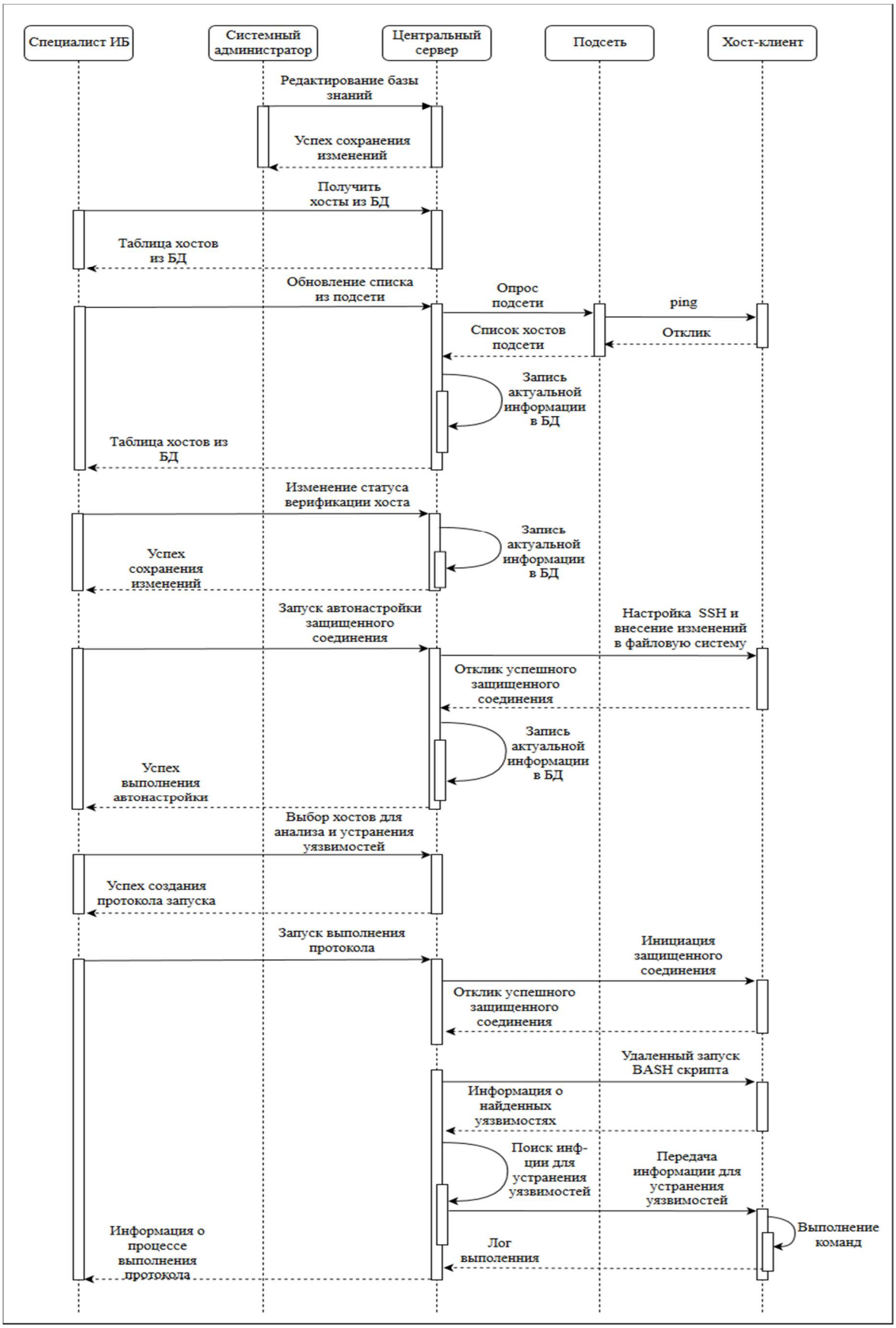


Рисунок 6 – Диаграмма последовательности действий

Выводы. Представлено описание разработанного ПО, в котором отражена структурная схема ЦСУУ, которая включает в себя описание серверного приложения, базы данных, скрипта-анализатора и интерфейса для взаимодействия специалиста по ИБ и системного администратора с ЦСУУ:

- основной интерфейс, в котором отражен список хостов подсети и их статусы;
- интерфейс просмотра информации о статусе устранения уязвимостей на анализируемых хостах;
- интерфейс заполнения и редактирования базы знаний.

Составлена диаграмма последовательности действий акторов.

Разработанное ПО ЦСУУ позволяет инсталлировать его в любой коммерческой организации, в которой используются локальные сети с хостами под управлением Unix-подобных систем; проводить централизованный анализ защищенности хостов с целью дальнейшего устранения диагностированных уязвимостей, тем самым повышая уровень их защищенности.

Описанное в данной статье ПО для централизованного устранения уязвимостей успешно прошло приемные испытания в ЗАО «БАККА СОФТ» (филиал в г. Астрахани) и введено в опытную эксплуатацию.

В процессе приемочных испытаний была произведена оценка скорости работы функции анализа и устранения найденных уязвимостей на хосте: среднее время при типовом наборе уязвимостей «чистой» операционной системы – 180 секунд.

Дальнейшее развитие ПО может включать в себя улучшение алгоритма, выполняющего поиск и применение решений для устранения уязвимостей, с целью повышения его эффективности. Например, решение для устранения какой-либо уязвимости может «закрыть» один из сетевых портов, что может повлиять на работу электронной почты. В этом случае предполагаемый для дальнейшего развития ПО алгоритм поможет предусмотреть возможные последствия применения решений, направленных на устранение уязвимостей, и учесть эти последствия на этапе формирования базы знаний.

Библиографический список

1. Белонежко П. П. Свободные облачные аппаратно-программные платформы. Аналитический обзор / П. П.Белонежко, В. В. Белоус, Н. А. Куцевич, Д. А. Храмов // Науковедение. – 2016. – С. 1–24.

2. Горкавенко В. С. Модель и проектные диаграммы системы централизованной защиты подсети хостов под управлением Unix-подобных операционных систем / В. С. Горкавенко, И. М. Ажмухамедов // Прикаспийский журнал: Управление и высокие технологии. – 2019. – № 1.

3. Дагаев А. Ф. Обеспечение информационной безопасности в вычислительной сети предприятия / А. Ф. Дагаев, А. Н. Самойлов, Е. А. Борисова // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – 2018. – С. 1–4.

4. Достова А. А. Анализ нововведений в объектно-ориентированном языке программирования Java / А. А. Достова, В. В. Тынченко // Актуальные проблемы авиации и космонавтики. – 2018. – С. 10–23.

5. Исхаков С. Ю. Разработка структуры системы управления сетью / С. Ю. Исхаков, А. А. Шелупанов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2011. – С. 259–262.

6. Коджешау М. А. Технологии и алгоритмы информационной безопасности / М. А. Коджешау // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. – 2017. – С. 129–132.

7. Медведев Н. В. Модели управления доступом в распределенных информационных системах / Н. В. Медведев, Г. А. Гришин // Машиностроение и компьютерные технологии. – 2011. – С. 1–19.

8. Настеренко Д. Ю. Объектно-ориентированное программирование на примере языка Java / Д. Ю. Настеренко // Научный журнал. – 2016. – С. 17–30.

9. Оладько А. Ю. Подсистема мониторинга и аудита информационной безопасности в операционной системе Linux / А. Ю. Оладько // Известия Южного федерального университета. Технические науки. – 2012. – С. 22–28.

10. Рогачев А. Ф. Использование UML-моделей для исследования и обеспечения информационной безопасности сложных технических систем / А. Ф. Рогачев, Я. В. Федорова // Известия Нижневолжского агроуниверситетского комплекса: наука и высшее профессиональное образование. – 2014. – С. 1–6.

11. Рукасуева С. Ю. Windows и альтернативные ей операционные системы / С. Ю. Рукасуева, А. П. Багаева // Актуальные проблемы авиации и космонавтики. – 2011. – С. 459–460.

12. Шубин А. Н. Оценка свойств информационных систем в стандартах по информационной безопасности / А. Н. Шубин // Известия Тульского государственного университета. Технические науки. – 2013. – С. 336–345.

13. George K. Thiruvathukal. What's in an Algorithm? / George K. Thiruvathukal // Computing in Science & Engineering– 2013. – № 15. – P. 15–27.

14. Johanson A. Software Engineering for Computational Science: Past, Present, Future / Arne Johanson, Wilhelm Hasselbring // Computing in Science & Engineering 20. – 2018. – С. 90–112.

References

1. Belonezhko P. P., Belous V. V., Kutsevich N. A., Khramov D. A. Svobodnye oblachnye apparatno-programmnye platformy. Analiticheskiy obzor [Free cloud hardware and software platforms. Analytical review]. *Naukovedenie* [Internet Journal Science], 2016, pp. 1–24.

2. Gorkavenko V. S., Azhmukhamedov I. M. Model i proektnye diagrammy sistemy tsentralizovannoy zashchity podseti khostov pod upravleniem Unix-podobnykh operatsionnykh system [Model and design diagrams of a centralized protection system for a subnet of hosts running UNIX-like operating systems]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and high technologies], 2019, № 1, pp. 147–150.

3. Dagayev A. F., Samoilov A. N., Borisova E. A. Obespechenie informatsionnoy bezopasnosti v vychislitelnoy seti predpriyatiya [Ensuring information security in the enterprise network]. *Politematicheskiy setevoy elektronnyy nauchnyy*

zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta [Polythematic network electronic scientific journal of the Kuban State Agrarian University], 2018, pp. 1–4.

4. Dostova A. A., Tynchenko V. V. Analiz novovvedeniy v obektno-orientirovannom yazyke programmirovaniya Java [Analysis of innovations in the object-oriented programming language Java]. *Aktualnye problemy aviatsii i kosmonavtiki* [Actual problems of aviation and astronautics], 2018, pp. 10–23.

5. Iskhakov S. Yu., Shelupanov A. A. Razrabotka struktury sistemy upravleniya setyu [Development of the network management system structure]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki* [Reports of Tomsk State University of Control Systems and Radioelectronics], 2011, pp. 259–262.

6. Kodzheshau M. A. Tekhnologii i algoritmy informatsionnoy bezopasnosti [Information Security Technologies and Algorithms]. *Vestnik Adygeyskogo gosudarstvennogo universiteta. Seriya 4: Yestestvenno-matematicheskie i tekhnicheskie nauki* [Bulletin of Adygea State University. Series 4: Natural Mathematical and Technical Sciences], 2017, pp. 129–132.

7. Medvedev N. V., Grishin G. A. Modeli upravleniya dostupom v raspredelennykh informatsionnykh sistemakh [Models of access control in distributed information systems]. *Mashinostroenie i kompyuternye tekhnologii* [Mechanical Engineering and Computer Technologies], 2011, pp. 1–19.

8. Nasterenko D. Yu. Obektno-orientirovannoe programmirovanie na primere yazyka Java [Object-oriented programming on the example of the Java language]. *Nauchnyy zhurnal* [Initial log], 2016, pp. 17–30.

9. Oladko A.Yu. Podsistema monitoringa i audita informatsionnoy bezopasnosti v operatsionnoy sisteme Linux [The subsystem for monitoring and auditing information security in the Linux operating system]. *Izvestiya Yuzhnogo federalnogo universiteta. Tekhnicheskie nauki* [Proceedings of the Southern Federal University. Technical science], 2012, pp. 22–28.

10. Rogachev A. F., Fedorova Ya. V. Ispolzovanie UML-modeley dlya issledovaniya i obespecheniya informatsionnoy bez-opasnosti slozhnykh tekhnicheskikh sistem [Using UML-models for research and ensuring information security of complex technical systems]. *Izvestiya Nizhnevolzhskogo agrouniversi-tetskogo kompleksa: nauka i vysshee professionalnoe obrazovanie* [News of the Nizhnevolzhsky agrouniversity complex: science and higher vocational education], 2014, pp. 1–6.

11. Rukasueva S. Yu., Bagaeva A. P. Windows i alternativnye ey operatsionnye sistemy [Windows and alternative operating systems to it]. *Aktualnye problemy aviatsii i kosmonavtiki* [Actual problems of aviation and astronautics], 2011, pp. 459–460.

12. Shubin A. N. Otsenka svoystv informatsionnykh sistem v standartakh po informatsionnoy bezopasnosti [Assessment of the properties of information systems in standards for information security]. *Izvestiya Tulskogo gosudarstvennogo universiteta. Tekhnicheskie nauki* [Proceedings of the Tula State University. Technical science], 2013, pp. 336–345.

13. George K. Thiruvathukal. What's in an Algorithm? *Computing in Science & Engineering*, 2013, no. 15, pp. 15–27.

14. Johanson A., Hasselbring W. Software Engineering for Computational Science: Past, Present, Future. *Computing in Science & Engineering*, 2018, no. 20, pp. 90–112.

УДК 004.62

О РАЗРАБОТКЕ МЕТОДА ПРОВЕРКИ ДОСТОВЕРНОСТИ ДАННЫХ ПРИ ПЕРЕДАЧЕ ИНФОРМАЦИИ[1]

Статья поступила в редакцию 14.04.2019, в окончательном варианте – 14.05.2019.

Багдасарян Рафаэль Хачикович, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
кандидат технических наук, e-mail: rafael_555@mail.ru

Осипян Валерий Осипович, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
доктор физико-математических наук, доцент, ORCID 0000-0001- 6558-7998, e-mail: v.osippyan@gmail.com

Лукащик Елена Павловна, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
кандидат физико-математических наук, e-mail: lep_9091@mail.ru

Синица Сергей Геннадьевич, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
кандидат технических наук, e-mail: podrugomu@gmail.com

Жук Арсений Сергеевич, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
аспирант, e-mail: arseniyzhuck@mail.ru

Литвинов Кирилл Игоревич, Кубанский государственный университет, 350040, Российская Федерация, г. Краснодар, ул. Ставропольская, 149,
аспирант, e-mail: lyrik-1994@yandex.ru

Предлагается метод проверки достоверности данных при передаче закрытой информации по открытым каналам связи между узлами сети. Приводятся требования к системам, внутри которых совершается передача данных. Описываются технические аспекты предлагаемого метода, совместно использующего мультиграф, шифрование и биометрическую идентификацию для проверки достоверности передаваемых данных. В схеме, соответствующей данному

[1] Работа поддержана грантом РФФИ № 19-01-00596.